



Reference

VERSION 5.5

Trademarks

Together® is a registered trademark of TogetherSoft Corporation.

LiveSource™, BigPlay™, MindMeld™, Pixie™, and RaftMaker™ are trademarks of TogetherSoft Corporation.

Acknowledgements

HP UX™ is a trademark of Hewlett-Packard Company

Java™, JavaBeans™, Enterprise JavaBeans™ and Solaris™ are trademarks of Sun Microsystems

Rational Rose™ is a trademark of Rational Software Inc.

True64® UNIX™ is a registered trademark of Compaq Computer Corporation

UML™ is a trademark of Object Management Group, Inc.

Weblogic™ is a trademark of BEA Systems, Inc.

WebSphere™ is a trademark of IBM Corporation

Windows, Windows NT, and Windows 98 are trademarks or registered trademarks of Microsoft Corporation.

All other trademarks or servicemarks referenced herein are property of their respective owners.

Table of Contents

About TogetherSoft and Together.....	5
TogetherSoft Corporation	5
Together ControlCenter.....	5
Together's Documentation.....	6
Available formats	6
Help documentation volumes.....	6
Other documentation	7
Let's get started.....	7
Copyright and Trademark Notice.....	8
Context Help.....	10
Add Linked dialog.....	10
Attach to Remote Process Dialog.....	11
Breakpoint Properties Dialog.....	12
Choose Pattern dialog.....	13
Code Template Expert.....	15
Template Editor.....	18
Connection Profile Dialog.....	19
Context Help Dialog.....	21
DBMS Configuration Expert	22
DTD/XSD Export Dialog.....	27
Edit Bookmarks Dialog.....	28
Expression Editor Dialog	29
Extract Operation Dialog	32
File Chooser dialogs.....	33
Generate DDL Expert.....	35
Generate Documentation by Template.....	39
Generate HTML Dialog	41
Import Database Expert.....	43
Inspector Property Builder	46
J2EE Deployment Expert.....	48
J2EE Deployment Expert for IBM WebSphere 3.5	48
J2EE Deployment Expert for BEA WebLogic Application Server	57
New Diagram Dialog	62
New Project - Project Properties dialog	63
Default mode.....	63
Advanced mode.....	64
Buttons	66
Options dialog.....	68
Print Diagram Dialog	69
Print Documentation Dialog.....	70
Print File dialog.....	72
QA: Audit dialog.....	74
Audit results report.....	75
QA: Metrics dialog.....	76

Metrics results report.....	77
QA: Print Dialog	79
Renaming Dialogs.....	80
Search by Query Dialog.....	81
Search for Usages Dialog.....	83
Search on Diagrams	88
Search / Replace Dialogs.....	89
Find/Replace strings.....	90
Selection Manager dialogs	91
Show Hidden Dialog.....	92
Synchronize with Default Printer Dialog.....	92
Version Control interaction dialogs	93
Dialog content and layout	93
Web Services Expert.....	99
Web Services Expert for Apache-SOAP.....	99
Web Services Expert for BEA WebLogic 6.1 beta Application Server.....	101
<i>Index</i>	103



Version 5.5

Thank you for choosing Together®- the exclusive Platform plus Building Blocks software development infrastructure solution for the 21st century.

About TogetherSoft and Together

If this is your first contact with Together® products or TogetherSoft Corporation, this section provides a brief "10,000-foot flyover" to help you get acquainted.

TogetherSoft Corporation

- Dedicated to improving the ways people work together™

TogetherSoft Corporation is the adaptive business-process automation™ company. TogetherSoft's software and services enable enterprises to develop better assets faster by providing proven and predictable ways to manage change, mitigate risk, and deliver frequent, tangible, team-driven results.

Together ControlCenter

TogetherSoft's flagship offering, Together® ControlCenter™, delivers adaptive business-process automation for teams building software solutions. Together ControlCenter brings your e-solutions team together, allowing business users, developers, and operations to collaborate using a common language, diagrams, and software. Together ControlCenter enhances productivity and process management in critical areas: automating mundane business processes (e.g. adaptive documentation generation); automating tedious and error-prone business processes (e.g. work required to adapt and deploy an application on an application server); and automating expert-level insights with guidance on how to adapt and apply those insights correctly (e.g. patent-pending expert-level pattern technology).

For more information, see [Introducing Together](#).

Together's Documentation

Available formats

Together provides a choice of electronic Help documentation formats.

JavaHelp^(tm) Installed with all products. Provides Table of Contents, Index, and Full-text Search. Requires the Sun JRE and JavaHelp runtime (both installed with Together).*

Adobe Acrobat (PDF) Available for download at www.togethercommunity.com. Requires Adobe Acrobat Reader, available free at www.adobe.com. Same content as Together Help documentation for those who want to print hard copy. (No internal hyperlinking.)

Together's Help is also available on the Web. To access it, either go to Together home page at www.togethersoft.com, or select **Help | On the Web** of the main menu and click on an appropriate item from the drop-down list:

www.togethersoft.com - main Together home page

www.togethersoft.com/support - Technical Support info

<http://www.togethersoft.com/order> - Customer Service

<http://www.togethercommunity.com/> - Together Community site

<http://www.togethercommunity.com/docs/> - Application Help

<http://www.togethercommunity.com/contriblist.pl?display=module> - Download Building Block (download the free Together Community Edition 5.0)

Together's Help is also available online at www.togethercommunity.com/docs/. Updates and corrections are posted to this URL between releases.

**(Note: URLs for Web resources are provided as text only, as JavaHelp does not currently support hyperlinking to external targets.)*

Help documentation volumes

Help documentation is arranged in three volumes, each with a particular scope.

Getting Started with Together is a startup guide intended primarily for reviewers, evaluators, and other new Together users who are installing and setting up the product for the first time. This volume also includes copies of the relevant license agreements.

User's Guide covers general introductory issues and provides in-depth explanations about how to accomplish your work using Together. Besides, it includes technical, system, UI, and other reference information related to Together's architecture, User Interface, and API.

Context help, or F1 help, delivers the context-sensitive help information for config options and dialogs. When a dialog is open, press F1 or click *Help* button on the dialog to view the appropriate Help topic.

Besides the main helpset, Together provides separate documentation for the modules that implement some integration or interoperability with the third-party software products.

Other documentation

Help documentation is your main source of information, but it is not the only documentation provided. In addition to Help, you will find:

Readme file: Resides in same directory as the installer program (before you install Together from CD) and in the root directory of your *Together* installation. It contains pre-installation information and tips, and/or late-breaking information not yet incorporated into documentation.

What's New: The file `whats_new.html` is written to the root directory of your installation and contains information about new features and enhancements for the current release. There's also a version history file in the same location if you're interested.

API Documentation: JavaDoc(tm) technical reference for the Together Open API. To access the main index file, open `%TOGETHER_HOME%/doc/api/index.html`

Comments in files: Files in the installation contain comments that you will find useful in the specific context. These include:

- Source files for modules
- Example projects (source and diagrams)
- All the configuration properties files (`./config/*.config`, `*.properties`)
- Example batch and command files (`*.bat`, `*.cmd`, `*,sh`) for launching Together.

Let's get started

People generally tell us that Together is quite easy to set up and begin using. We try to put the basics right up front, making them as easy to find and as intuitive to use as possible. But don't be deceived by the apparent simplicity. There is tremendous capability built into Together, powerful functionality just off to one side waiting until you're ready to unleash it.

Copyright and Trademark Notice

TogetherSoft Copyright & Trademark

Together® and its documentation, modules, samples, and source code are Copyright ©2000, 2001 TogetherSoft Corporation. All rights reserved.

Together is a registered trademark of TogetherSoft Corporation.

Together ControlCenter, Together Solo, and LiveSource are trademarks of TogetherSoft Corporation.

Third-party Trademark Acknowledgments:

HP UX is a trademark of Hewlett-Packard Company

Java, Java2, Java2 Enterprise Edition, J2EE, JavaBeans, Enterprise Java Beans, EJB, JSP, and Solaris are trademarks of Sun Microsystems, Inc.

Rational Rose and Rational Unified Process are trademarks of Rational Software Corporation

SPARC is a registered trademark of SPARC International, Inc.

Tru64 is a trademark of Compaq Computer Corporation

UML, Unified Modeling Language, and CORBA are trademarks of Object Management Group, Inc.

Windows, Windows NT, Windows 2000, Windows98, Windows95 are trademarks or registered trademarks of Microsoft Corporation

Other trademarks referenced herein are the property of their respective owners.

Other Acknowledgments

The "Coad Modeling Components" that ship with Together are based upon the components described in the book: Java Modeling in Color with UML: Enterprise Components and Process, by Peter Coad, Eric Lefebvre, and Jeff De Luca, Prentice Hall, 1999, ISBN 013011510X, and are included with the kind permission of the publisher.

Portions of the Together documentation are derived from concepts and terminology published in The Unified Modeling Language User Guide by Booch, Rumbaugh, and Jacobson. Addison-Wesley, 1998. ISBN 0-201-57168-4.

Other portions of documentation may include material that is Copyright ©1998-99 Object Management Group, Inc. (OMG), and is reproduced by permission.

Portions of on-screen descriptions of Metrics and Audits derived from documentation published online by Sun Microsystems at <http://java.sun.com/docs/codeconv/>.

Together includes software, which is Copyright ©1989, 1991 Free Software Foundation, Inc. All Rights Reserved. That software is subject to the terms of the GNU General Public License available at www.gnu.org. You must accept the terms of that agreement to use that software. See copyright statement and disclaimer below.

Together includes Transformational Patterns as developed in the TROOP (EP 27291, Transformation of Object Oriented design using design Patterns) project funded by the European Commission.

Together includes software, which is Copyright ©1996;1997 Original Reusable Objects, Inc. All Rights Reserved. That software is subject to the terms of the Original Reusable Objects OROMatcher License available at www.savarese.org/oro. You must accept the terms of that agreement to use that software.

Together includes software developed by the Apache Software Foundation (<http://www.apache.org/>), which is Copyright ©1999 The Apache Software Foundation. All rights reserved. See copyright statement and disclaimer below.

Sun J2EE Patterns

The Sun J2EE Patterns Catalog from Sun's Java CenterSM consulting organization is listed in its entirety within this product and will be implemented in future product releases.

The Sun J2EE Patterns are used with permission from the book "Core J2EE Patterns" by Deepak Alur, John Crupi, and Danny Malks, published by Sun Microsystems Press/Prentice Hall. Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303. All rights reserved. SUN PROVIDES EACH J2EE PATTERN "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

GNU Copyright Statement and Disclaimer for bundled CVS version control and gnuMake software

The following information is provided in compliance with the GNU Public License Agreement:

The CVS version control and gnuMake software bundled with some Together editions is Copyright © 1989-1999 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

See also: GNU General Public License Agreement

Apache Copyright Statement and Disclaimer for bundled Apache Tomcat server

The following information is provided in compliance with The Apache Software License, Version 1.1

Copyright ©1999 The Apache Software Foundation. All rights reserved.

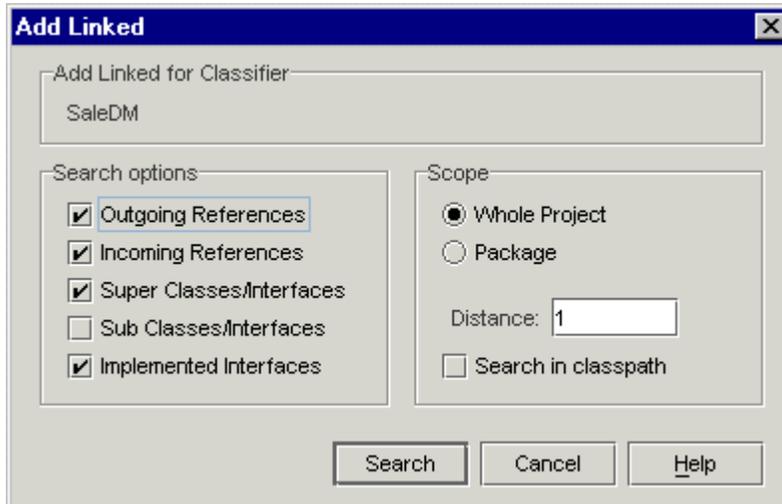
THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org>.

Reference

Add Linked dialog

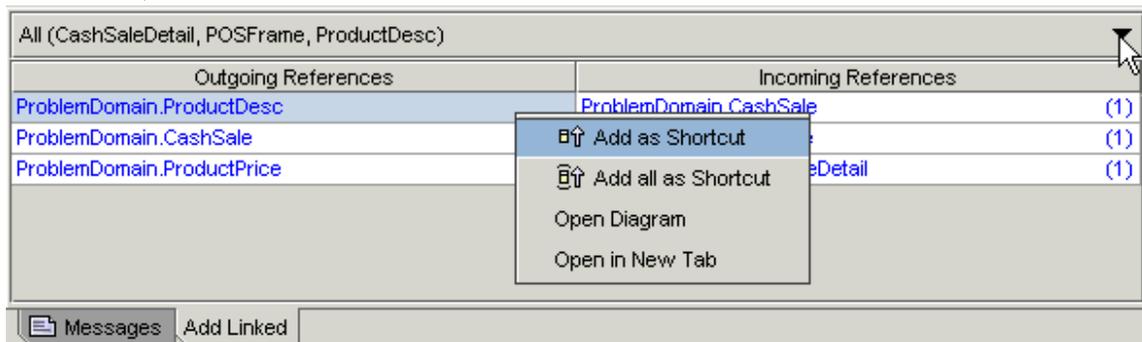
AddLinked dialog is available for class , XML and ER diagrams, and provides search for the references, implementations, and inheritance according to the specified types and scope. The search is applicable to a single objects or to a group of selected objects.



Inheritance and implementation flags are disabled for XML and ER diagrams.

Types of links to be sought are marked in the *Search Option* checkboxes. The search can be performed in the *Whole project* or within a single *Package* . The depth of search is specified in the field *Distance*.

Search result shows in the *Add Linked* tab in the Message pane, and refers to the latest search process. If the result contains references for a number of objects, the Add Linked tab provides a combobox that allows to select references for a particular object, or for all selected objects at once:

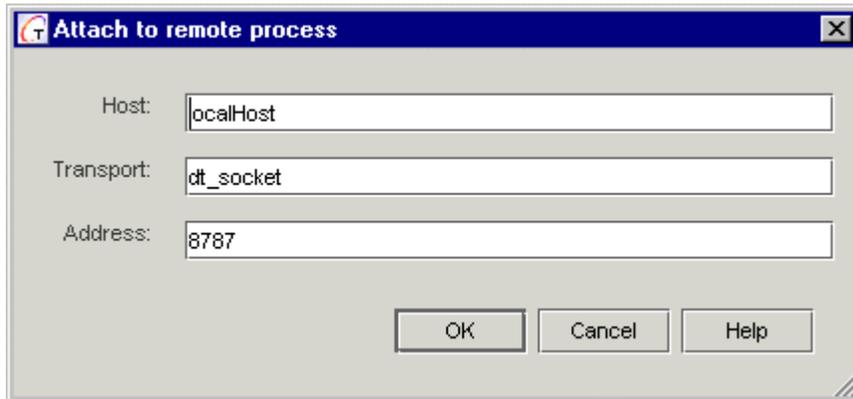


Right click on each entry of the search result invokes a speedmenu that allows to add selected elements as shortcuts to the current diagram, and to open specific fragments of the diagrams. Links between the elements are drawn automatically.

Attach to Remote Process Dialog

This dialog allows to enter connection parameters for remote debugging of an external java program.

See example of the command line to launch the program for debugging in the section Attaching to a Remote Process.



Host: host name of the computer where the external program runs.

Transport: socket name

Address: port number

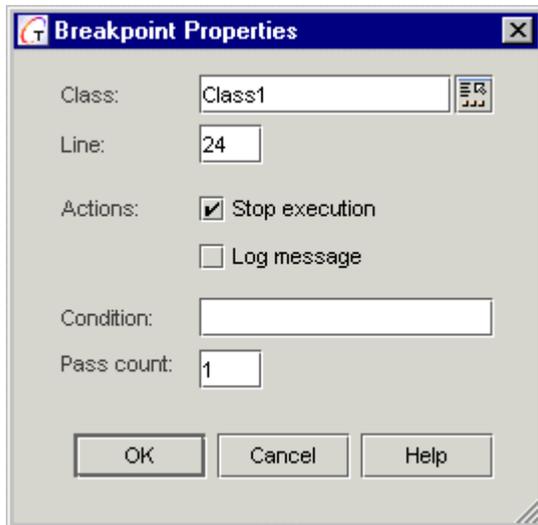
OK: saves changes and closes the dialog.

Cancel: discards changes and closes the dialog.

Help: displays this topic.

Breakpoint Properties Dialog

Breakpoint Properties dialogs are available for the line, class, method and exception breakpoints.



Class: enter the name of the class where the breakpoint will be set. A class chooser button is available, to select the desired class from the classpath or model.

Line: enter number of the line where the breakpoint will be set.

Actions: Set flags that control the program behavior at the breakpoint. Note that both flags may not be cleared.

Stop execution: check this flag, if you want to stop execution at the breakpoint.

Log message: checking this flag displays the messages about all encountered breakpoints in the *Console* tab of the Debugger. The messages are displayed in blue font.

Condition: Enter a valid boolean expression that returns `TRUE` or `FALSE`. The return value is evaluated every time when a breakpoint is encountered. If condition is specified for a breakpoint, the Debugger breaks execution, when the expressions returns `TRUE`.

Pass count: specify number of passes through this breakpoint before stop.

OK: applies changes and closes the dialog.

Cancel: discards all changes and closes the dialog.

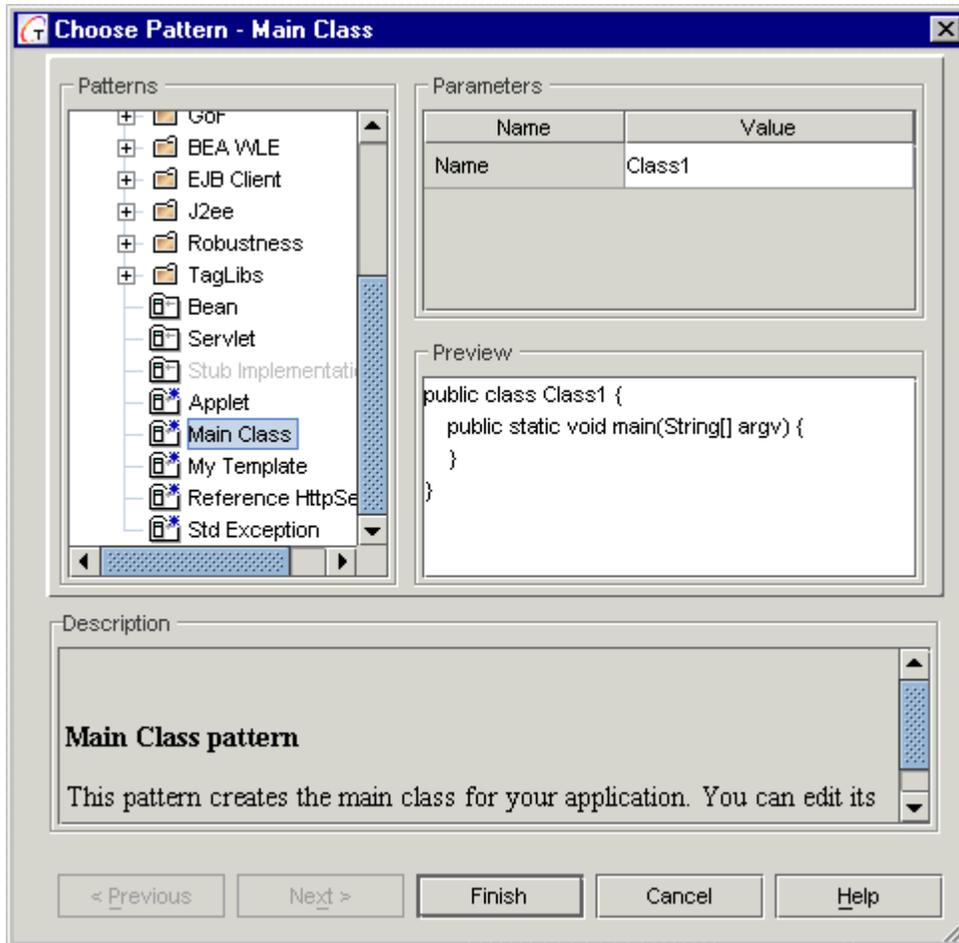
Help: displays this topic.

Choose Pattern dialog

Invoking the dialog

This dialog may be invoked in several ways:

- Class diagram toolbar (Class by Pattern, Link by Pattern)
- Class or member speedmenu (Choose Pattern)
- Object menu (when class or member is selected in diagram)
- Explorer node speedmenu for a class or member (Choose Pattern)



Choose Pattern dialog as invoked from Class by Pattern on the diagram toolbar

Panes

Choose Pattern dialog has two panes:

Selector/Editor: occupies the top portion of the dialog. Initially it displays two panels:

Pattern: presents an Explorer treeview of the available patterns for the current operation. For example, if you invoked the dialog from "Link by Pattern", only link patterns are displayed. Note that templates and patterns display different icons in the patterns treeview (templates are marked with an asterisk).

Parameters: displays relevant information about default patterns, including the source that will be generated upon selection. For more advanced patterns, this pane is blanked.

If you select an advanced pattern requiring custom specifications, the wizard buttons are enabled and the above panels are replaced by edit controls appropriate to specify parameters, etc. for the selected pattern.

Description: Displays context-sensitive Help text for the more advanced patterns such as GoF or EJB.

Help for the more complex patterns appears directly in the dialog rather than in the *Together* Help system. You don't need to leave your work and switch back and forth between applications. Just click on an individual pattern to read its associated Help text. **Note:** some of the basic patterns do not display Help text. This is intentional. More advanced patterns such as the GoF should have Help.

See also

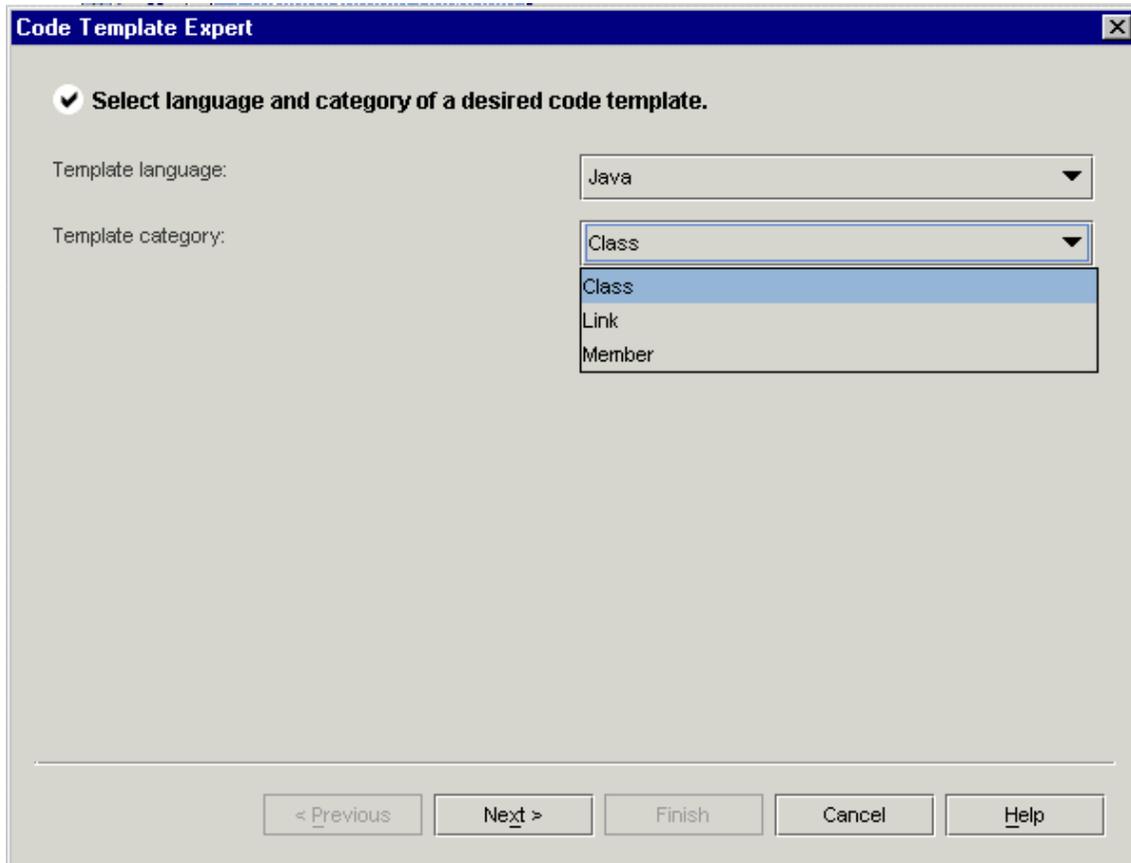
Working with patterns

Code Template Expert

This dialog facilitates handy creation of custom code templates. To invoke the expert, select Code Template Expert command from the Tools menu.

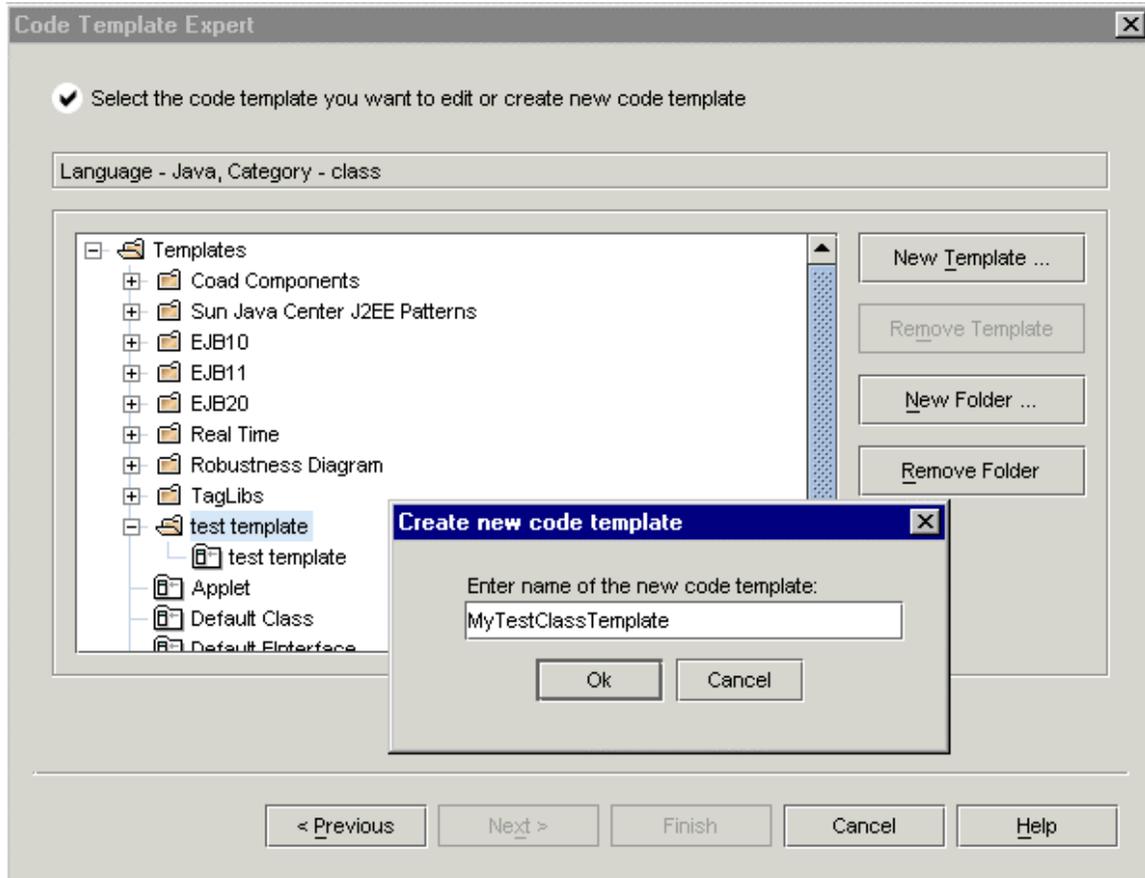
1st page

The first page provides two drop-down lists, where you can select language and category of the desired template:



2nd page

The second page of the expert displays the treeview of the Templates folder, and the buttons to add/remove templates and folders.



New Template

Press this button to invoke *Create new code template* window and enter the template name.

Remove template

Deletes the highlighted template. **Note:** This button is disabled for the default templates.

New folder

Press this button to create a new folder under the selected node.

Remove folder

Deletes the highlighted folder. **Note:** This button is disabled for the top level folder.

3rd page

On the third page you can set up template properties and edit template code and description.

Default name

Enter the name of the element to be generated by this template.

Generate prologue and epilogue

Check this box to generate pre-defined prologue and epilogue comments in the resulting code.

Hide in choose list

If this box is unchecked, the template shows in the Template Chooser, invoked by Class by Pattern, Link by Pattern, or Member by Pattern dialogs. Always checked for the default templates.

Paste all created classes to one file

Check this box to allow placing several classes to a single file.

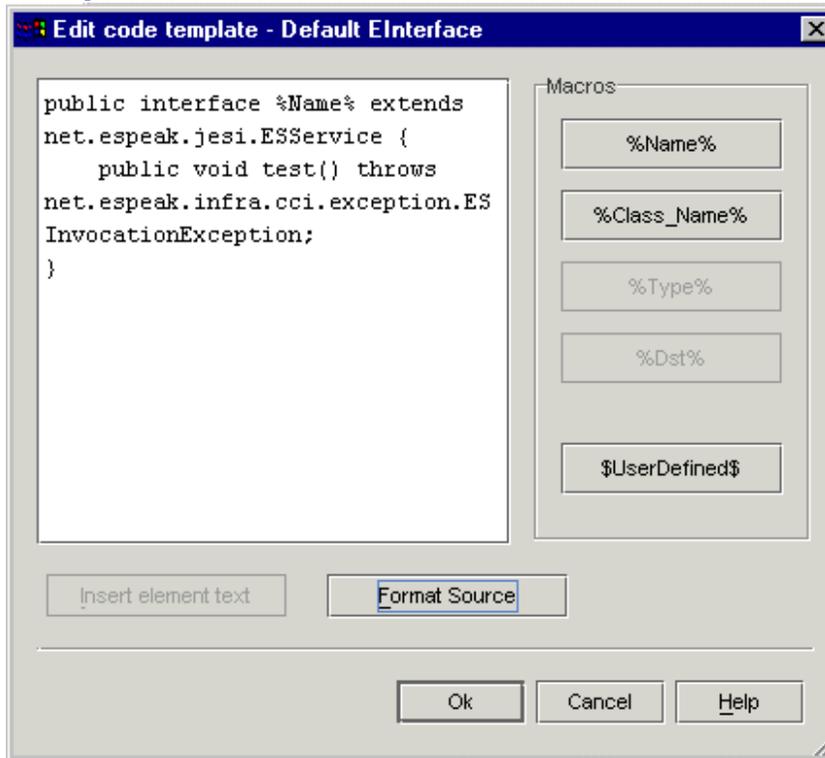
Template description

Text area to edit or enter description of the selected template.

Edit template code

Press this button to invoke the template editor:

Template Editor



In the text area you can edit template source code. Availability of the macros buttons depends on the selected template type. For example, *%Type%* and *%Dst%* buttons are disabled for the Class templates.

Template Editor Buttons

Macros buttons

%Name% inserts name of a class, interface, link or member.

%Class_Name% inserts class name. This button is disabled for the Link templates.

%Type% inserts type of a link or member. This button is disabled for the Class templates.

%Dst% inserts destination of a link. This button is disabled for the Class templates.

\$UserDefined\$ inserts *\$UserDefined\$* string that can be replaced with the required macro.

Insert element text

This button is enabled in case of creating template by class. Press to insert source code of the selected diagram element into the template. When creating template from scratch, this button is disabled.

Format source

Provides proper indentation of the source code.

OK

Completes operation, closes the editor and returns to the last page of the Code Template Expert.

Cancel

Discards all changes and closes the editor.

Help

Invokes this topic.

See also

Template macros

Connection Profile Dialog

This dialog is invoked from DBMS configuration Expert or from Import Database Expert by clicking on *Add profile* or *Edit profile* buttons. Both dialog windows differ in title only.

Fields

Profile name

Enter the name to be stored in the profile list

JDBC driver

Enter JDBC driver name.

Driver Location

Specify driver for the selected database server. The driver can reside locally or remotely.

URL

Read-only field displays location and name of the source database. The displayed value is composed from the settings specified in the following fields according to the URL pattern.

URL pattern

This is DBMS-dependent string with four predefined macros: %prefix%, %port%, %host%, %database%. These macros can follow in arbitrary order. Each particular DBMS allows to use only a certain set of macros. Using macros is the only convenient way to arrange connection URL. While you are typing parameters in *Prefix*, *Host name*, *Database* and *Port* fields, *URL* field changes automatically according to *URL Pattern* field.

Prefix

A string of a form "jdbc:<subprotocol>:" where <subprotocol> is the driver name or the name of database connectivity mechanism. For example, "odbc", "oracle:thin" etc.

Database

Database identifier varies with the database driver used, but generally provides a logical name that is mapped by the database administration software to a physical directory where the database tables are located. To make your database identifier meaningful, you must register the name using your database administration software. (The process of registration varies from platform to platform.)

Host - Port

If you are connecting across a network, these are used to identify the remote machine (they are disabled for MS Access and Cloudscape).

User name - Password

Initialized to the server's defaults. Specify appropriate values for both fields. Leave blank for MS Access databases that do not require log-in.

Fill in the required parameters and press Next. Together attempts to connect to the specified database. Waiting for connection time is limited by the value specified in *Connection time out* node of the Options dialog | Database tab.

Buttons**Test connection**

Tries to establish connection with the specified parameters, thus allowing to make sure that profile is correct.

OK

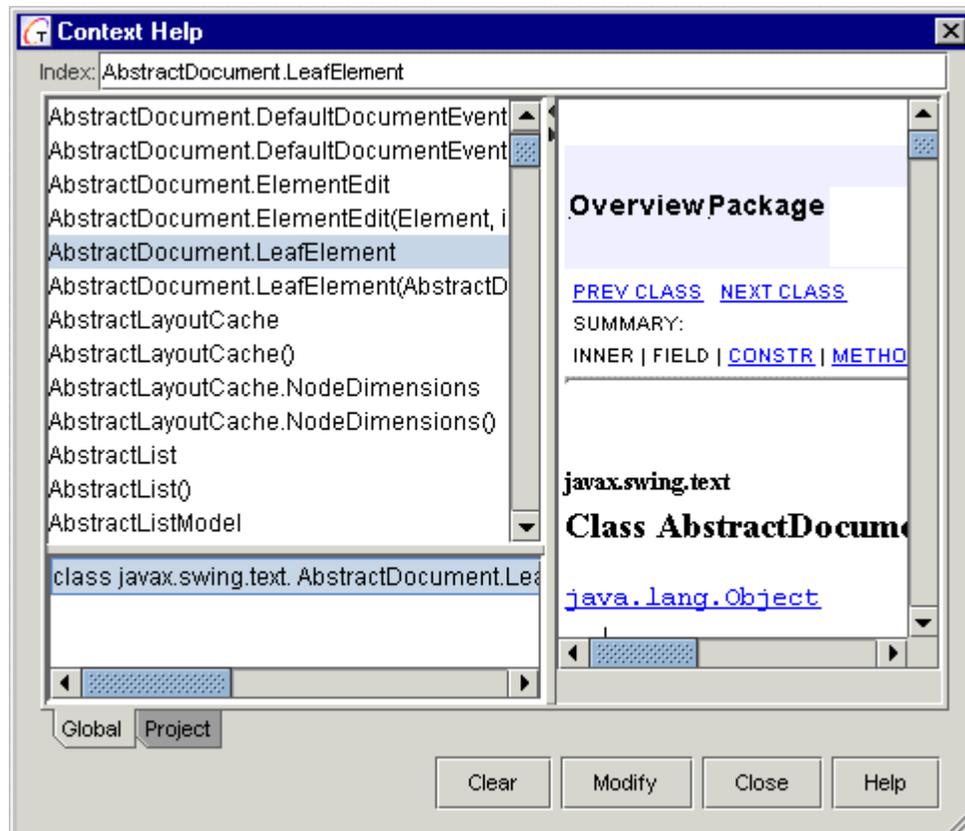
Closes the dialog and saves new or modified profile in the list.

Cancel

Discards all changes and closes the dialog.

Context Help Dialog

This dialog is available upon pressing F1 button on the source code in the Editor pane. Initially, the dialog displays blank database. You can create new help database, or modify the existing one on the global, or project level, by selecting *Global* or *Project* tab.



Index field displays the keyword for which the context help is invoked. The upper left frame displays the list of available documents. The lower left frame displays fully qualified path to the selected document. The right frame displays documentation for the selected class or package.

Buttons

Clear

Resets the current help database. Pressing this button invokes confirmation dialog. Choose 'yes' to clear the database, 'no' to keep its current content.

Modify

Invokes Selection Manager dialog, which allows to change the current help database. Refer to the description of this dialog for details.

Close

Saves changes and closes the dialog window.

Help

Invokes this topic.

See also

Context Help

Selection Manager dialog

DBMS Configuration Expert

New DBMS Configuration Expert is intended to create configuration files for arbitrary JDBC-enabled relational database management systems.

Using this Expert, you can:

- Obtain parameters of the database used in Together through JDBC driver.
- Edit this information if necessary
- Generate configuration file and use it same way as the configuration files for the databases supplied with *Together*.

1st Page

The first step is very similar to that of Import / Export Experts. To avoid repeated typing of connection parameters, Together allows to use connection profiles, which contains all necessary parameters. By default the profile list is empty.

Use *Add profile* and *Edit profile* buttons to create and modify connection profiles.

Server name

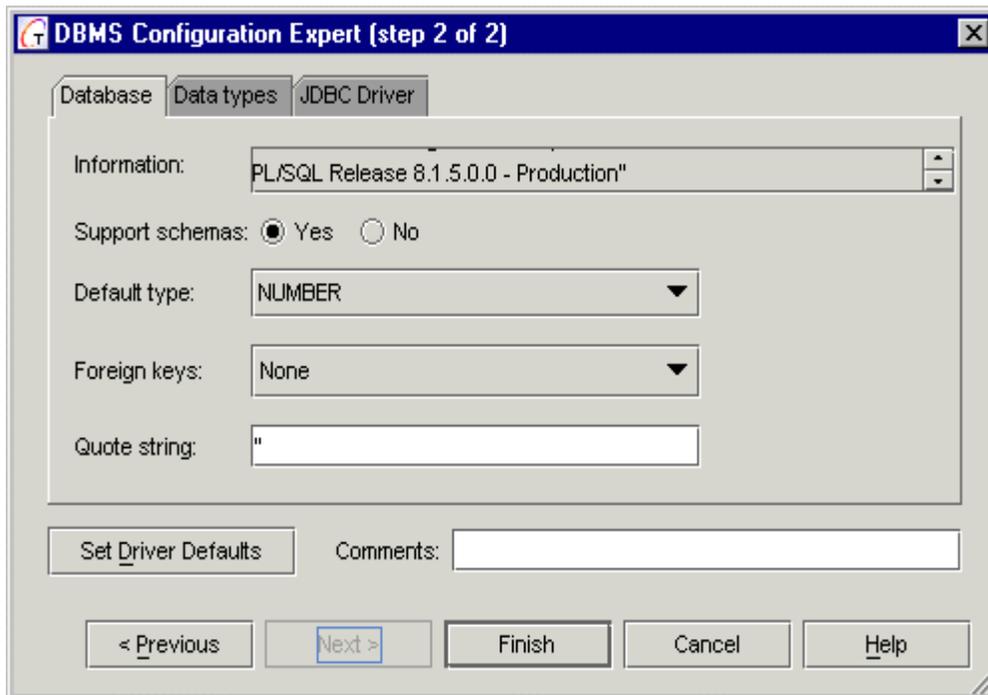
Enter the name of the server to which you wish to connect. This value will also be used for the name of the generated config file.

Which profile will you use for connect?

Select the required profile from the drop-down list.

Having entered in all necessary information, click *Next* to continue.

2nd Page



If Together successfully connects to the database, the second page of the expert shows up. On this page you can:

- Obtain information about DBMS
- Obtain information about JDBC driver being used
- Get the values of various DBMS parameters through the JDBC driver
- Adjust obtained parameters if necessary

The 2nd page of the Expert contains three tabs: *Database*, *Data Types* and *JDBC Driver*.

Database tab

This tab contains the following fields and controls:

Information

This field displays database name and version.

Support schemas

Two radio buttons allow to define whether the DBMS supports the schemas. By default the setting is defined by the information returned by the driver.

Default type

This field provides a dropdown list to select the default type that will be used in creating attributes. By default one of the integer types is assigned.

Foreign keys

Choose the way to calculate the foreign keys from the dropdown list. There are 4 possible options:

None: do not use foreign keys

Table: Calculate foreign keys for a certain table

Model: Calculate foreign keys for the entire schema

Heuristic: This value can be used as an alternative for those databases who do not support foreign keys (for example, MySQL), or when the driver doesn't enable obtaining foreign keys information through JDBC (MS Access, MS SQL). This method is recommended if neither Table nor Model allow to correctly import the foreign keys data.

Note: Some drivers enable getting foreign keys information using Table, the others support Model, still the other drivers support both. There are no rigid standards, and the only way is to take risks of changing settings for each specific driver.

Quote String

Specifies a string (normally a single character) that encloses all names of the table, attributes and restrictions involved in DDL generation. See also Options | Database | Quote names.

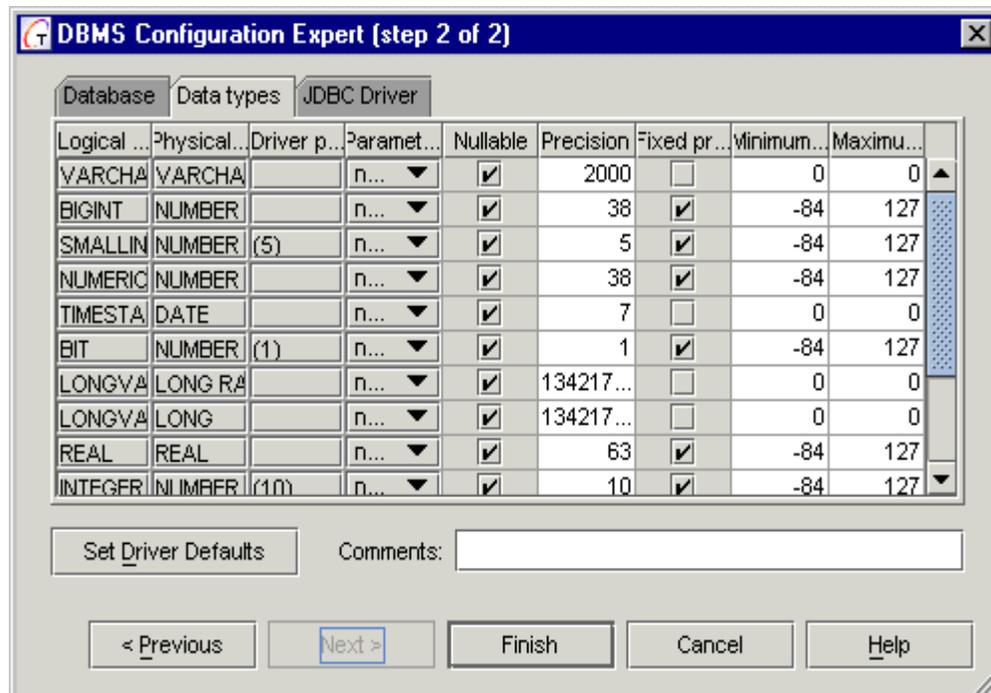
Set Driver Default Button

Pressing this button restores default driver settings.

Comments

Add comment to the generated configuration file.

Data types tab



This tab displays the list of supported types, returned by JDBC driver. First three columns are read-only.

Logical type

Displays the name of JDBC logical type that corresponds to the specified type.

Physical type

Actual name of the DBMS type.

Driver parameter value

This column displays possible parameter values for the types, returned by the driver.

Parameters

In this field you can choose parameters for the types taking into consideration the Driver parameter values. There are four possible options:

None: Don't use parameters for a type.

%size%: for the types that enable size specification, for example, CHAR(45), VARBINARY(100).

%precision%, %scale% template: for the decimal types that enable defining fraction size, for example DECIMAL(6,2), NUMERIC(10,5). In this case, *%precision%* stands for the entire length of the number, and *%scale%* stands for the length of fractional part after decimal delimiter.

Default value: The driver can return specific value for the type. For example, ORACLE driver returns parameter values (1), (3), (5), (10) for the NUMERIC types. Use Default Value to set specific value for the parameter.

Note: *none* is assigned by default.

Nullable

Select the checkboxes to allow NULL values for a certain type.

Precision

The value in this column depends on the selected parameter. If *%size%* is selected, the value entered in this field defines maximum allowed value size. If *%precision%, %scale%* template is selected, the field defines maximum allowed *%precision%* size.

Fixed precision/scale

Set this option for the types with specific numeric value of the parameter, e.g. for ORACLE type NUMERIC (5). If this option is selected for a certain type, editing *Size* and *Digit* values for this type is disabled on ER diagram.

Minimum scale

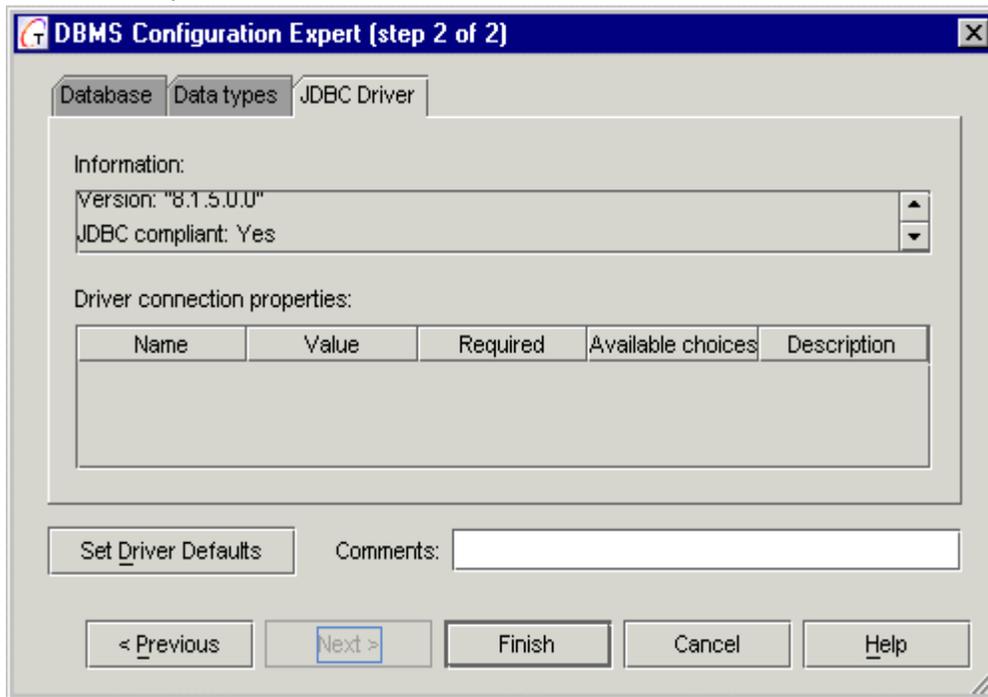
Specify minimum allowed fraction size for the types with *%precision%, %scale%* parameter.

Maximum scale

Specify maximum allowed fraction size for the types with *%precision%, %scale%* parameter.

JDBC Driver tab

This read-only tab allows to view information about the driver in use.



Information

This field displays name and version of the driver in use, and also informs if the driver is JDBC compliant. JDBC compliance requires full support for the JDBC API and full support for SQL 92 Entry Level.

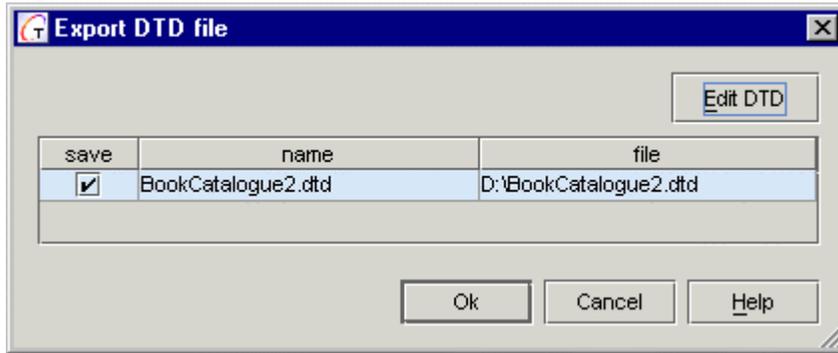
Driver connection properties

Displays additional connection parameters information, if any.

Having filled out all the required fields, press *Finish*. The Expert will generate the resulting configuration file, which you can use without restarting *Together*.

DTD/XSD Export Dialog

DTD/XSD Export feature allows to export XML structure diagram into a DTD or schema file, for further use. To invoke this dialog, choose DTD Import-Export | Export to DTD or XSD Import-Export | Export to XSD respectively on the Tools menu. Both dialogs differ in title only.



Name and target file location of the XML structure diagram are specified in the Properties inspector. DTD/XSD Export dialog displays the current inspector settings.

If *save* checkbox is set, resulting DTD or schema file for the selected XML diagram will be saved in the location defined by the *file name* field on the *Properties* tab of the inspector.

Buttons

Edit DTD/XSD

Invokes text editor window and displays contents of the resulting DTD file.

OK

Completes operation and closes the dialog.

Cancel

Discard all changes and closes the dialog.

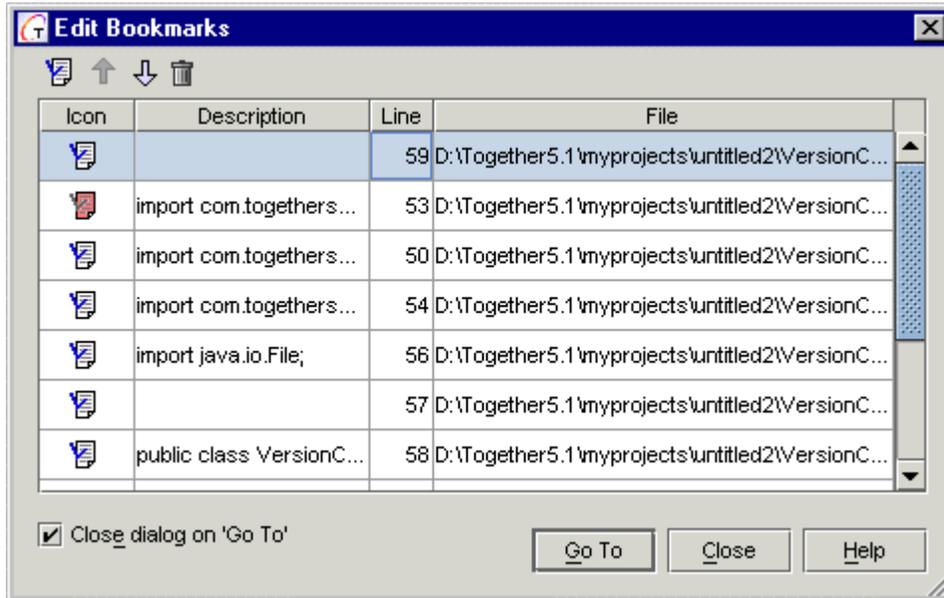
Help

Invokes this topic.

Edit Bookmarks Dialog

Edit Bookmarks dialog displays a list of all the bookmarks currently set in files in the current project. In this dialog you can:

- Edit the default bookmark description
- Reorder the list of bookmarks
- Visually classify bookmarks using different icons
- Delete individual bookmarks



Toolbar

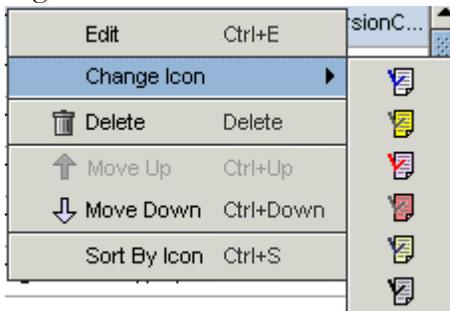
Toggles the current bookmark

, Navigates through the list of bookmarks

Deletes the current bookmark

Menu

Right-click on the selected bookmark in the list displays its speedmenu:



Edit: activates in-line editor in the Description field

Change icon: displays the list of available bookmark icons

Delete: removes the selected bookmark from the list and from the file opened in the Editor

Move Up/Move Down: Navigates through the list of bookmarks

Sort by Icons: Visually classifies bookmarks.

Buttons

Go To: navigates to the line in the Editor that corresponds to the selected bookmark. The dialog can be closed, if *Close on 'Go To'* option is selected.

Close: Saves all changes and closes the dialog window.

Help: displays this page.

Expression Editor Dialog

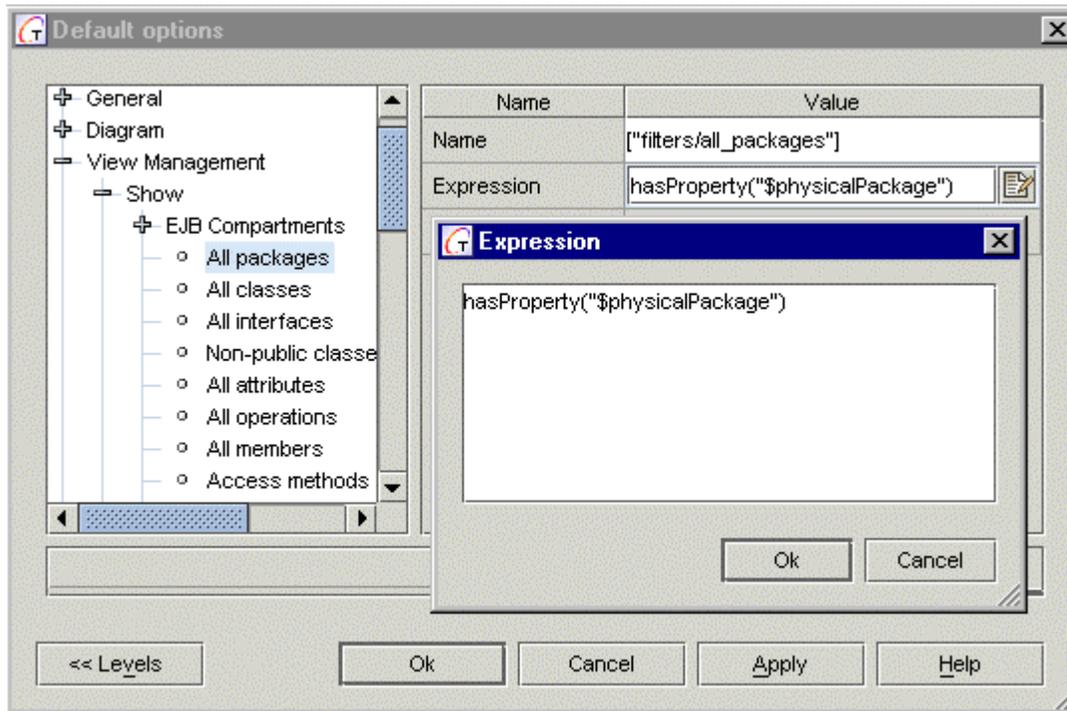
The "Expression" field is the essence of a mechanism for filtering out elements in the diagrams. It contains an expression written using the queries syntax described below.

This is how it works: before *Together* displays an element, it first checks if some "Show ..." options are turned off. If the user has unchecked some options, *Together* then:

- applies the "Expression" query of each unchecked option to the element.
- if the returned value for at least one expression is "true", then an element will not be shown. Otherwise, it will be visible.

Don't let this description confuse you. In practical terms: if you do not wish to show certain elements in the diagram, create an expression that will return "true" for the kind of element you want to hide. The effect is that if you uncheck that option, your expression will filter out all unwanted elements.

Together ships with a set of predefined filters sufficient for most cases. Nevertheless, if you wish to adapt *Together* to your special needs, here is an explanation of the expression syntax used in the "Expression" field.



Expression field syntax

The "Expression" field must contain a Boolean query. A query is a command recognizable by *Together* which can be run immediately and can perform some actions. A query can also return some result-- strings, objects, or a Boolean value for example.

Several Boolean queries can be grouped into one Boolean expression to formulate a nontrivial condition, using these logical operators: **!**, **&&**, **||** (they behave just like corresponding Java/C++ logical operators), and parentheses. For example:

```
((query1) || (query2)) && (!query3)
```

Possible Boolean queries are:

- `isDiagram()` - checks if an element is `RwiDiagram`.
- `isLink()` - checks if an element is `RwiLink`.
- `isMember()` - checks if an element is `RwiMember`.
- `isNode()` - checks if an element is `RwiNode`.
- `isPackage()` - checks if an element is `RwiPackage`.
- `hasProperty(propertyName)` - checks if an element has the specified `RwiProperty`.

The `propertyName` is a quoted string containing a name of a property. For example:

```
hasProperty("$interface")
hasProperty("$public")
```

- `hasPropertyValue(propertyName, propertyValue)` - checks if an element has the given value of the specified property.

The `propertyName` is a quoted string containing a name of a property. The `propertyValue` is a quoted string containing a value of a property. For example:

```
hasPropertyValue("$shapeType", "Class" )
hasPropertyValue("$shapeType", "AssociationLink" )
hasPropertyValue("$shapeType", "Attribute" )
```

The names and values of some properties are defined in the RWI API, so it is expected that you are familiar with RWI (if you are not, see the API documentation in the `\doc` directory under your installation). The property names are the attributes of `RwiProperty` interface. The following table lists the attributes and appropriate properties that are most often used in queries expressions. Refer to the relevant API for detailed descriptions of the fields and return values.

Interface attribute	Property
ABSTRACT	\$abstract
AGGREGATION	\$aggregation
ASSOCIATES	\$associates
BY_VALUE	\$byValue
COMPILED	\$compiled
CONSTRUCTOR	\$constructor
DESTRUCTOR	\$destructor
DOC	\$doc
EXTENDS	\$extends
FINAL	\$final
FULL_NAME	\$fullName
HYPERLINK	\$hyperlink
IMPLEMENTS	\$implements

Interface attribute	Property
INITIAL_VALUE	\$initialValue
INLINE	\$inline
INTERFACE	\$interface
LANGUAGE	\$language
LINK_MEMBER	\$linkMember
MEMBER_LINK	\$memberLink
NAME	\$name
NATIVE	\$native
PACKAGE_LOCAL	\$packageLocal
PARAMETER	\$parameter
PARAMETERS_TEXT	\$parametersText
PRIVATE	\$private
PROTECTED	\$protected
PUBLIC	\$public
RETURN_TYPE	\$returnType
SHAPE_TYPE	\$shapeType
STATIC	\$static
SYNCHRONIZED	\$synchronized
TEXT	\$text
THROW	\$throw
TRANSIENT	\$transient
TYPE	\$type
VOLATILE	\$volatile

Here is how to create your own filter. Suppose, you want to be able to omit those protected classes in the diagrams which are read-only or implement any interface: In the "User defined" Show option, open an "Expression" field and enter this value:

```
((hasPropertyValue("$shapeType", "Class") && (!hasProperty("$interface"))) &&
hasProperty("$protected")) && (hasProperty("$readOnly") ||
hasProperty("$implements"))
```

Caveat

When editing filter expressions or configuration files, always refer to the Open API for the most up-to-date information.

Extract Operation Dialog

Name and modifiers

Name:

Visibility: Other `static` `final` `synchronized`

Comment:

Operation parameters

type	name	comment
java.math.BigDecimal	paymentAmt	
java.math.BigDecimal	total	
java.math.BigDecimal	change	

Operation body preview

```

/** Generated by Together on Aug 31, 2001 Checks the amount of
private void checkChange(BigDecimal paymentAmt, BigDecimal to
    if (change.compareTo(new BigDecimal(0.0)) < 0) {
        NumberFormat cf = NumberFormat.getCurrencyInstanc
        String msg = new StringBuffer("Payment of ").appe
            .append(cf.format(total)).toString();

```

Ok Cancel Help

Fields

Name and Modifiers

Name: enter the name of the method to be created on the base of selected code fragment.

Visibility: choose visibility modifier from the drop-down list. Possible options are: *private*, *package local*, *protected*, *public*.

Other: set the flags *static*, *final* or *synchronized* if necessary.

Comment: Type in your comment, which will appear in the javadoc comment before the method.

Operation parameters

This table displays the list of the method parameters generated on the base of selection analysis. *name* and *comment* columns are editable.

Operation body preview

This read-only field displays generated method.

Buttons

OK: closes the dialog and puts Extract Method into action with the current settings.

Cancel: discards all changes and closes the dialog.

Help: invokes this topic.

File Chooser dialogs

The Together® File Chooser dialog can take two forms: for choosing files and for choosing paths.

Each of these dialog forms represents the file system in a tree view, from which you can access a speedmenu (right click or Shift+F10) offering commands that are appropriate to the currently selected node.

The speedmenu commands allow you to:

Create new folder: This command is available only from a folder-level speedmenu because a new folder is created within the currently selected folder.

Rename, or in-place edit: This command enables you to change the name of the currently selected folder or file. Alternatively, you can use the F2 shortcut key to do in-place editing of the current node. You can also initiate in-place editing by doing a triple mouse click on the node or by a click, pause, click sequence, followed by a short delay.

Copy path name: This command copies the full path name of the current node to the system clipboard. This can be useful when you want to paste the full path name in another location. You can see the full path name of a selected node as a tool tip.

Refresh: This command refreshes the folder content. This can be useful when you have changed the file structure in an external application. This command can also be initiated using the F5 shortcut key.

Delete: This command deletes the currently selected folder or file. This can be useful when, for example, you need to delete a new folder you have created by error. This command can also be initiated by using the Delete key. A confirmation dialog is presented for all delete requests in the File Chooser dialogs. **Warning:** note that deleting a folder also deletes all sub-folders and files within the deleted folder.

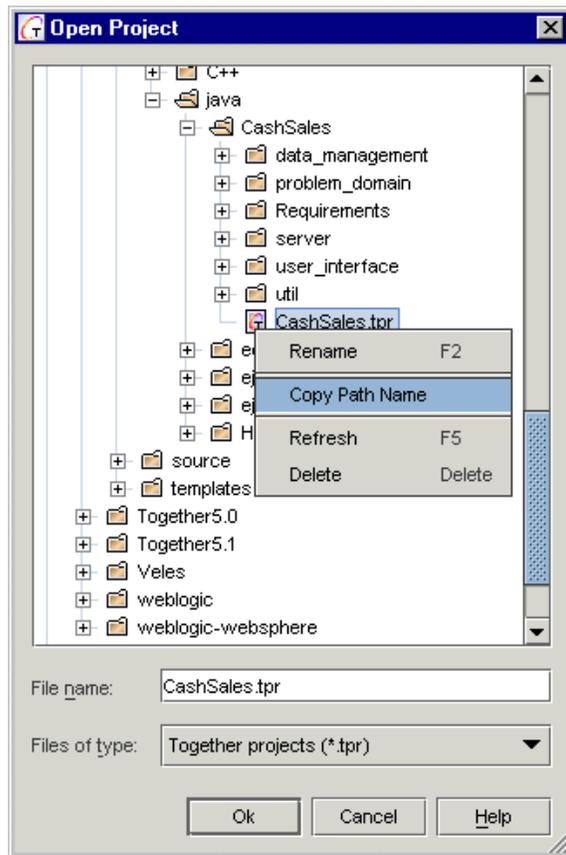
The File Chooser dialogs also offer such useful features as:

Multiple selection: The file and path File Chooser dialogs allow multiple node selections within the same call to the dialog. When multiple selections are made, however, the speedmenus of the individual selections are not available.

Expand and open functions: Use the Enter key to expand a currently selected tree-view folder or open a selected file. Alternatively, double click on the selected node.

Favorites: Note that Favorites node is added to the bottom of the treeview for the sake of handy search.

The two forms of the File Chooser dialog are illustrated here:

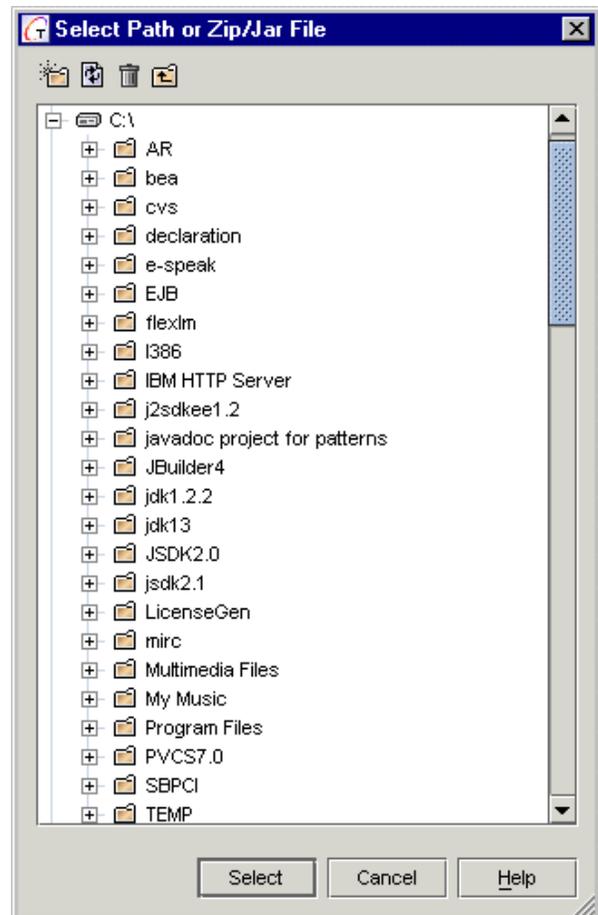


File Chooser dialog

The file form of the File Chooser dialog offers *File Name* field to enter manually the file name, and *Files Of Type* drop-down list to select appropriate file-type filter. Depending on the context of the File Chooser dialog, the file-type filters available may be limited. For example, in the dialog shown above, its Open Project context limits the file-type filter to displaying only Together project files (*.tpr).

Path Chooser dialog

The path form of the File Chooser dialog enables selection of folders or archives (such as .zip or .jar) according to an internal filter.

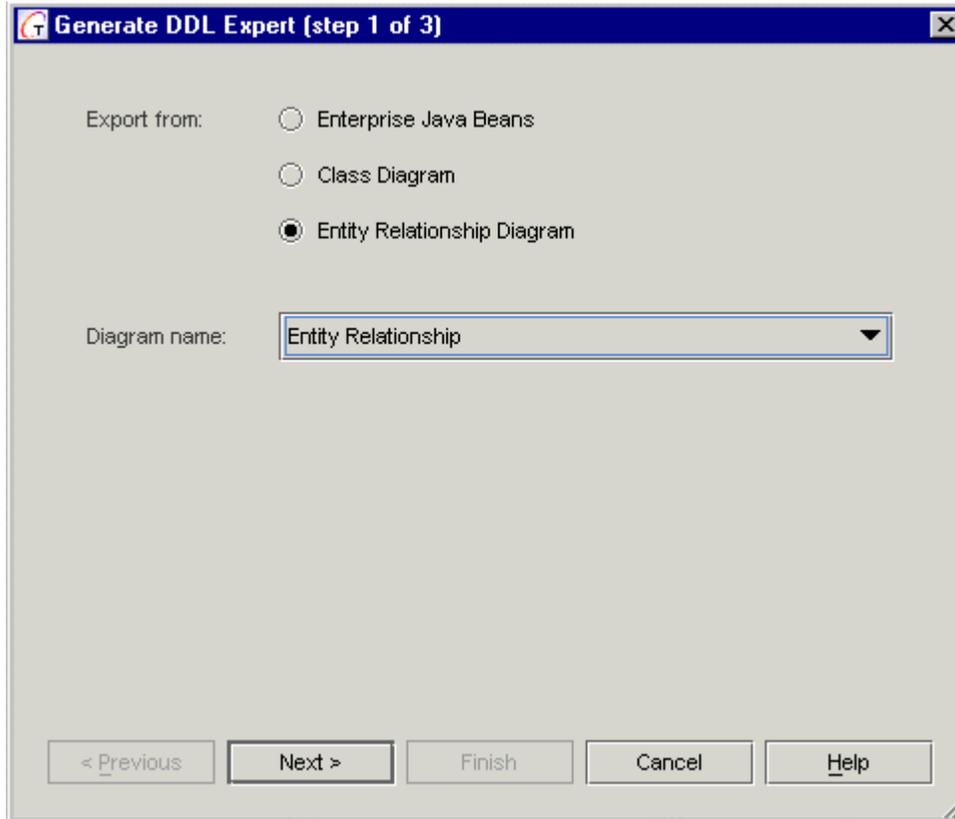


Generate DDL Expert

Generate Database dialog is a multi-screen dialog that allows you to create new table(s) in a specified database.

1st Page

Export from



Specify which type of diagram your want to export from. The table(s) to be created can be described by a diagram of any of the following types:

Enterprise Java Beans: relationships in the database are mapped to associations between supplier Bean class and client Remote interface. Identifying relationship is presented by two associations:

- Client Remote interface is associated by both the supplier Bean and Primary Key classes.
- Similarly the associations between Beans (and, maybe, Primary Keys) and Remote interfaces transform to non-identifying (identifying, respectively) relationships. Other types of associations between EJBs are not supported.

Class Diagram: recreates the database's structure in terms of classes with attributes whose types are Java equivalents of the database's column types (such class diagrams will closely resemble an ER diagram).

Entity Relationship Diagram: a high level data model that shows the major entities and relationships which support a general business area. The objective of the entity relationship diagram is to provide a view of business information requirements sufficient to satisfy the need for broad planning for development of its information system.

Diagram name

Select a diagram from among the currently opened diagrams (for example, if you choose the "ER Diagram" as the "Export from" option value, the "Diagram name" field will contain only currently opened ER diagrams). The database table(s) will be created according to the selected diagram.

2nd Page

On the second page of the expert you can define what's to be done with the exported information, using two radio buttons: *Generate DDL only* and *Generate and Run*.

Generate DDL only

If you choose this option, the Expert brings you to the next page, where you can define the destination file for the generated script.

Generate and run

If you select Generate and run, you can run the script immediately.

Server type

Choose the target database server type from the dropdown list of currently supported server types.

JDBC driver

This is a read-only field that informs you about the driver being used. Displays the driver for the server selected in Server type field.

Driver Location

Choose location of the driver for the server being used. The driver can be located locally or remotely.

URL

Read-only field displays location and name of the source database. The displayed value is composed from the settings specified in the following fields.

Prefix

A string in a form jdbc:<subprotocol>: where <subprotocol> is the driver name, or the name of a database connectivity mechanism. For example, odbc , oracle:thin, etc.

Database

Database identifier varies with the database driver used, but generally provides a logical name that is mapped by the database administration software to a physical directory where the database tables are located. To make your database identifier meaningful, you must register the name using your database administration software. (The process of registration varies from platform to platform.)

Host - Port

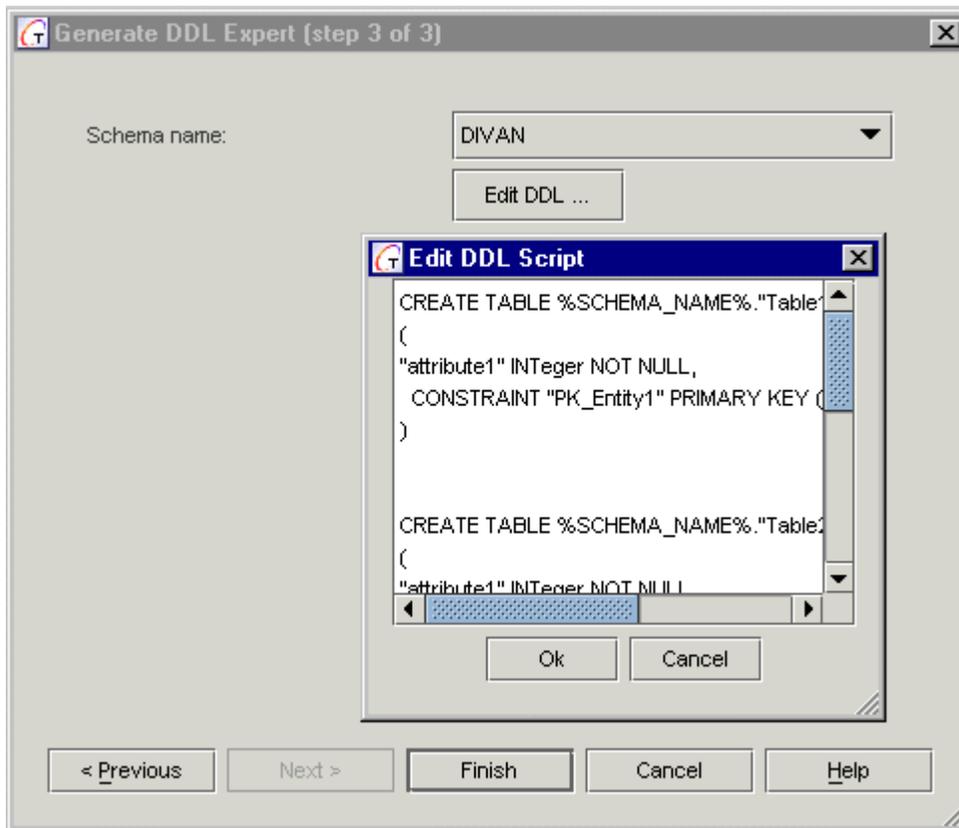
If you are connecting across a network, these are used to identify the remote machine (they are disabled for MS Access and Cloudscape).

User name - Password

Initialized to the server's defaults. Specify appropriate values for both fields. Leave blank for MS Access databases that do not require log-in.

Fill in the required parameters and press Next. Together attempts to connect to the specified database. Waiting for connection time is limited by the value specified in *Connection time out* node of the Options dialog | Database tab.

3rd Page



If connect succeeds, the second page of the Expert shows up with a choice of possible schemas. The view of the 3rd page depends on the action selected in the 2nd page.

Generate and run option selected

If you have selected *Generate and run* on the 2nd page of the Expert, this screen contains both the Schema name choice field and the Edit DDL button.

Note: Since MS Access and mySQL don't support schemas, the *Schema* option will be disabled for MS Access and mySQL.

Edit DDL

If you click *Edit DDL* button, a simple editor displays the generated DDL script. You can review the script and make any changes manually. Click OK when ready to return to the Generate Database dialog.

Generate DDL only option selected

If you selected *Generate DDL only* on the previous screen, the last screen of the dialog displays the File name field rather than the schema chooser. It likewise has the Edit DDL button.

If you want to generate DDL script file to a location other than the default, specify the target path and filename.

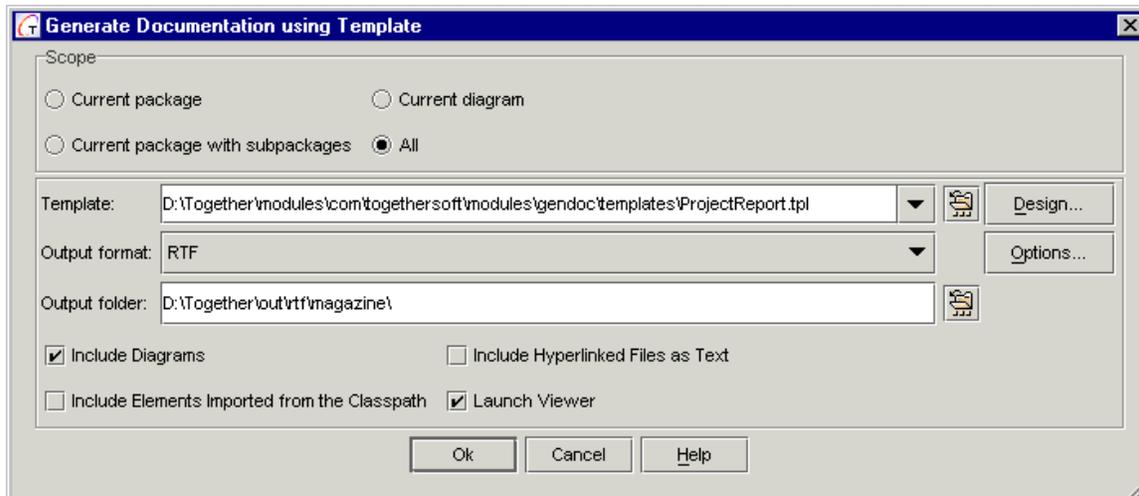
Click *Finish* to complete the database export process.

See also

Exporting model information

Generate Documentation by Template

This dialog is invoked from the Tools | Documentation | Generate Using Template command.



Use the Scope section to specify what part(s) of the project will be parsed and included in the generated output.

Current package: Only the current package selected in the Model tab of the Explorer is included in the generated output

Current package with subpackages: Only the current package selected in the Model tab of the Explorer, and any subpackages under it are included in the generated output.

Current diagram: Only the current diagram open and focused in the Diagram pane is included in the generated output

All: The entire project is included in the generated output.

Use the lower section to specify the template by which documentation will be generated, output format and destination, and optional actions for the process.

Template: Specify location of the template to be used. You can select one from the history dropdown list, or press the file chooser button to choose one of the available templates. By default the file chooser navigates to the Templates folder of GenDoc module.

Design: Press this button to invoke Template Designer.

Output format: Select HTML, RTF or TXT format from the dropdown list.

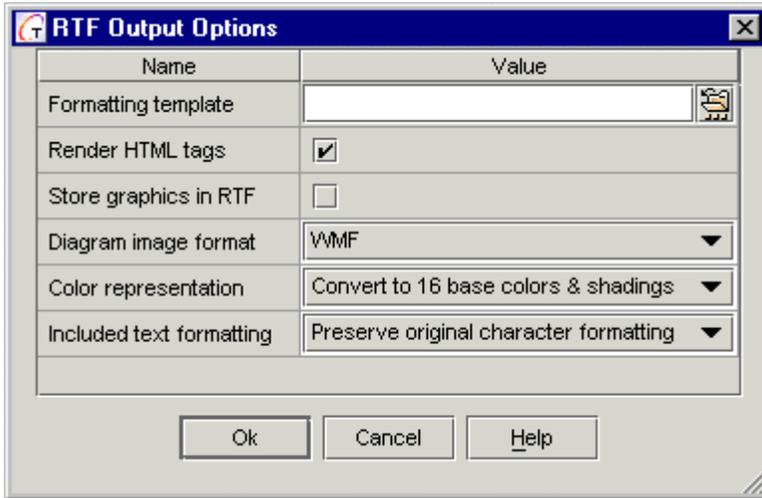
Output folder: Specify the location for the generated documentation output, or select from the file chooser.

Include elements imported from classpath: Check if you want to include these elements in the generated output.

Launch Viewer: Check if you want to launch the browser registered on your system and load the generated documentation when the docgen process is completed.

RTF documentation

Selecting RTF format enables Options button that allows to set output options for the printed documentation.



Formatting template: enter fully qualified name of the desired template file, or select one using the file chooser button.

Render HTML tags: check this flag to show HTML tags as appropriately formatted text in the printed documentation. The following tags are now supported:

```
<b></b>, <i></i>, <u></u>, <h1></h1> ... <h6></h6>, <code></code>,
<tt></tt>, <em></em>, <font color></font>, <pre></pre>, <p>, <br>, <ol>,
</ol>, <ul>, </ul>, <li>.
```

Store graphics in RTF: check this flag to make DocGen produce a single RTF file with all graphics data inside.

Note: RTF format doesn't support gif format and provides png format as the replacement. Hence, storing gif images may cause problems with some RTF viewers. However, Word 2000 handles RTF files with gif images properly. To avoid problems, it is advised to re-save RTF files with gif images, generated by DocGen, in Word 2000. So doing, Word automatically replaces all gif images with png images. Be aware, that to make this working, you have to make a minor change to the file (for example, insert/remove a space).

Diagram image format: select the desired image format from the dropdown list. Presently wmf and gif formats are available.

Color representation: select the desired color scheme from the dropdown list.

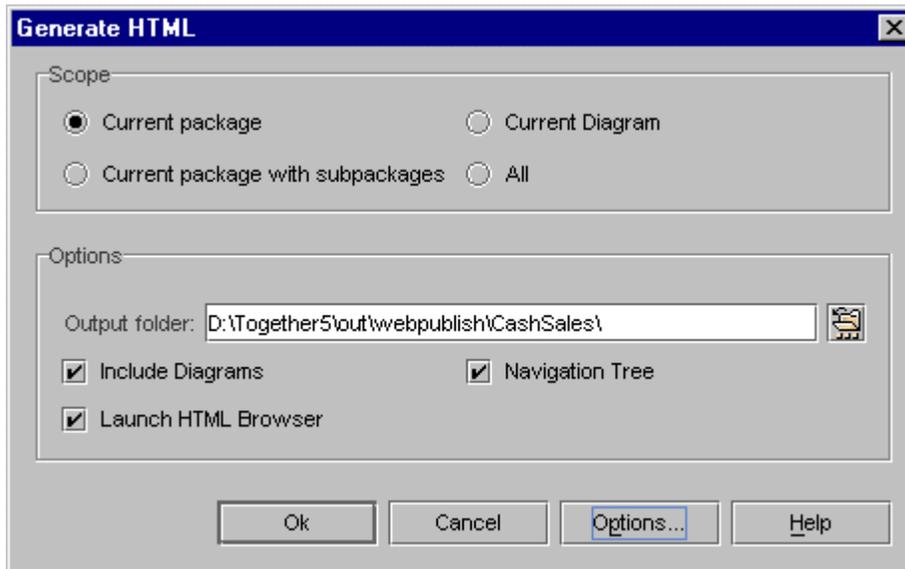
Included text formatting: you can preserve original formatting, or use the one from the specified template.

Buttons

OK: Accepts the input and starts the doc generation process.

Cancel: Cancels your input and closes the dialog.

Generate HTML Dialog



Generate HTML dialog (Tools | Documentation | Generate HTML)

Scope

Use this section to specify which part(s) of the project will be parsed and included in the generated output.

Current package: Only the current package selected in the Model tab of the Explorer, or in the Diagram icon or workspace, is included in the generated output

Current package with subpackages: Only the current package selected in the Model tab of the Explorer, or in the Diagram icon or workspace, and any subpackages under it are included in the generated output.

Current diagram: Only the current diagram open and focused in the Diagram pane is included in the generated output

All: The entire project is included in the generated output.

Options

Use this section to specify the location of the output, and optional actions for the process.

Output folder: Specify the location for the generated documentation output

Include diagrams: Check if you want to include GIF images of diagrams in the generated output. Consider checking this option if you have very large diagrams, as some HTML browsers will not handle large GIF images.

Launch HTML browser: Check if you want to launch the HTML browser registered on your system and load the generated documentation when the docgen process is completed.

Navigation tree: Check if you want to include a frame containing a navigation treeview in the generated output. This treeview enables you to visually navigate the hierarchy of your project and access the relevant documentation.

Buttons

OK: Accepts the input and starts the doc generation process.

Cancel: Cancels your input and closes the dialog without generating documentation.

Options: Launches the Options dialog at the Project level displaying only the *Generate HTML* page. You can change any of the configuration options before generating documentation. Note that your changes will persist at the Project configuration level after the documentation generation finishes. If you only want to change settings temporarily for the process, choose Options | Project | Generate HTML after docgen is finished and reset the options to their previous values.

Import Database Expert

Import Database Expert enables you to specify a database and select tables whose structure you want to model in Together project. You can import database into ER diagram, or Class diagram. To invoke the expert, choose Database Import-Export | Import Database Expert on the Tools menu.

1st page

Server for connect

Choose server type for the source database from the drop-down list. .

Connection profile

Select an existing connection profile and create a new one. Using profiles spares you from repeated typing of connection settings. See Connection Profile dialog for details. Use *Add* and *Edit* buttons to create new profiles or modify the existing ones.

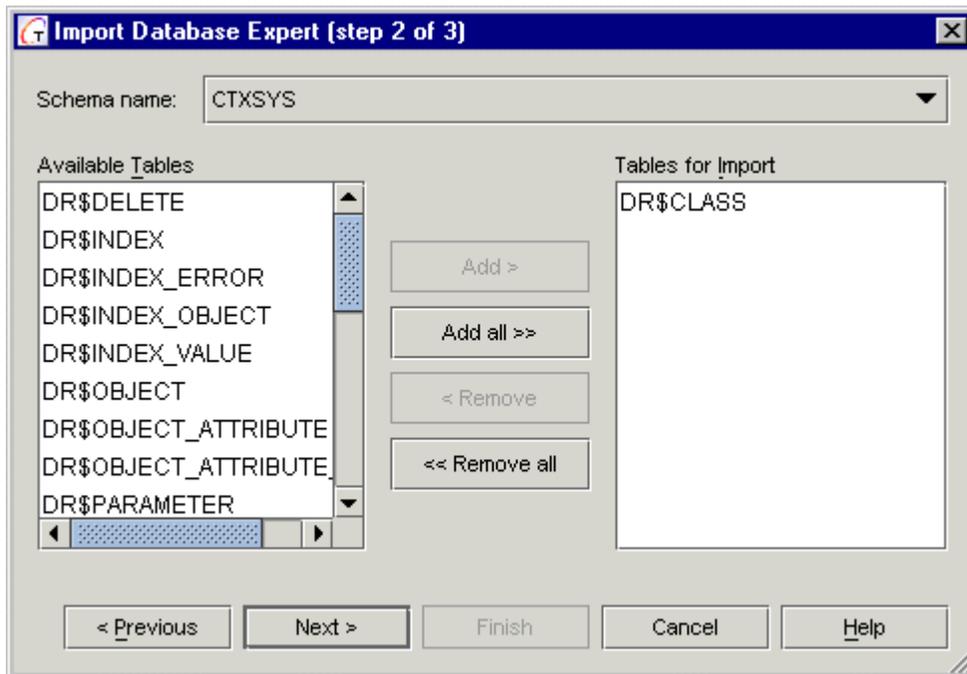
Connection URL

This a read-only field that displays the connection string generated according to URL pattern (see Connection Profile dialog for details).

User name and Password

Enter user name and password to access the database. Both values are validated against the selected connection profile.

2nd Page



If connect succeeds, the second page of the Expert shows up with a choice of possible schemas and the list of tables pertaining to a certain schema.

Schema name

A schema is a logical set of tables and other database objects. Usually, it is thought of as simply "the database", but some databases can hold more than one schema. For example, a star schema can be a set of tables where one large, central table holds all important information, and is linked, via foreign keys, to dimension tables which hold detail information, and can be used jointly to create detailed reports.

Available Tables / Tables for Import

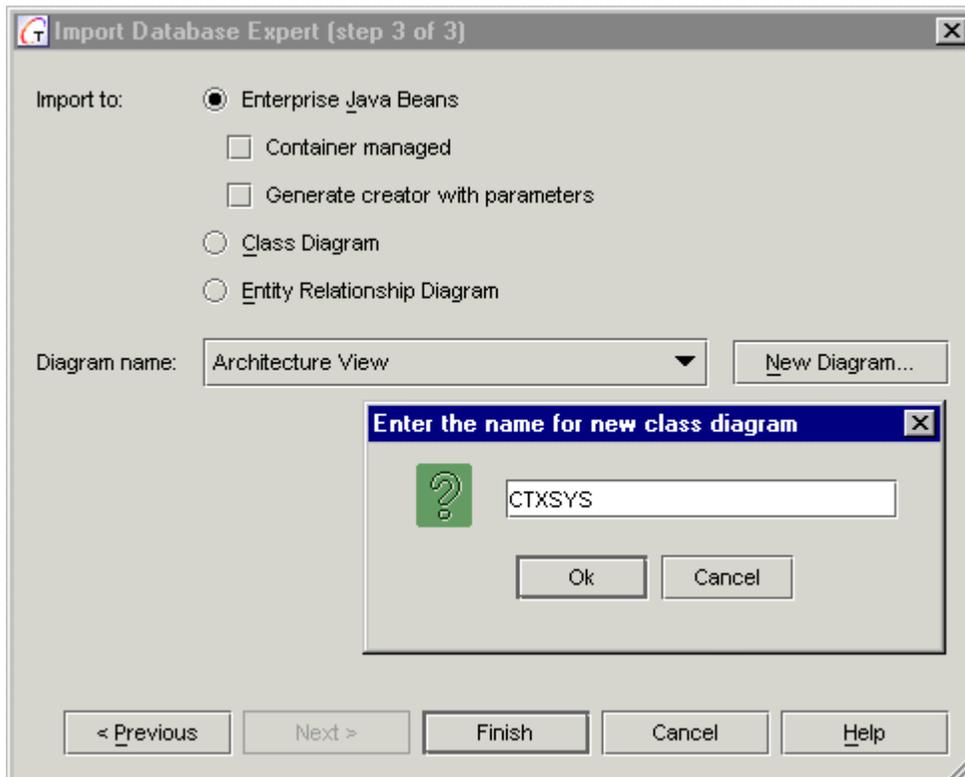
The source list (*Available Tables*) contain the tables that correspond to the selected schema. Use *Add / Remove* buttons to select tables to be imported. These tables are added to or removed from the target list (*Tables for Import*).

Spaces in the names are converted into underscores.

Note that since MS Access and MySQL don't support schemas, *Schema* option is disabled for MS Access and MySQL. The target list for import should not be empty.

Press *Next* to continue.

3rd Page



The last page is intended to define the diagram structure.

Import to

Choose one of the available diagram formats: Enterprise Java Beans, Class diagram or ER diagram.

Enterprise Java Beans: If selected, relationships in the RDB are mapped to associations between the supplier Bean class and the client Remote interface. Identifying relationship is presented by two associations:

- Client Remote interface is associated by both the supplier Bean and Primary Key classes.
- Similarly, the associations between the Beans (and, maybe, Primary Keys) and Remote interfaces transform to non-identifying (identifying, respectively) relationships.
- Other types of associations between EJBs are not supported.

Note: EJBs are created according to EJB specification patterns, which you can define in the project Options dialog (EJB page)

Container managed: if this checkbox is cleared, the target EJB is Bean managed, otherwise Container managed.

Generate creator with parameters: If this option is selected, `ejbCreate()` method is created. All columns of the table in question are included as the parameters of this method. For example, if there are fields CHAR col1 and INT col2 in the source table, then the resulting method will have the signature `ejbCreate(String col1, int col2)`.

Class Diagram: If selected, the import process recreates the database's structure in terms of classes with attributes whose types are Java equivalents to the database's column types (result looks similar to an ER diagram).

Note that the results of importing to the Class diagram depends on the target language. If the target language is Java, a constructor is created by default for each class that represents a table.

If the target language is C++, both constructor and destructor are created by default for each class that represents a table.

Entity Relationship Diagram: If selected, import creates high level data model that shows the major entities and relationships which support a wide business area.

Diagram name

In this choice field you can select a diagram of the appropriate type among the currently opened diagrams (for example, if you choose the "Entity Relationship Diagram" as the "Import to" option value, the "Diagram name" field will contain only currently opened ER diagrams). The database model will be created in the selected diagram.

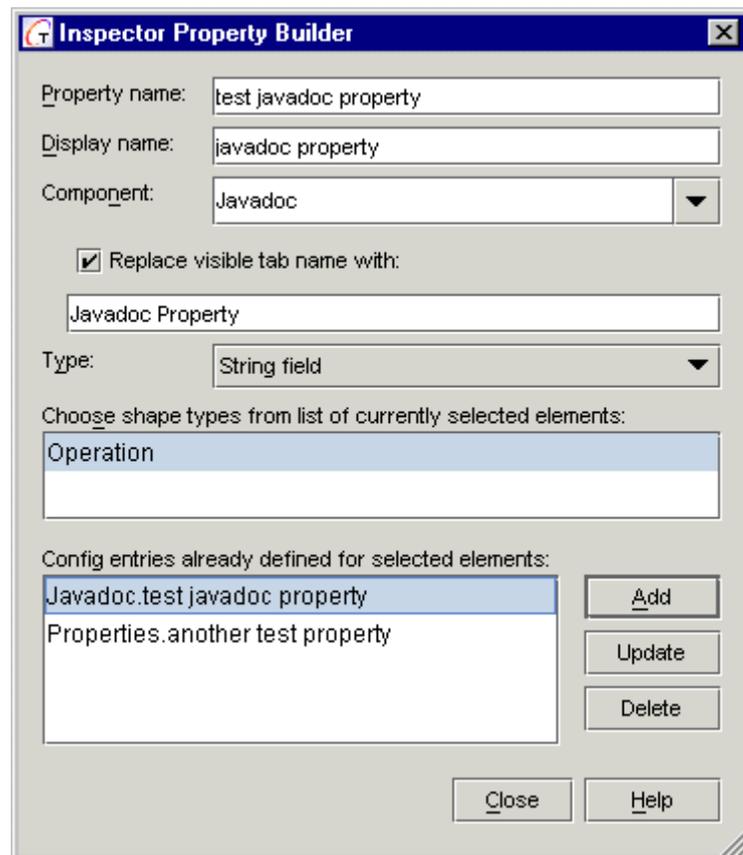
You can also use the *New Diagram* button to create a new diagram of the specified type "on the fly".

Click *Finish* to launch Import.

Inspector Property Builder

This dialog is invoked by Inspector Property Builder command from the Tools menu, under the pre-condition that Support User-Defined Inspector flag is set in the General tab of the Options dialog. The dialog window is non-modal and allows to navigate through the Model tab of the Explorer and the Diagram pane. Note that if you check this options, Inspector Property Builder command becomes available after Together's restart only.

The dialog allows to create or modify property tabbed pages in the inspector. New tab adds to the object inspectors of the selected element types, which is reflected in the changes .config file under %TGH%\config directory.



Fields

Property name: Enter the name of the property to be added.

Display name: Enter the name that will show up in the new tab of the inspector.

Component: Select inspector component for the new property from the drop-down list of available components (Properties, View etc.).

Replace visible tab name with: check this flag to assign the desired name for the new tabbed page of the inspector. When checked, the text field below is enabled.

Type. Select type of the property field from the drop-down list (boolean, String etc.).

Choose shapetypes from the list of currently selected elements: As you navigate through the diagram or a model tree and perform multiple selection (using CTRL or SHIFT keys), appropriate items add up to the list of selected elements. To create a new property, choose the desired shapetype.

Buttons

Add. Adds new property with the specified parameters to the inspector. The newly created tabbed page in the inspector corresponds to the selected component and provides all appropriate controls.

Update. You can change a property from the config entries' list. *Update* button modifies the inspector with the values specified for the selected config entry.

Delete. Removes selected config entry from the list.

Note: All changes immediately show up in the inspector.

Close. Saves changes and closes the dialog.

Help. Displays this topic.

J2EE Deployment Expert

This topic provides detailed description of the server-specific pages for the most popular application server platforms: the IBM WebSphere 3.5 and the BEA Weblogic 5.1, 6.0.

Working with the Expert, you have to describe what's to be done. Each page of the *Expert* contains a number of tasks. After you have entered all necessary fields and clicked *Finish*, the Expert starts executing the selected tasks. This topic provides explanations of the available tasks. The tasks (with some exceptions) can be executed separately.

J2EE Deployment Expert for IBM WebSphere 3.5

Process options

J2EE Deployment Expert

1 Select the application server platform for your deployment process from the list of supported servers

IBM WebSphere AE 3.5

2 You can optionally verify/correct EJBs, compile classes, generate Deployment Descriptors, and delete temporary classes before/after deployment. For certain servers "hot" deployment is available.

Process options

- Add libraries required for deployment to the current project's Search/Classpath
- Verify and/or correct EJBs for the selected server
- Compile the classes referenced in the currently selected diagram
- Generate server-specific Deployment Descriptor(s)
- Open XML editor for the generated Deployment Descriptor(s)
- Pack modules for deployment
- Hot Deploy to server
- Generate executable files for the client
- Generate a simple JSP client
- Process Servlet(s)
- Generate a command line file with instructions for deployment
- Clear temporary folder before starting the deployment process
- Clear temporary folder after deployment is complete

< Previous Next > Finish Cancel Help

Add libraries required for deployment to the project's Search/Classpath

J2EE deployment requires a number of server specific libraries. Since diverse servers use specific versions of J2EE specification, they generally suggest their own libraries that cover J2EE specification for the particular server usage. Therefore, a server specific library is normally added to the Classpath instead of Sun J2EE. Check this box to compile and deploy EJB or client program, and to make sure that you have all required libraries on your Classpath.

Verify and/or correct EJBs for selected server

This task verifies that the classes on the current diagram correspond to EJB specification supported by the destination server. Moreover, some servers have individual specification features. The primary objective of this task is to make sure that all the required classes exist on the diagram.

Note: The current diagram is used as the source of beans. Therefore, you have to open the appropriate class or Assembler Diagram with one or more EJB's.

Compile the classes referenced in the currently selected diagram

Compiles the bean and packs it to a JAR file. The new archive is stored in a temporary folder. Standard Sun tools (`javac.exe` and `jar.exe`) are used for compilation and packing.

Generate server-specific Deployment Descriptor(s)

Check this box to create deployment descriptors.

Open XML Editor for the generated Deployment Descriptor(s)

This option opens XML Editor, where you can edit the generated deployment descriptor. Two additional descriptors are generated, one for each CMP entity bean. As soon as the descriptors are ready, they are added to the JAR archive together with some additional classes. The WebLogic `ejbc.exe` tool is responsible for these operations, which results in a JAR file ready for deployment.

Pack modules for Deployment

Creates a deployable JAR.

Hot Deploy to server

If you check this box, deployment will be executed immediately. WebSphere must already be started. This step makes use of a bean that has already been compiled and jarred.

Note: As a rule, it is strongly advised to compile classes, generate descriptors and deploy the beans all together, because some tasks use the results of the previous tasks. For example, the Hot Deploy task requires a `*.jar` file to be prepared in a temporary folder, and moves this file to the `deployableEJBs` folder under WebSphere installation.

Generate executable files for the client

Together and WebSphere use different versions of JDK. Strange as it may seem, WebSphere requires using IBM JDK for the client programs. Thus, we cannot make use of Together's internal tools and have no choice but to generate two batch files, one of them being destined to compile, and the other one to run the client program. Subsequently, you can start them separately from Together.

Generate a simple JSP client

Check this box to generate a JSP client for the Enterprise Beans. The client is a set of interrelated JSP and HTML files, which can be viewed in any Internet browser. The intention of this client is to demonstrate access to remote EJB objects through an open interface. The semantic information required for JSP client generation is taken from the

active diagram of the current project. Due to the lack of information, it is impossible to create a client program ready for commercial use. That's why JSP client can be regarded as a universal testing facility.

It is possible to generate a JSP client from Web Application and Enterprise Application diagrams.

Process Servlet(s)

Check this box to deploy Servlets.

Generate a command line file with instructions for deployment

Check this option to generate a command file. The deployment process involves a sequence of calls to various console tools. This sequence can be generated and stored in a command file. Further, this file can be launched to perform all actions required for deployment.

Clear temporary folder before starting the deployment process / Clear temporary folder after deployment is complete

You can skip these options if you want to reuse temporary files.

"Common Properties" page

Name	Value
Temporary folder	c:\temp
WebSphere server home directory	c:\WebSphere\AppServer
Relative target path for generated JAR file	deployableEJBs
IBM JDK root directory	c:\WebSphere\AppServer\jdk
Prefix of the deployable JAR file	DeployableTogetherEJB
Display messages from tools	<input checked="" type="checkbox"/>

The fields on this page are quite self-explanatory. However, this section provides brief descriptions for some of them .

Relative target path for generated JAR file

Specify the location where the resulting JAR file will be stored and from where it will be loaded by WebSphere.

Prefix of deployable JAR file

This name is assigned to the JAR file (or files) generated in the *Compile Classes* task from the selected diagram. While preparing for deployment, an additional JAR file is created, whose name is based on the prefix assigned here.

"EJB Deployment Properties" page

Name	Value
Admin Node Name	docnew1
Dependent Classpath	d:\WebSphere\AppServer\lib\ibm...
Application Server Name	Default Server
Application Server Command Line Arguments	
Application Server Classpath	
EJB Container Name	Default Container
User ID (only for CMP EJBs)	together
Password (only for CMP EJBs)	*****
Create Table for CMP EJBs	<input type="checkbox"/>
Port number of the WebSphere node	900
Target WebSphere server directory	d:\WebSphere\AppServer 
Launch server in debug mode	<input type="checkbox"/>
Debug port number	8787
Generate WLM JAR for Beans	<input type="checkbox"/>
Stop server and remove existing Beans/Servlets before deployment	<input checked="" type="checkbox"/>
Start Beans/Servlets after deployment	<input checked="" type="checkbox"/>

This page provides entries for the Hot Deploy task and some other properties.

Admin Node Name

By default this name corresponds to the local computer name and is assigned automatically.

Dependent Classpath

Contents of this field should be the path to source classes, service libraries jars, etc.

Application Server Name

A single WebSphere node can hold several virtual application servers. By default, only one is allocated. Its name is Default Server. The same name is used by default in Together's J2EE Deployment Expert. If you are going to use a different application server, specify its name in this field. If the server with the specified name does not exist, it will be created.

Application Server Command Line Arguments, Application Server Classpath

Contents of these fields should correspond to the appropriate properties of the specific application server at the WebSphere.

EJB Container Name

A single WebSphere virtual Application Server can hold several EJB containers. By default, only one is allocated. Its name is Default Container. The same name is used by default in Together's J2EE Deployment Expert. If you are going to use a different container, specify its name in this field. If the container with the specified name does not exist, it will be created.

User ID / Password (only for CMP beans)

These fields are necessary to maintain container managed persistence. You can use the user name and password assigned during WebSphere installation.

Create Table for CMP EJBs

When checked, this task creates new tables in the server persistence repository. These tables are destined to support Container Managed Persistence. The structure of the tables corresponds to the beans being deployed. This task pertains to CMP beans only.

Port number of the WebSphere node

This port is used for communication between the application server and client program. It is assumed that the host is `localhost`. Before proceeding with the next page of the J2EE Deployment Expert, make sure that the server is available for the selected port and host.

Target WebSphere server directory

Specify the path to the WebSphere application server product installation root.

Launch server in debug mode

If you are going to debug your beans, you can run the server in the debug mode and attach to it from Together's debugger process. Actually, this option does not restart the server in the debug mode, but adds special command line arguments to the application. After that you have to restart the application server from the WebSphere Administrative Console.

Note: A number of additional libraries are required to launch WebSphere in the debug mode. See WebSphere InfoCenter documentation for details.

Debug port number

Enter the port number for communication between WebSphere and the external debugger. The server and debugger must use the same port number. (Good advice: keep this number in a safe place.) You can use the default port number 8787.

Generate WLM Jar for Beans

Check to create Work Load Management file.

"Servlet Deployment Properties" page

You can see this page only if *Process Servlet(s)* was checked. You can use this page for creating servlets and WebSphere service servlets.

Name	Value
Virtual Host name	default_host
Servlet Engine name	Default Servlet Engine
Web Application name	default_app
Web Application Document root	
Relative path to Servlet	/
Relative Web Path to Servlet directory	/servlet
Stop server before re-deploying Servlets	<input type="checkbox"/>
Create "JSP 1.0" Support Servlet	<input type="checkbox"/>
Create the serving Servlet "File"	<input type="checkbox"/>
Load Servlets at startup	<input checked="" type="checkbox"/>
Enable Servlets	<input checked="" type="checkbox"/>
Enable Debug	<input type="checkbox"/>

Note: In order to access any WEB resource we have to specify the URL. A standard Web URL includes protocol, host, port, web resource alias, and path. The path to a resource is implied relative to the document root. It consists of directories and a file name at the end.

Virtual Host name

A single WebSphere Application Server can hold and serve several virtual Web hosts. Each virtual host is associated with a set of aliases. By default, only one host is allocated. Its name is `default_host`. It is associated with the `localhost`. See *Chapter 0.16: What are virtual hosts?*, *WebSphere InfoCenter documentation* for details.

The `default_host` is used in the J2EE Deployment Expert by default. If you want to use a different virtual host, specify its correct name here.

Servlet Engine name

Each WebSphere virtual Application Server can hold several Servlet Engines. By default, only one engine is allocated. Its name is *Default Servlet Engine*. The same name is used in the J2EE Deployment Expert by default.

If you are going to use a different *Servlet Engine*, specify its name in this field. If the engine with the specified name doesn't exist, it will be created.

Web Application name

Each WebSphere Servlet Engine can hold several Web Applications. After the server installation there are several web applications. J2EE Deployment Expert uses the `default_app` web application.

If you are going to use a different web application, specify its name in this field. If the application with the specified name doesn't exist, it will be created.

Web Application Document root

Each Web application on the WebSphere server has its own *Document root*. *Document Root* is a file system address accessible through an Internet browser. All subdirectories of the root are also accessible. Contents of this field should be equal to the corresponding property of the *Web Application* in the *WebSphere Advanced Administrative Console*.

Relative path to Servlet

Each Web Application on the WebSphere server has its own *Web Path*. It is a string similar to a directory path, which however doesn't refer to any actual directory. This is a Web resource alias for the Web application. This *Web path* is appended to the virtual host alias. For example, if the virtual host is `localhost:900` and the *Web path* is `/webapp/Examples`, then the result will be `http://localhost:900/webapp/Examples`. If *Web Application* is empty, the *Web path* consists of a single slash character `"/`.

Relative Web Path to Servlet directory

In this field you have to specify the part of *Web path* to access a servlet. This path is a string similar to the directory path, which however doesn't refer to any actual directory. It is appended to the *Relative path to Servlet* (see previous paragraph). For example, if the virtual host is `localhost:900`, *Web path* is `/webapp/Examples` and *Relative Web Path to Servlet* is `/servlet`, then the result will be `http://localhost:900/webapp/Examples/servlet`. If *Relative Web Path to Servlet directory* is empty, it consists of a single slash character `"/`.

Stop Server before re-deploying Servlets

Stopping or restarting a running servlet requires Application Server Node shutdown. However, this results in the loss of all deployed objects, including the EJB's. Check this flag if the running beans are of no interest, and restart the servlets together with the Application Server. If you preserve the running beans, the new servlets still deploy, but they do not stop or start, regardless of the previously set flags *Stop (remove) EJB's*, *Servlets*, and *Start EJB's*, *Servlets* on the page *EJB Deployment Properties*.

Create "JSP 1.0" Support Servlet

Check this flag if you are going to use JSP in your project. This flag indicates whether a special servlet will be added to the Web application. The name of this servlet class is `com.sun.jsp.runtime.JspServlet`. In WebSphere it is normally called `jsp10`. This servlet can be added either by checking the flag, or manually.

Create the serving Servlet "File"

Check this flag if you are going to use HTML resources in your project. This flag indicates whether a special servlet, responsible for processing requests to HTML resources, will be added to the Web application. The name of this servlet class is `com.ibm.servlet.engine.webapp.SimpleFileServlet`. In WebSphere it is normally called `file`. This servlet can be added either by checking the flag, or manually.

Load Servlets at Startup

This flag indicates whether to load a servlet when the application server starts, or to wait until the servlet is requested. By default it is set to `False`.

Enable Servlets

This flag indicates whether the servlet is available to handle requests. By default it is set to `True`.

Enable Debug

This flag indicates whether to start the servlet in Java debug mode. By default it is set to False.

"Client Properties" page

This page contains entries for the properties required to generate start and compile files for the client task (appropriate flag checked in the first page of the expert).

Name	Value
Main class of the client, including package names	Main
Path to the client sources	.
Path to the target directory	C:\Together5.5\out\magazine
Deployed JAR file name	deployableEJBs\DeploySession1 Bean.jar

Main class of the client, including package names

Enter the class name that contains the main method to run the client application. Automatically generated batch files will refer to this class.

Path to the client sources

Enter the fully qualified path to the directory where client source files are stored.

Path to the target directory

Enter the fully qualified path to the directory where compiled files will be placed.

"Simple JSP Client Generation" page

This page shows up if you have checked the appropriate flag on the first page of the Expert and contains entries for the properties required to generate a Simple JSP client. On this page you can enter properties for the *Generate a simple JSP client* task.

Name	Value
Root path for JSP storage	d:\WebSphere\AppServer\hosts\default...
Subdirectory for JSP files	myJSP
URL for browsing JSP	http://localhost/webapp/examples
WebSphere JNDI service provider URL	iiop://localhost:900
Show JSP client in default Internet browser	<input checked="" type="checkbox"/>

The fields are quite self-explanatory.

Root path for JSP storage

Specify the root of file hierarchy, which can be accessed from outside. Thus, if you want to allow access to some HTML or JSP files, you have to place these files under the document root.

Subdirectory for JSP files

To avoid disorder in the public directory it is advisable to group the files in a subdirectory. Enter the path to the subdirectory where your files will be stored. This path is relative to the root public directory. If the subdirectory does not exist, it is created automatically (after confirmation).

URL for browsing JSP

In order to browse a remote WEB resource, we have to set its URL. A standard web URL includes protocol, host, port, web resource alias, and path. Specify all parts of the URL, except for the path. During request processing, the server substitutes this base URL to the WebSphere public directory in order to access the requested web resource. An invalid value in this field or the WebSphere public directory causes access error. By default, the base URL is `http://localhost`.

WebSphere JNDI service provider URL

This field is required to establish connection between JSP client and EJB server. Usually, the client should first send a request to the name service provider in order to get access to an instance of EJB. The value entered in this field is inserted into a JSP file during the course of generation and is subsequently used during execution. Normally, you do not need to change this field. The default value is `iiop://localhost:900`.

Show JSP client in default Internet browser

Check this box to launch your default Internet browser and display the start page.

"Command Line File with Instructions for Deployment" page

This page contains options for creating the command line file.

Name	Value
Generate JAR file	<input checked="" type="checkbox"/>
Stop and remove Beans, Servlets before deployment	<input checked="" type="checkbox"/>
Hot Deploy to server	<input checked="" type="checkbox"/>
Start Beans, Servlets after deployment	<input checked="" type="checkbox"/>
Command line file name	WAS35DeployCmd.bat

The fields are quite self explanatory and allow to define the command line file's contents and properties.

"Verify/Correct Sources" page

This page contains entries for the Verify/Correct task. The fields are quite self explanatory and allow to terminate deployment if verification fails, correct sources to make them compliant with WAS 3.5, make backup copies and change return types in the created class.

Name	Value
Stop deployment if verification fails	<input type="checkbox"/>
Correct sources to comply with WebSphere 3.5 (specification 1.0)	<input checked="" type="checkbox"/>
Back up original sources	<input checked="" type="checkbox"/>
Backup location	c:\temp\backup

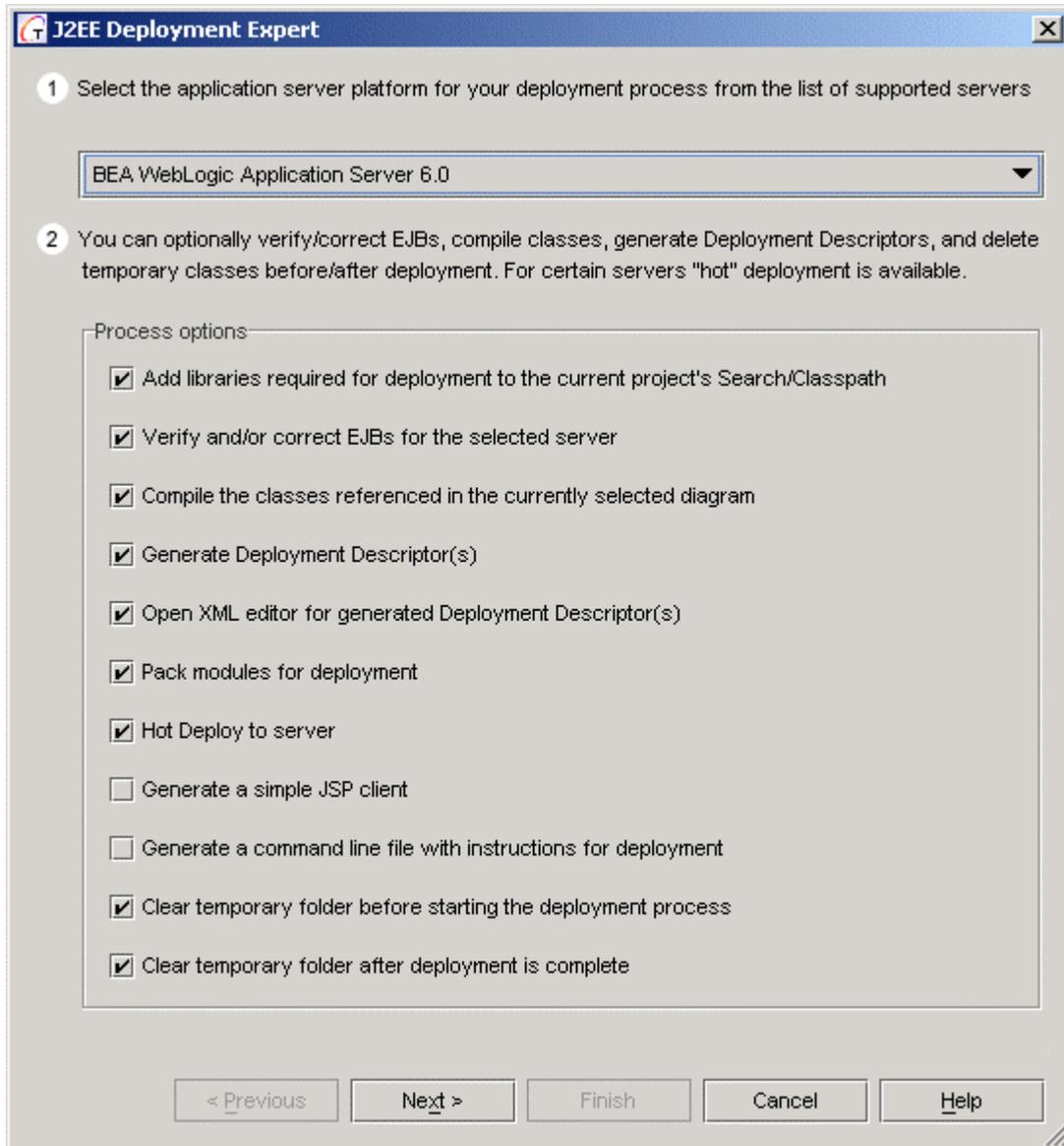
Correct sources to comply with WebSphere 3.5 (specification 1.0)

WebSphere Application Server version 3.5 supports EJB specification 1.0. This is not the latest EJB specification. Later specifications require that create methods of home interface should return `primary key` type, while the 1.0 specification requires that create method returns `void`. For the sake of compliance with EJB 1.0 specification, the correction program replaces all changed types with `void`.

J2EE Deployment Expert for BEA WebLogic Application Server

Process options

Select Weblogic Application Server (4.5.1, 5.1, or 6.0)



Add libraries required for deployment to the project's Search/Classpath

EJB deployment requires a number of server specific libraries. Since diverse servers use specific versions of J2EE specification, they generally suggest their own libraries that cover J2EE specification for the particular server usage. Therefore, server specific library is normally added to the Classpath instead of Sun J2EE. Check this box to compile and deploy EJB or client program, and to make sure that you have all required libraries on your Classpath.

Verify and/or correct EJB source for the selected server

This task verifies that the classes on the current diagram correspond to EJB specification supported by the destination server. Moreover, some servers have individual specification features. The primary objective of this task is to make sure that all the required classes exist on the diagram.

Note: The current diagram is used as the source of beans. Therefore, you have to open appropriate class or assembler diagram.

Compile the classes referenced in the currently selected diagram

Compiles the bean and packs it to a JAR file. New archive is stored in a temporary folder. Standard Sun tool (`javac.exe` and `jar.exe`) is used for compilation and packing.

Generate Deployment Descriptor(s)

Generates deployment descriptor. Two additional descriptors are generated, one for each CMP entity bean. As soon as the descriptors are ready, they are added to the JAR archive together with some additional classes. `WebLogic ejbc.exe` tool is responsible for these operations, which results in a JAR file ready for deployment.

Open XML Editor for the generated Deployment Descriptor(s)

Opens XML Editor, where you can edit the generated Deployment Descriptor.

Pack Modules for Deployment

Creates deployable JAR, WAR, or EAR.

Hot Deploy to server

Check this box to perform deployment. This step makes use of a bean already compiled and jarred. Therefore you have to use this step in conjunction with the others.

Warning: Be aware that all hot deployed beans will be lost after WebLogic termination. You will have to re-deploy necessary beans again. In order to deploy your beans permanently, you have to add certain lines manually:

- to `weblogic.property` file for Weblogic 5.1
- to `config.xml` file or Weblogic 6.0

Note: As a rule, it is strongly advised to compile classes, generate descriptors and deploy the beans all together, because some tasks use the results of the previous tasks. For example, Hot Deploy task requires `*.jar` file to be prepared in a temporary folder.

Generate a simple JSP client

Check this box to generate a JSP client for the Enterprise Beans. Client is a set of interrelated JSP and HTML files, which can be viewed in any Internet browser. The intention of this client is to demonstrate access to remote EJB objects through the opened interfaces. The semantic information required for JSP client generation is taken from the active diagram of the current project. Due to the lack of information, it is impossible to create a client program ready for commercial use. That's why JSP client can be regarded as a universal testing facility.

It is possible to generate a JSP client from *Web Application* and *Enterprise Application diagrams*.

Generate a command line file with instructions for deployment

Generates a command file for deployment.

Clear temporary folder before starting the deployment process / Clear temporary folder after deployment is complete

Skip these tasks if you want to reuse temporary files.

Page "Common properties"

Name	Value
Folder for the temporary files	c:\temp
Folder for the generated JAR file	c:\temp
JDK 1.3 root directory	c:\bea\jdk130
WebLogic Server 6.0 root directory	c:\bea\wlserver6.0

The fields on this page are quite self-explanatory. However, this section provides brief descriptions for some of fields.

Note: Set paths to jdk1.2 or higher for BEA Weblogic 5.1, and to jdk1.3 for BEA Weblogic 6.0. It is impossible to use jdk1.2 for BEA Weblogic 6.0.

Folder for the generated JAR file

Specify location where the resulting JAR file will be stored and from where it will be loaded by WebLogic.

Page "Verify/Correct sources"

Name	Value
Stop deployment if verification fails	<input checked="" type="checkbox"/>
Correct sources to comply with W...	<input checked="" type="checkbox"/>
Backup original sources	<input checked="" type="checkbox"/>
Backup folder (if "correct sources...	c:\temp\backup

On this page you can select whether deployment will stop if verification fails, correct the source code to comply with the requirements of WebLogic 5.1 specification, and create backup copy of the source code... just in case.

Page "Run-time deploy"

This page enables assigning properties required for the Hot deploy task.

Name	Value
Pack modules for deployment	<input checked="" type="checkbox"/>
Hot Deploy to BEA WebLogic 6.1	<input checked="" type="checkbox"/>
Short command line file name	DeployWLS61Cmd.bat
System password	*****
Server port	7001
Server host	localhost

System password

System password is defined in course of the WebLogic installation.

Server port

Default server port is 7.3X. If you don't know the server port number, you can find it:

- in `weblogic.property` file for Weblogic 5.1,
- in `config.xml` file for Weblogic 6.0.

Server host

As a rule, the server host is assigned to `localhost`, since this deployment expert uses the file system access to WebLogic server, and therefore the server runs on the same computer.

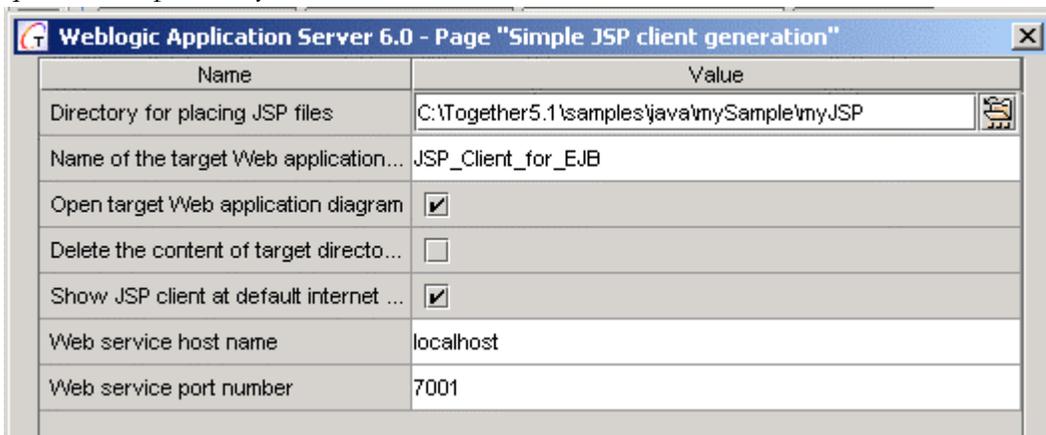
Update already deployed JAR

During Run-time deployment to Weblogic x.x, it is possible to update the modules, which have already been deployed. To do that, check the flag *Update already deployed JAR*. Note that this facility is not available for the other application servers.

Before proceeding to the next page, make sure that the server is available at the selected port and host.

Page "Simple JSP client generation"

This page shows up if you have checked appropriate flag on the first page of the Expert and contains entries for the properties required to generate a Simple JSP client. The fields are quite self-explanatory.



Name	Value
Directory for placing JSP files	C:\Together5.1\samples\java\mySample\myJSP
Name of the target Web application...	JSP_Client_for_EJB
Open target Web application diagram	<input checked="" type="checkbox"/>
Delete the content of target directo...	<input type="checkbox"/>
Show JSP client at default internet ...	<input checked="" type="checkbox"/>
Web service host name	localhost
Web service port number	7001

Weblogic 5.1 public directory

This directory is the root of the file hierarchy, which can be accessed from outside. Thus, if you want to allow access to some HTML of JSP files, you have to place them here. By default, Weblogic public directory resides at `Weblogic home] /myserver/public_html`.

Subdirectory for placing JSP files

To avoid disorder in the public directory it is advisable to group the files in a subdirectory. Enter path to the subdirectory where your files will be stored. This path is relative to the root public directory. If the subdirectory does not exist, it is created automatically (after confirmation).

Base URL for browsing JSP

In order to browse a remote WEB resource, we have to set its URL. Standard web URL includes protocol, host, port, web resource alias, and path. Specify all parts of the URL, except for the path. During request processing, the server substitutes this base URL to

WebLogic public directory in order to access the requested web resource. Wrong value of this field or WebLogic public directory causes access error. By default base URL is `http://localhost:7001/`.

Show result after generation

Check this box to launch your default Internet browser and display the start page.

Run JSP client under Tomcat in debug mode

If this flag is checked, the generated JSP client will run under Tomcat, rather than WebLogic server. In this case, JSPs interact with an EJB remotely. For this purpose, the source code of these JSP contains lines for initialization of the `Initialling` object. The initialization text is stored in a configuration file, and can be modified if necessary.

Local root directory for placing JSP files (for Tomcat)

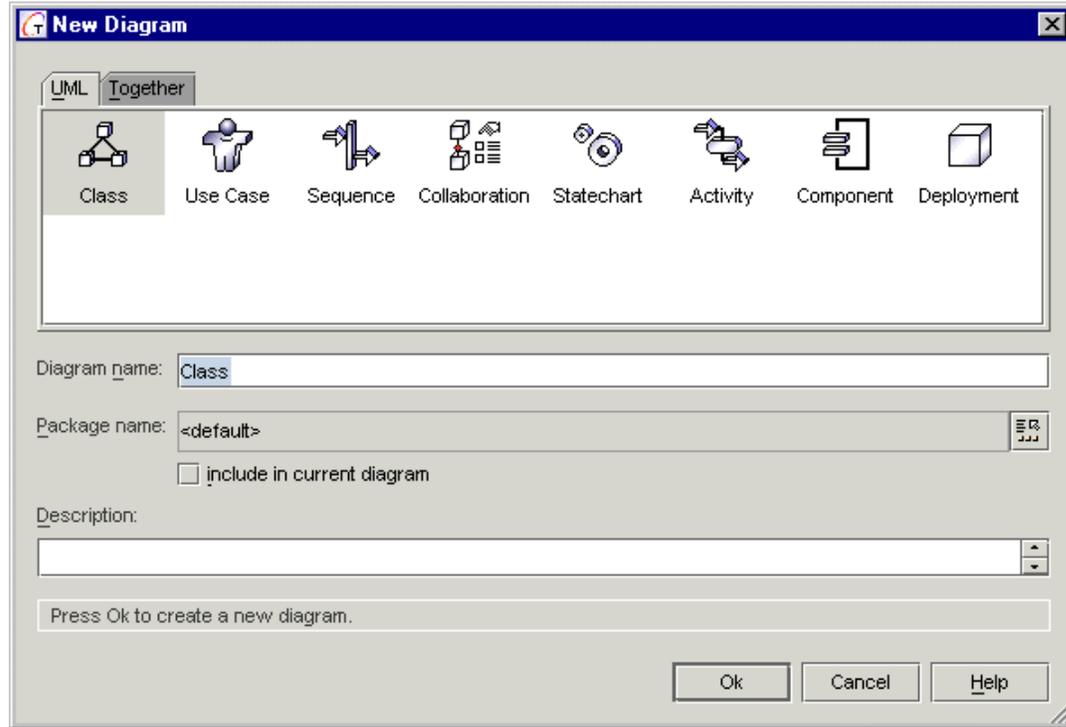
This field defines allocation of the generated JSP files for running under Tomcat. The folder specified in this field is alternative with regards to the one in *Weblogic public directory*. By default the root directory of the current project is assumed.

WebLogic JNDI service provider URL (for Tomcat)

This parameter is quite self-explanatory. The value of the URL is used for creating parameters for the `InitialContext` object. This helps avoid hacking the configuration file, if the Weblogic server is remote, or uses different port number.

If debug mode is selected, the *Navigation Page* of the JSP client runs automatically. In this case, *Show result after generation* flag is ignored.

New Diagram Dialog



Type

Lists the supported diagram types. Click on one of the tabs and select the type of diagram you want to create in the project.

Name

By default your selected diagram type is displayed in the Diagram name edit box: you may edit it or enter a new name for the new diagram. This name displays in the Model tab of Explorer and in the diagram's tab in the Diagram pane when the diagram is open for editing. by default .

Package

Specify the package in which the new diagram should reside. Default value is the package currently open in the Explorer, or <default> if no package is opened. You can select a different package by clicking the browse button to the right of the Package field to launch an explorer dialog where you navigate to the desired package and select it as the destination.

Include in current diagram

If checked, a logical link to the new diagram will be created and displayed on the diagram currently selected in the Diagram pane (if any). The link is displayed as a package and carries the icon representing the diagram type.

Description

Optionally enter any free-form text. This text will display in the Description field of diagram properties when the diagram is open for editing.

Buttons

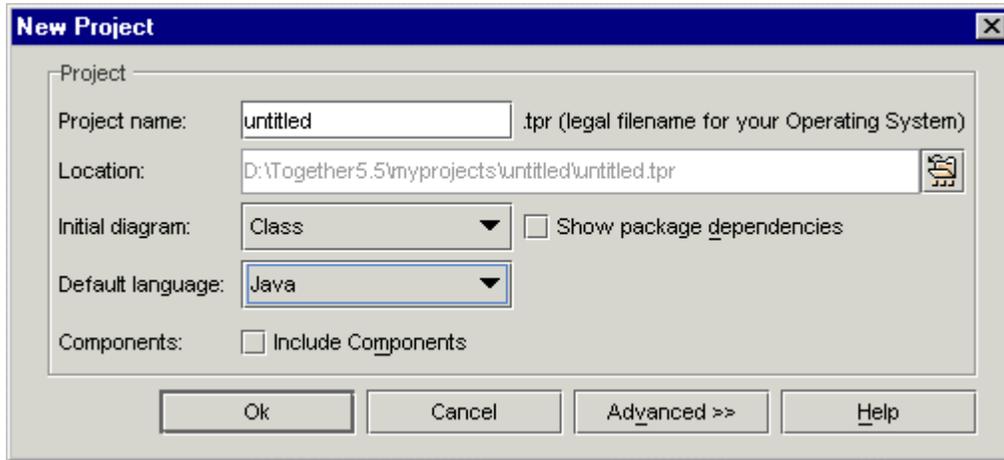
OK: Creates the new diagram of the selected type in the specified package, opens it in the Diagram pane in a new tab, and closes the New Diagram dialog.

Cancel: Closes the New Diagram dialog abandoning any changes.

Help: displays this Help topic.

New Project - Project Properties dialog

Default mode



Project

Project name

Enter a name for the new project. It will be used as the filename when *Together* creates a project file in the operating system. (**Note:** project file has a .tpr extension).

Location

A read-only field that displays the full path specification for the location where *Together* will create the new project file and related properties files. The path specified here is the *project root directory*. Alternatively, you can select the project root directory using the path chooser button. In the *Project Properties* dialog this button is disabled.

Initial diagram

Displayed only in New Project dialog. Pick the type of diagram you want to create in the project root directory when the new the project is created.

Default value is Class (diagram). For free or feature-limited editions, Class is the only diagram type that will work. You can specify *none* (i.e. don't create any diagram initially).

Default language

Pick the default target programming language for round-trip engineering of this project.

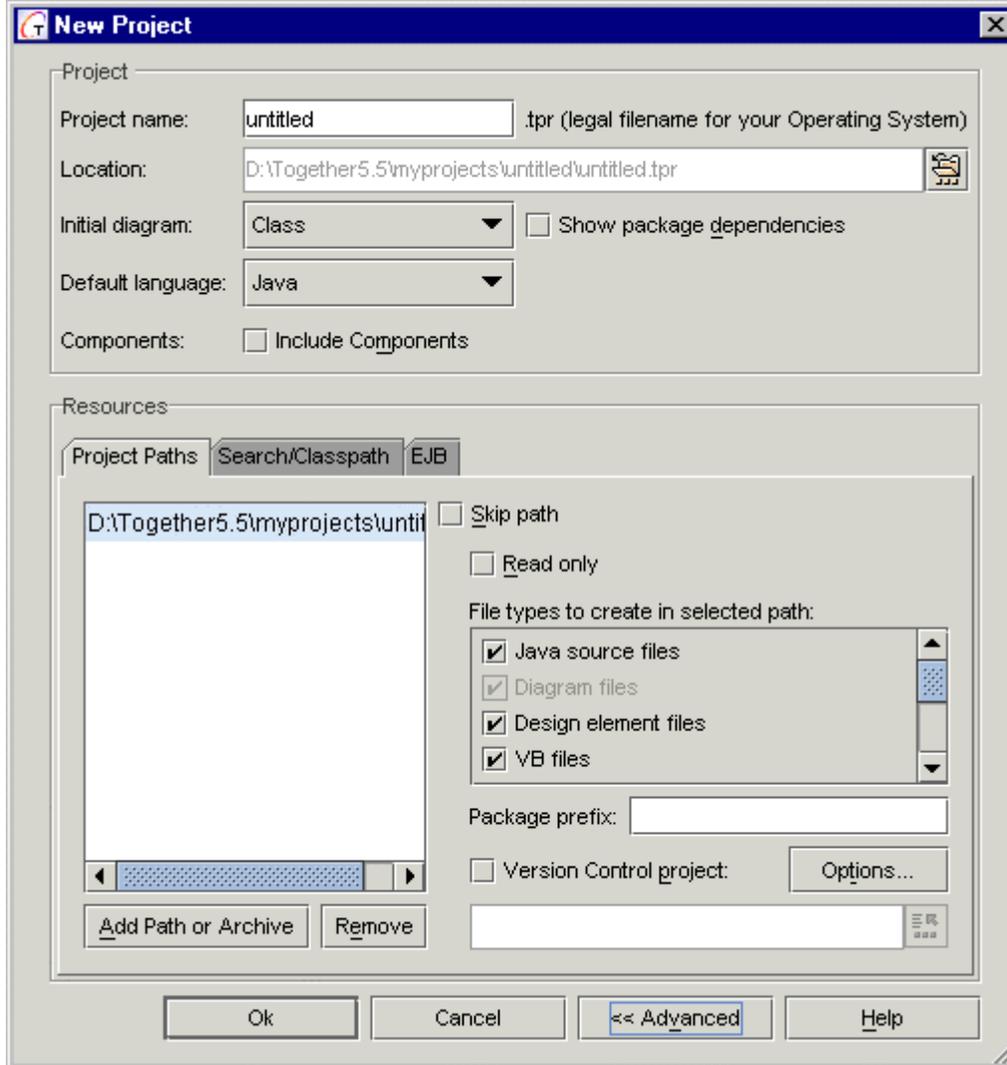
Show package dependencies

Displayed only in New Project dialog. When checked, automatically draws links between packages.

Components

Check this box to include Coad Components into the project. Fully qualified path to the Coad Components appears in the Search/Classpath tab of the Resources pane.

Advanced mode



Resources

Advanced mode displays project resources and enables you to add or delete resources, control reading and writing of resources, and specify version control system project for the *Together* project. The dialog displays in this mode when called from File | Project Properties.

Project paths tab

Displays all the resources that pertain to the project. These may be either directories or archives containing classes or source files. The resources listed display in the Model Tab of the Explorer and are round-trip engineered according to your specifications on the other controls at right. The default settings initially apply to all resources in the list. You can select individual resources and modify the settings of the other controls.

File types

Specify which types of files are writable in the selected resource path or archive. Refer to the information about the supported languages for details.

Package prefix

Specify a value to be substituted for the full resource path in package statements.

Version control project

These controls are disabled unless you have enabled version control system support in Options | Default (or Project) | Version Control.

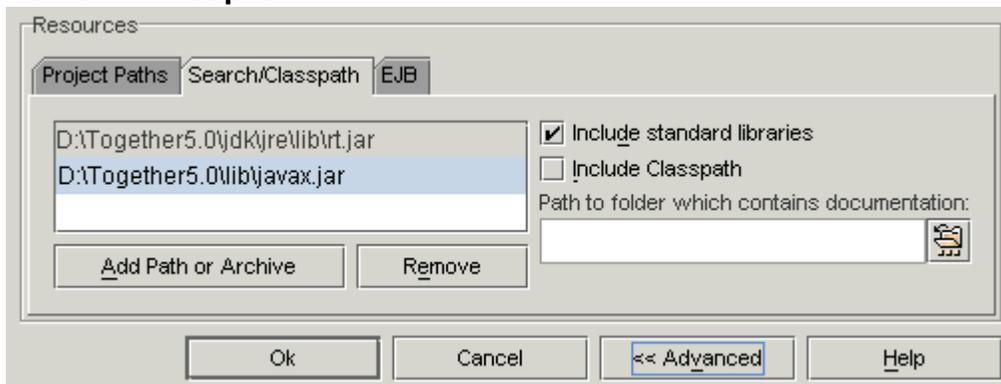
If version control is enabled, check this box to associate the new or current project with a project in your version control system (VCS). Clicking the Select button activates your VCS client and enables you to select an existing VCS project. *Together's* version control mechanisms will interact with this VCS project when the new or current *Together* project is open.

Read only

Select this option if you want *Together* to treat the selected resource directory or archive as read-only. When selected, the File Types list is disabled.

Skip path

Select this option if you want *Together* to ignore this resource when parsing the project structures. The resource is still a part of the project, but it is not visible in the Explorer's Model tab, not compiled and not included in the generated documentation.

Search / Classpath tab

Resource directories and archives in this list do not "belong" to the project and are not shown in the *Model* tab of the Explorer. They are *logically linked* to the project, and you can include their contents in diagrams using the *Add Shortcut* command of diagram speedmenus. In diagrams, Classes from these resources display a "shortcut" symbol on one corner. Logically linked resources cannot be modified from *Together*, and are not included in generated documentation. If you want such resources included, you must remove them from this list and add them to the Project Paths list.

By default the list contains the component directory of your *Together* installation and standard classes on the Classpath (if these are found in the environment).

Include standard classes

Check this option to include only standard classes from the Classpath in the list of logically linked resources available on search paths. The standard classes to be added are defined in the Options dialog | New Project | Include standard libraries.

Include Classpath

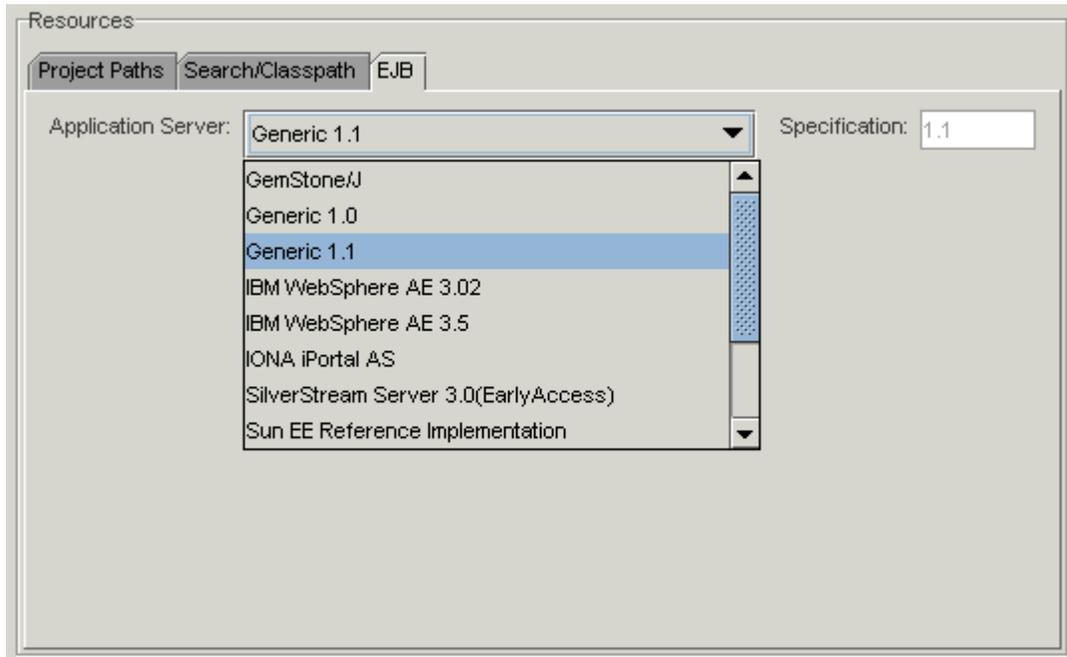
Check this option to include the full project Classpath in the list of logically linked resources available on search paths.

Path to folder which contains documentation

If an object from the search/classpath is imported into the project, you can view its previously generated documentation (choose Tools | Show HTML Documentation on the speedmenu).

Use this text area and file chooser button to specify location of the external project documentation folder.

EJB tab



Application server

This selection in the Application Server list represents the default target app server platform for Enterprise JavaBeans (EJBs) in the project. When you create EJBs, any server-specific elements for the classes/interfaces are included in the generated code. The EJB Specification version supported for the selected application server platform displays in the read-only field to the right.

If the project will not include EJBs, you can simply accept the default *Generic 1.1*. For a project that does include EJBs, you can optionally select an application server from the drop down list. Note that your selection here does not prevent you from deploying EJBs on another app server: the EJB Deployment Expert enables you to deploy to any supported server platform.

Buttons

Resource control

These buttons operate on the list of resources in all three tabs of the Resources section of the dialog. They are accessible in the Advanced mode only.

Add Path: Add a directory to the project. Select directory from standard OS file selection dialog.

Add Zip/JAR: Add an archive as a resource for the project. Select archive file from standard OS file selection dialog.

Remove: Remove the selected resource from the project.

Dialog control

OK: creates a new *Together* project in the location specified and opens the default diagram in the Diagram pane.

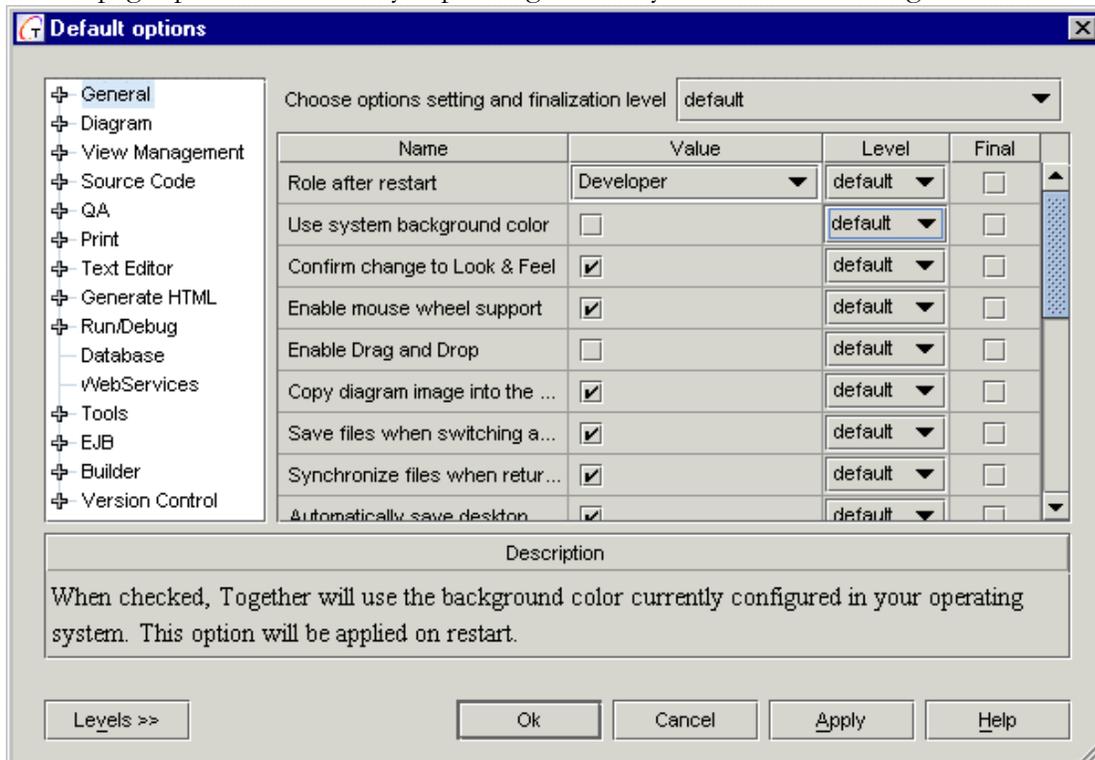
Cancel: closes the dialog abandoning any changes.

Advanced: toggles display of the Resources section of the dialog

Help: displays this Help topic.

Options dialog

The Options dialog presents several pages of configuration options. Editing these options enables you to customize the way Together works. The dialog title, and the number of tabbed pages presented can vary depending on how you invoke this dialog.



The full Options dialog for Default configuration level

Understanding the configuration level

Together's multi-level configuration feature enables you to set configuration options at different configuration *levels*, so it's important to know which level you are about to change before you begin editing any options. The level is selected automatically depending on whether or not a project is currently open, and/or the menu you use to invoke the dialog. (Multi-level configuration is described fully in the Together User's Guide.

If you are not sure which configuration level is currently active, click the Levels button in the Options dialog. The available levels display in the additional *Level* column, and the current level displays on top of the dialog window.

Context-sensitive Help for pages and options

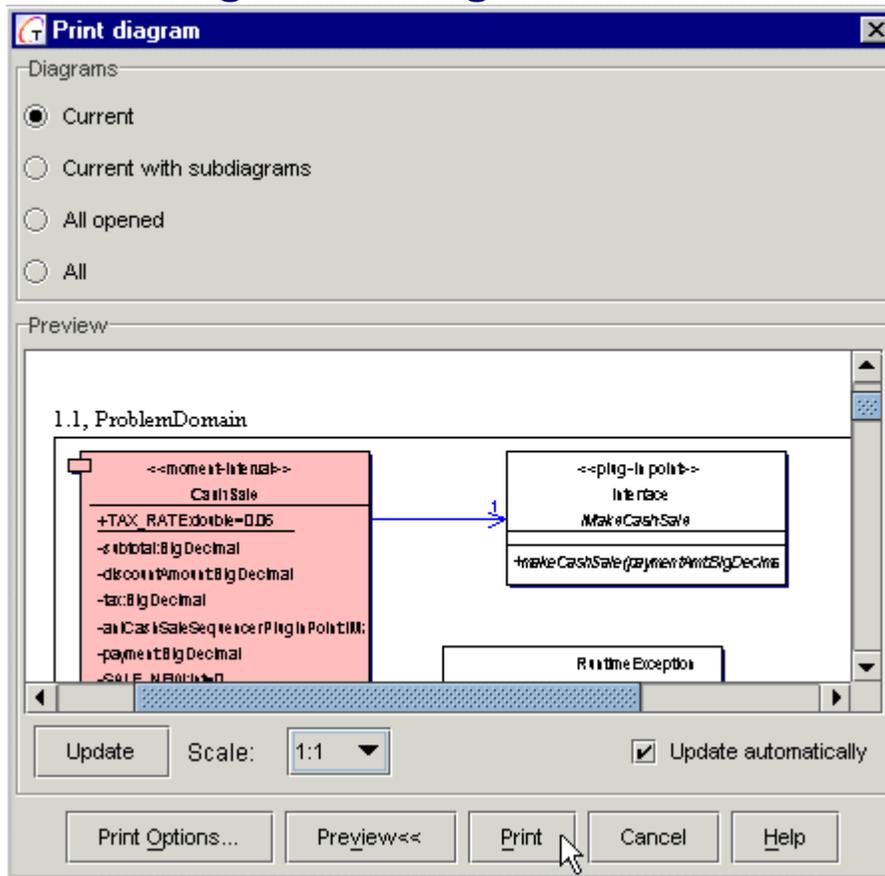
Help for each page tab and each individual option appears directly in the Options dialog rather than in the Together Help system. You don't need to leave your work and switch back and forth between applications... just click on a page tab or on an individual option to read its associated Help text.

Note: some top-level nodes of multi-valued options do not display Help text. This is intentional... each of the sub-fields should have Help.

See also

- Multi-level architecture overview
- Guide to the Options pages
- Using the Options Dialog
- Common Configuration Tasks

Print Diagram Dialog



Diagrams

This section contains radio buttons that allow selection of the printout scope.

Current

Prints only the currently selected diagram from the Diagram pane.

Current with subdiagrams

Prints the currently selected diagram from the Diagram pane plus any other diagrams that have been packaged into it (subdiagrams).

All opened

Prints all diagrams currently open for editing in the Diagram pane.

All

Prints all diagrams in the current project.

Buttons

Print Options: Launches the appropriate level Options dialog with Print options focused.

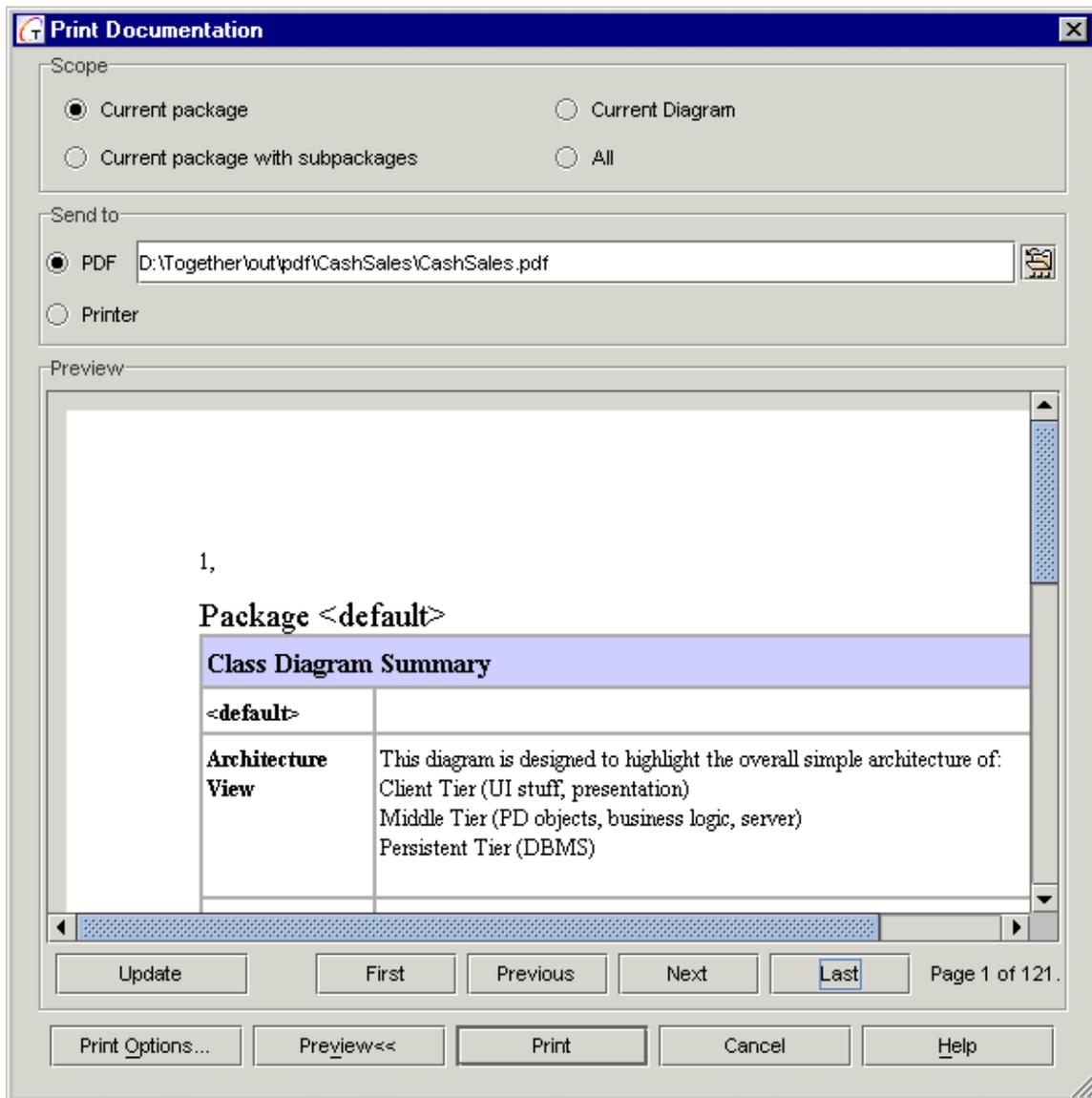
Preview: Dialog changes to *preview* mode (see figure above) and displays all diagrams selected for printing. In this mode, you can set the scale from 1:1 to 1:16.

Print: sends selected diagrams to the configured system printer.

Note: settings specified in the Print Options do not override default printer settings. If, for example, you set landscape format in the Print Options, you also have to make same change in the settings of your default printer. Besides that, it is important to confine the custom paper size to the size supported by the printer. Otherwise allocation of the printout will be in conflict with the paper format.

Cancel: Closes the dialog abandoning any changes.

Print Documentation Dialog



Scope

This section contains radio buttons that allows to select documentation for printing.

Current package: prints documentation for the currently selected package.

Current package with subpackages: prints documentation for the currently selected package and all its subpackages.

Current diagram: prints documentation for the currently selected diagram.

All: prints all generated documentation

Send to

You can choose the destination for documentation printout.

PDF: creates PDF file in the specified location.

Printer: prints hard copy on the default printer.

Buttons

Print Options: Launches the Options dialog at the Project level displaying only the *Print* page. You can change any of the configuration options before printing the file. Note that your changes will persist at the Project configuration level after the print job finishes. If you only want to change settings temporarily for the process, choose Options | Project | Print after the print job is finished and reset the options to their previous values.

Preview: Toggles preview mode in which a WYSIWYG preview of the printed output displays in the Print File dialog. (See Preview Mode section.)

Print: Sends the file to the selected destination.

Cancel: Closes the dialog without printing the file.

Preview mode

In preview mode the dialog displays the following additional controls:

Update: Refreshes the WYSIWYG view, after changes to the font size or line spacing.

First: Jumps the WYSIWYG view to the first page when another page is displayed.

Previous: Jumps the WYSIWYG view to the previous page.

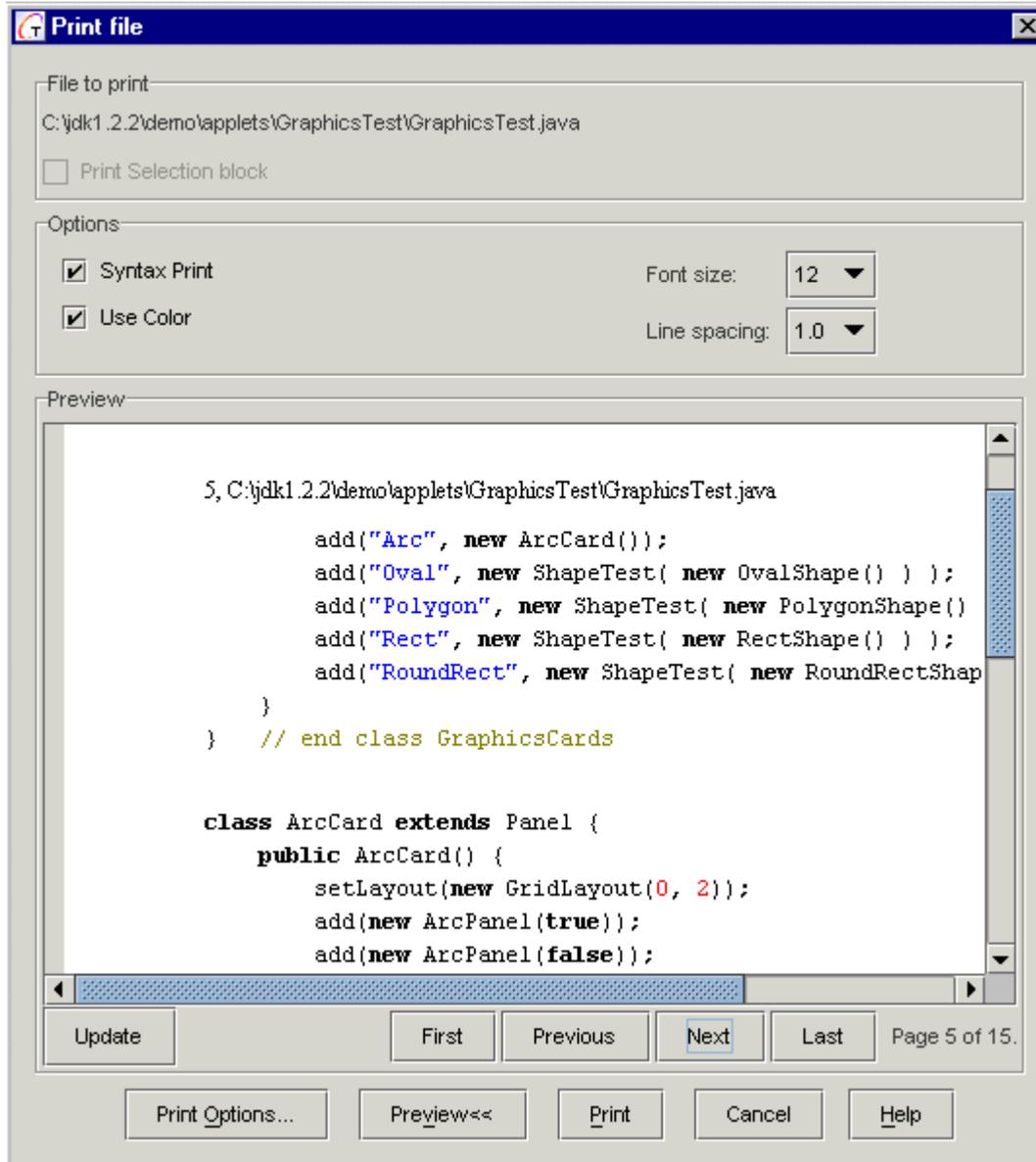
Next: Jumps the WYSIWYG view to the next page when there is more than one page to be printed.

Last: Jumps the WYSIWYG view to the last page when there is more than one page to be printed.

Page count: Read-only display of the number of pages in the print job

Tip: If the WYSIWYG view does not properly display color scheme and fonts, increase font size.

Print File dialog



File to Print

You can print a file when it is open in the Editor pane. The first line in this box displays the path and file name of the file that is about to print.

Print Selection Block: Enabled only when a block of text is selected. When checked, only the selected block will be printed.

Options

Use this section to specify the location of the output, and optional actions for the process.

Syntax print: When checked, font styles for reserved words, etc. print in the hard copy as they display on screen (bold, italic, etc.)

Use color: Check if you are using a color printer and want the on-screen color scheme replicated in your printed output.

Font size: Select the font size to be used in the printed output.

Line spacing: Select the desired amount of space to be left between lines in the printed output. Range is from 0.3 to 2.0 lines; default is 1 line.

Buttons

Print Options: Launches the Options dialog at the Project level displaying only the *Print* page. You can change any of the configuration options before printing the file. Note that your changes will persist at the Project configuration level after the print job finishes. If you only want to change settings temporarily for the process, choose Options | Project | Print after the print job is finished and reset the options to their previous values.

Preview: Toggles preview mode in which a WYSIWYG preview of the printed output displays in the Print File dialog. (See next section.)

Print: Sends the file to your default system printer or print queue.

Cancel: Closes the dialog without printing the file.

Preview mode

In preview mode the dialog displays the following additional controls:

Update: Refreshes the WYSIWYG view, after changes to the font size or line spacing.

First: Jumps the WYSIWYG view to the first page when another page is displayed.

Previous: Jumps the WYSIWYG view to the previous page.

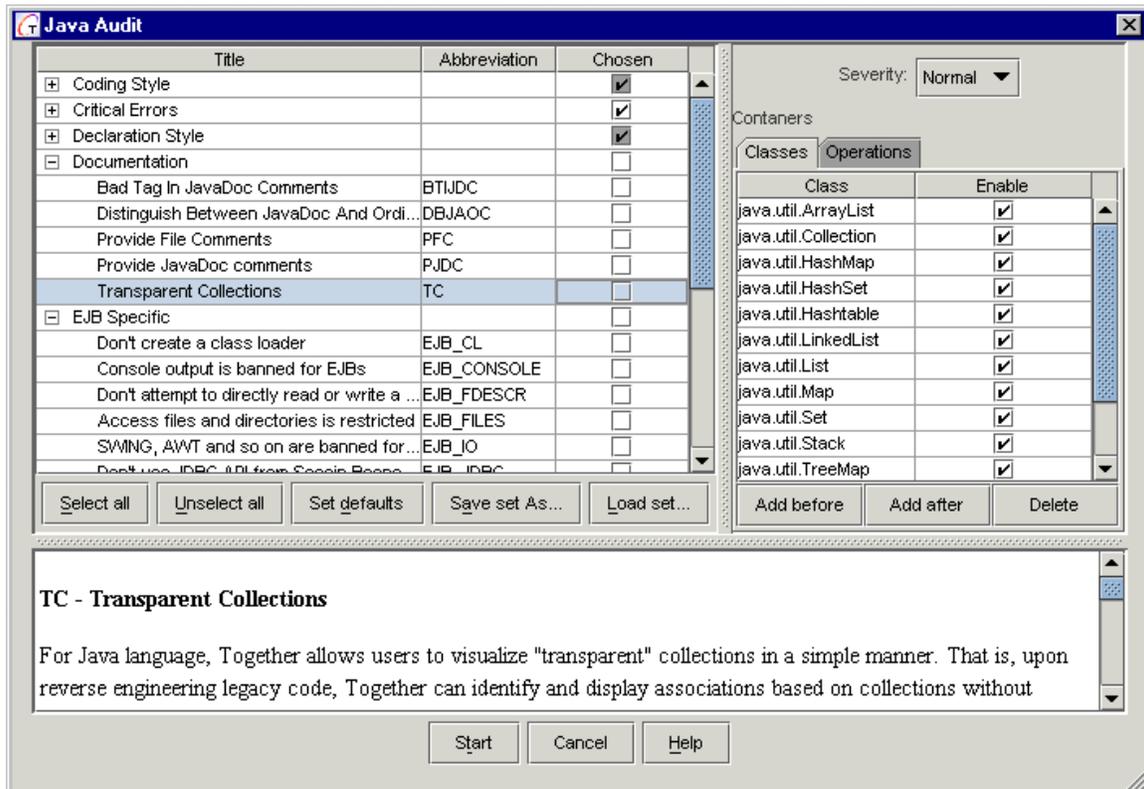
Next: Jumps the WYSIWYG view to the next page when there is more than one page to be printed.

Last: Jumps the WYSIWYG view to the last page when there is more than one page to be printed.

Page count: Read-only display of the number of pages in the print job

Tip: If the WYSIWYG view does not properly display color scheme and fonts, increase font size.

QA: Audit dialog



Select panel

Provides a list of available audit checks with the following fields:

Title: full name of the audit check.

Abbreviation: shorthand name that appears in the result report.

Chosen: check to include the audit check in the analysis

Severity: select severity level from the drop-down list.

Description panel

Provides a brief description of each audit check highlighted in the list.

Options panel

The view of this panel and the set of controls vary depending on the selected audit. Severity drop-down list is always present, where you can assign high, normal or low level of severity to each selected audit.

Find explanations of the most sophisticated controls of the Description panel for the particular audit.

Buttons

Select all and **Unselect all:** global quick-select/deselect buttons

Set defaults: press to select the default set of audit checks.

Save set As: press to save your customized list of audit checks.

Load set: press to invoke the file chooser dialog and select the desired audit file (*.adt).

Print: prints results to the default printer. You can specify details in the Print dialog.

Dispose of: allows to exclude certain rows from the audit report.

Group by: allows to group the report by rows of the selected column.

Sort:: This command invokes the *Sort* dialog, where you can sort the audits results by three columns selected from the drop down lists.

Note: columns selected in the Sort dialog are marked with arrows icons and the level numbers (1, 2 or 3). Clicking on the column header toggles sorting of its content in ascending or descending order.

Alternatively, SHIFT+click on a column header marks it with the sort icon and level index, and clicking on this column header reverts the sorting order. When all three sorting columns are already selected, double-click on any column header resets sorting levels to level 1.

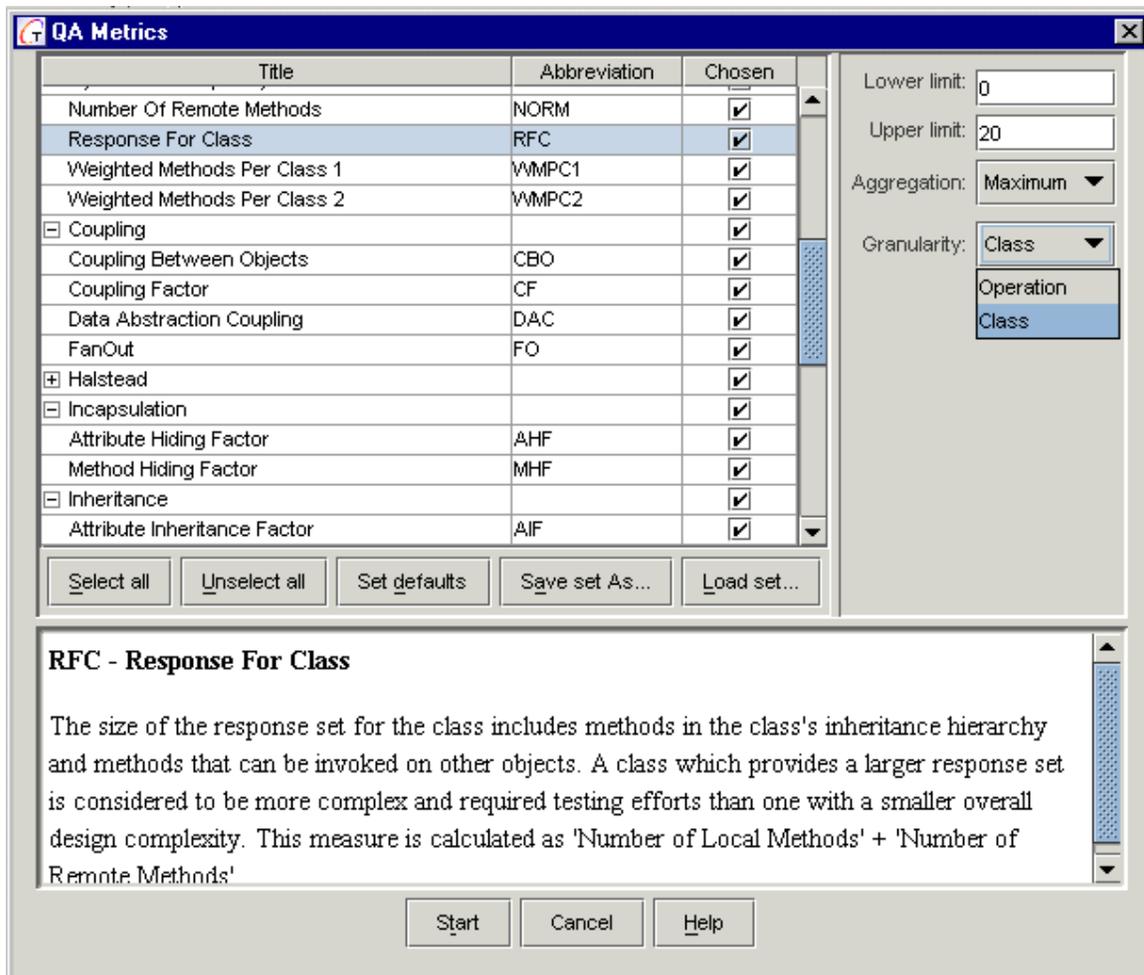
Sort Ascending: sorts the report rows in the ascending order of the selected column

Sort Descending: sorts the report rows in the descending order of the selected column

Related topics

How to Perform Audit Analysis

QA: Metrics dialog



Select panel

This panel displays the list of available metrics and a set of options that can differ for different metrics.

List of available metrics contains the fields:

Title: full name of the metric

Abbreviation: shorthand name that appears in the metrics result report.

Chosen: check to include the metric in the analysis

The Options pane allows to define the settings of the analysis. Options listed below are common to actually all the metrics.

Lower limit and **Upper limit:** enter maximum and minimum number of occurrences to be included into the analysis.

Aggregation: select the type of aggregation from the drop-down list (sum, average, maximum etc.)

Granularity: defines the depth of analysis (project, package, class etc.)

Properties panel

Provides a brief description of a metric highlighted in the list.

Buttons

Select all and **Unselect all:** global quick-select/deselect buttons

Set defaults: press to select the default set of metrics.

Save set As: press to save your customized list of metrics.

Load set: press to invoke the file chooser dialog and select the desired metrics file (*.mts).

Start: runs the metrics report

Cancel: abandons all changes and quits the dialog.

Help: invokes this page.

Metrics results report

Displays the results of the analysis run on the selected metrics in a treeview:

Item: name of the analyzed object.

Metrics fields: the number of metrics fields varies according to the metrics selected for analysis. One column is displayed per each selected metric. Information can be sorted in ascending or descending order by each column of the report.

Right click on a Metrics field cell of the result report invokes a speedmenu.

Result Report Speedmenu

Item	AC	CC	NORM	RFC	VMPC1	VMPC2
<default>	44	55	88	412	55	73
DataManagement	0	1	0	1	1	2
ProblemDomain	23	33	27	51	33	28
CashSale	67	33	27	51	33	27
CashSaleDetail	10	7	1	8	7	9
IMakeCashSale	0	1	0	1	1	2
InsuffPaymentExc	0	2	0	19	2	3
ProductDesc	26	22	13	30	22	28
ProductPrice	32	16	2	18	16	27
Requirements	7	4	0	4	4	6
UserInterface	100	36	73	393	36	29
CashSalesApp	1	5	10	12	5	3
POSFrame	172	36	73	393	36	29
POSFrameAboutE				313	6	6
SaleUI				354	2	3
data_management				53	11	14
problem_domain				67	44	43
server	14	55	25	53	55	73
user_interface	142	46	88	412	46	34
util	4	13	1	11	13	16
IMakeCashSale	0	1	0	1	1	1

Open: allows to open the selected element in the Diagram and Editor panes.

View: this command enables a choice of three options to display the metrics results by classes, packages or by both classes and packages.

Select All: highlights all items in the report

Copy: copies the selected rows to the clipboard.

Refresh: synchronizes the report with updates to the source code.

Restart: re-invokes the Metrics dialog, thus enabling to change settings and run Metrics again.

Show Description: displays a window with the description of the selected metric.

Export: saves the whole report or selected rows in a text file.

Print: prints results to the default printer. You can specify details in the Print dialog.

Bar Graph: displays graphic representation of the selected metric. Double click on each bar of the resulting graph moves to the source code.

Kiviat Graph: displays distribution of metrics for the selected item.

Sort: This command invokes a Sort dialog window, where you can sort the metrics results by three columns selected from the drop down lists.

Sort ascending / Sort descending: check the appropriate flag or click on a column header to sort results in the required order.

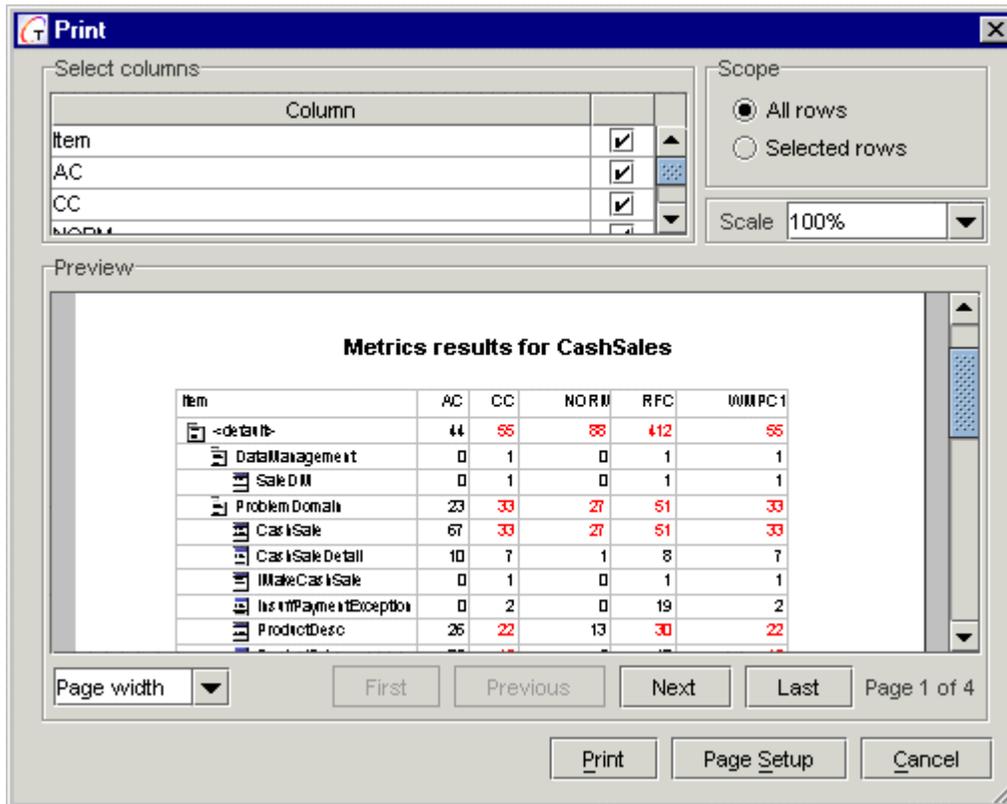
Note: columns selected in the Sort dialog are marked with arrows icons and the level numbers (1, 2 or 3). Clicking on the column header toggles sorting of its content in ascending or descending order.

Alternatively, SHIFT+click on a column header marks it with the sort icon and level index, and clicking on this column header reverts the sorting order. When all three sorting columns are already selected, double-click on any column header resets sorting levels to level 1.

Related topics

How to Perform Metrics Analysis

QA: Print Dialog



Select columns: Set flags for the columns of the resulting table to include in the printout.

Scope: choose radio button to print the entire table, or selected rows only.

Scale: Select the scale of printout from the dropdown list. Note that printout magnification is limited to 200%. If a greater value is specified, 100% is assumed by default.

Preview: This pane displays the preview of the printout according to the above settings. Select the preview scale from the dropdown list under the pane, or enter the desired value. Note that the preview scale is limited to 300%. If a greater value is specified, 100% is assumed by default.

First / Previous / Next / Last: Use these buttons to navigate to the desired page of the preview.

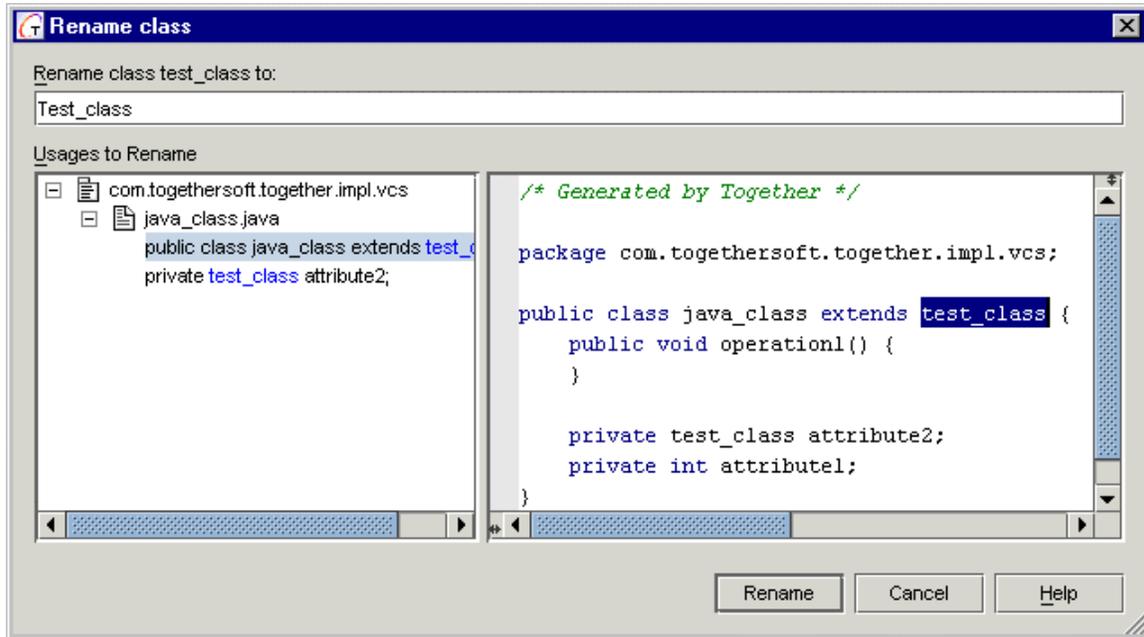
Print: invokes standard system print dialog.

Page setup: displays page setup window where you can adjust paper size and orientation.

Cancel: discards all changes and closes the dialog.

Renaming Dialogs

These dialogs are invoked by *Rename class/attribute/operation* command under *Refactoring* node of the object speedmenu.



The dialogs only differ in the window title.

Rename class/attribute/operation to...: use this field to enter new name for object to be renamed.

Usages to Rename: left frame displays treeview of usages to be updated. Right frame displays the code of the selected node. Use CTRL+U hotkey to toggle to the treeview.

Buttons

Rename: performs the operation and closes the dialog

Cancel: discards all changes and closes the dialog

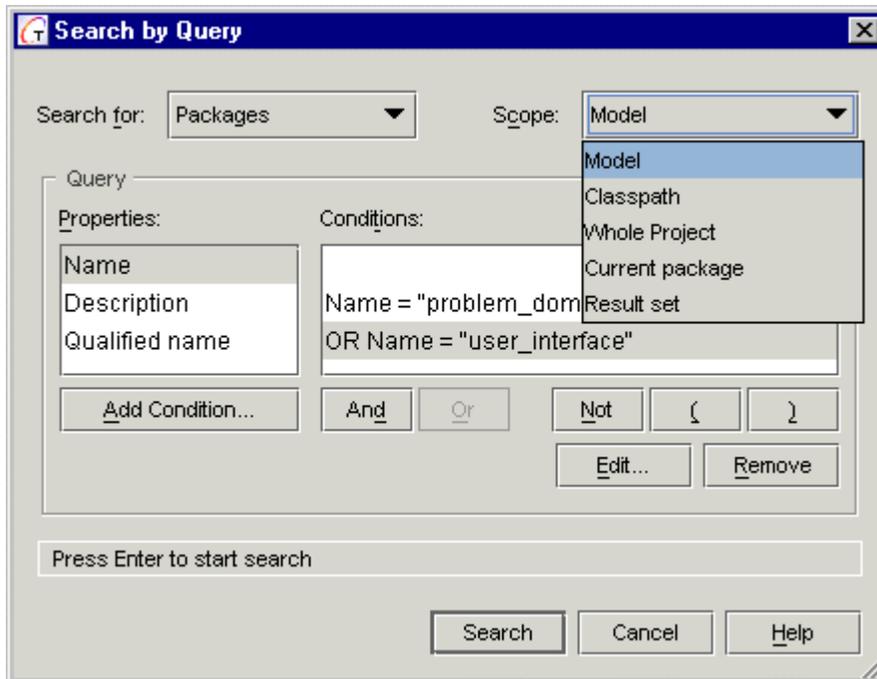
Help: displays this topic

See also

Renaming

Search by Query Dialog

This dialog implements a handy tool to search elements by query. Selected element is sought for in the specified scope, according to the user-defined filter expression.



Search for

Choose the type of element to find from this combobox. Possible values are

- package,
- class,
- interface,
- operation,
- attribute.

Scope

Choose the range of search from this combobox. Possible values are:

- model* - search in the model of the open project
- classpath* - search in the imported classes of the open project
- whole project* - search in the model and classpath
- result set* - search in the results of the previous search
- current package*

Query

In this section you can define the query for your search. Query, or filter is a logical expression that consists of a set of conditions. The Query section of the dialog includes a list of Properties and a field of Conditions.

Properties are specific for each element type. Conditions look like:

<property> <operation> <value>, or
hasProperty<property>

where <property> is the name of element's property, and <operation> is "=", "!=", "<", or ">". Conditions may use brackets and logical binary operations (AND, OR, NOT). The dialog provides relevant buttons for each logical operation.

Buttons

Add condition

This button is default when the list of properties gets the focus. Invokes Condition dialog to select logical operation and enter value. The search can be made context sensitive.

Edit

This button is default when Conditions list gets the focus. Double-click on a selected element works as Edit button, i.e. invokes Condition dialog to modify the current settings.

Choose the required logical operation using one of radio-buttons (equal - not equal). It is possible to make search case-sensitive, and use wildcard patterns.

Remove

This button is enabled together with *Edit* button, when Conditions list gets the focus and deletes selected condition from the list.

Logical operations buttons

These buttons allow to create logical expressions for the queries. Every new condition is added by default with the AND operation, except for the very first condition. Buttons *AND* / *OR* are mutually dependent and exclusive (when one is pressed, the other is disabled).

Search

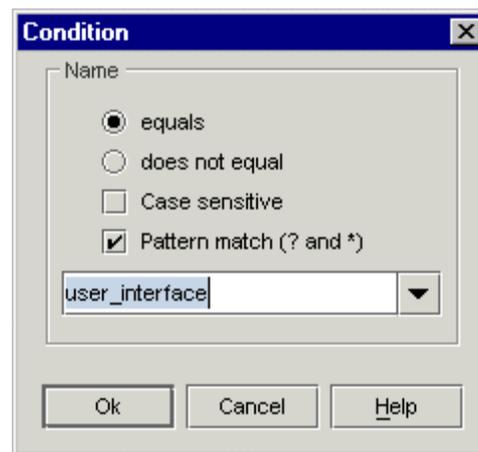
This button becomes enabled when the logical expression in the Conditions field is correct. If the filter expression is wrong, *Search* button is disabled, and appropriate diagnostic message displays in the status line of the dialog.

Cancel

Discards all changes and cancels the search.

Help

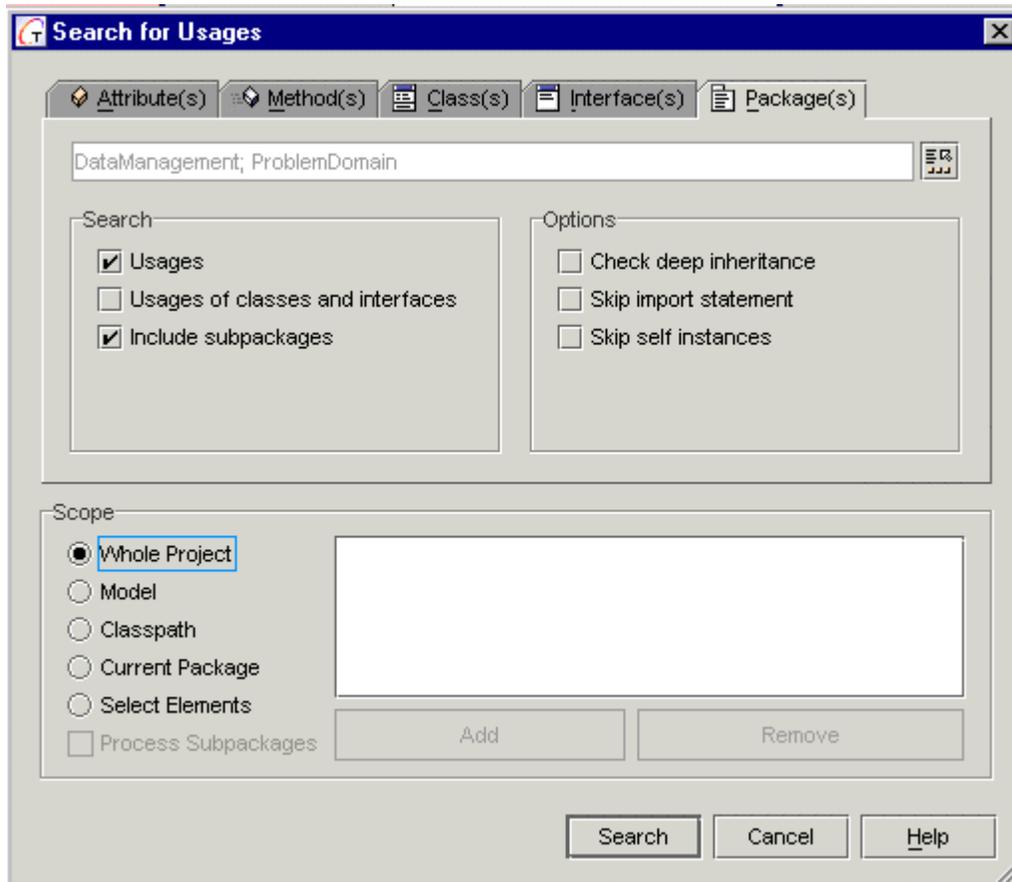
Invokes this help topic.



Search for Usages Dialog

This dialog is invoked by the **Search for Usages** command from a diagram element speedmenu or from the menu **Search**, and looks for occurrences of the selected element according to the specified search options and scope. Search is carried out by *FindUsages* module. The dialog is arranged as a tab set, each tab corresponding to selected element type.

Package tab



Search

Usages finds all places in the source code, where the package is referred by its name.

Usages of classes and interfaces finds all places in the source code, where classes and interfaces from the selected package are referred by their names.

Include subpackages finds all places in the source code where subpackages of this package are referred to by their names.

Options

Check deep inheritance locates indirectly derived classes and interfaces.

Skip import statements excludes import statements from search results.

Skip self instances excludes from the search results any references to the target package within itself.

Scope

Whole Project includes Model and Classpath into the search.

Model provides search throughout the model

Classpath provides search throughout the classpath

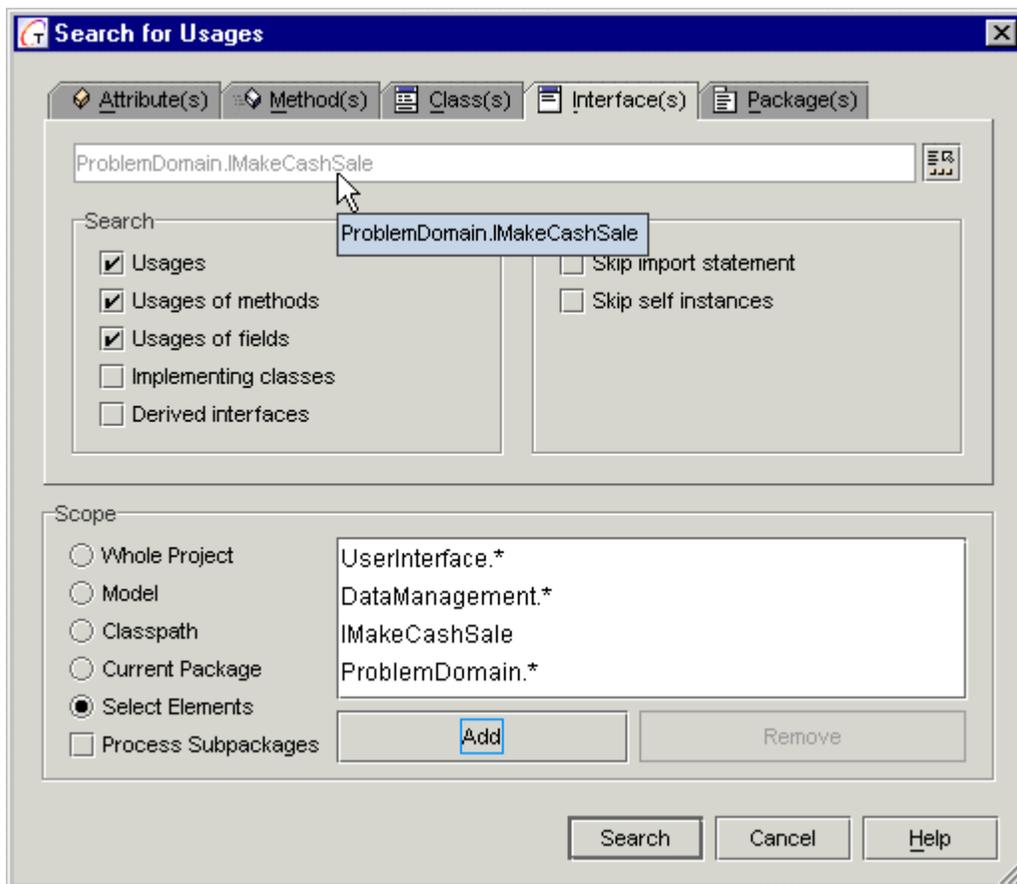
Current Package provides search throughout the currently selected package. With this option selected, the **Process Subpackages** checkbox is enabled, that allows to recurse into subpackages.

Select Elements allows to create a custom search path and enables **Add** button that invokes Select Classes and Packages dialog (see Selection Manager Dialogs). **Remove** button allows to delete selected element from the search path. This option also enables Process Subpackages checkbox.

Important note for C++ users

In C++ projects, when the search scope is specified as the current package, or a set of selected packages, the result still can include occurrences outside the selected scope. It happens due to the fact that packages are not C++ intrinsic objects, and definition files can reside in other folders than the header files. Actually, they are stored in different packages. It is possible to look for usages in definitions and bodies of the operations in the folders that do not correspond to the selected packages.

Interface tab



Search

Usages finds all places in the source code, where this interface is referred by its name.

Usages of methods finds all places in the source code, where methods of this interface are called from.

Usages of fields finds all places in the source code, where the variables of this interface are used.

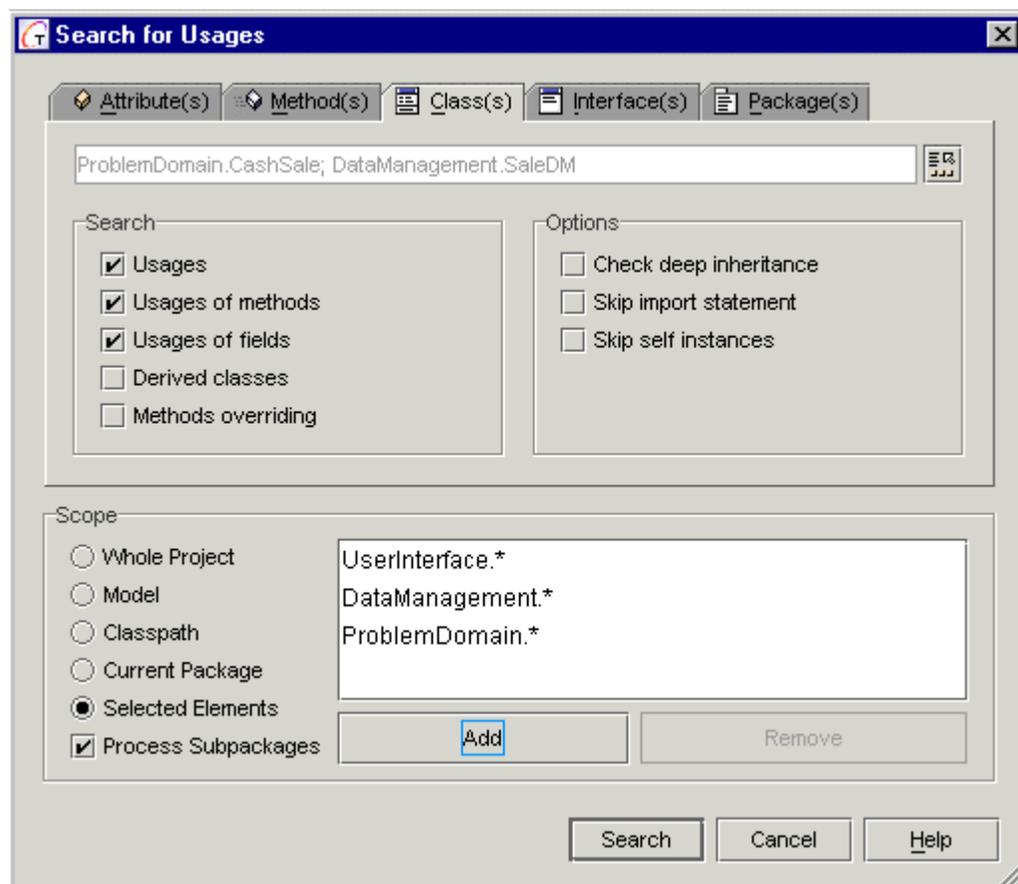
Implementing classes finds all classes implementing the selected interface.

Derived interfaces finds all interfaces extending the selected interface.

Refer to the Package tab section for the Options and the Scope of search.

Class tab

Controls for the classes in general are similar to those of the interfaces, with some differences.



Search

Usages finds all places in the source code, where this class is referred by its name.

Usages of methods finds all places in the source code, where methods of this class are called from.

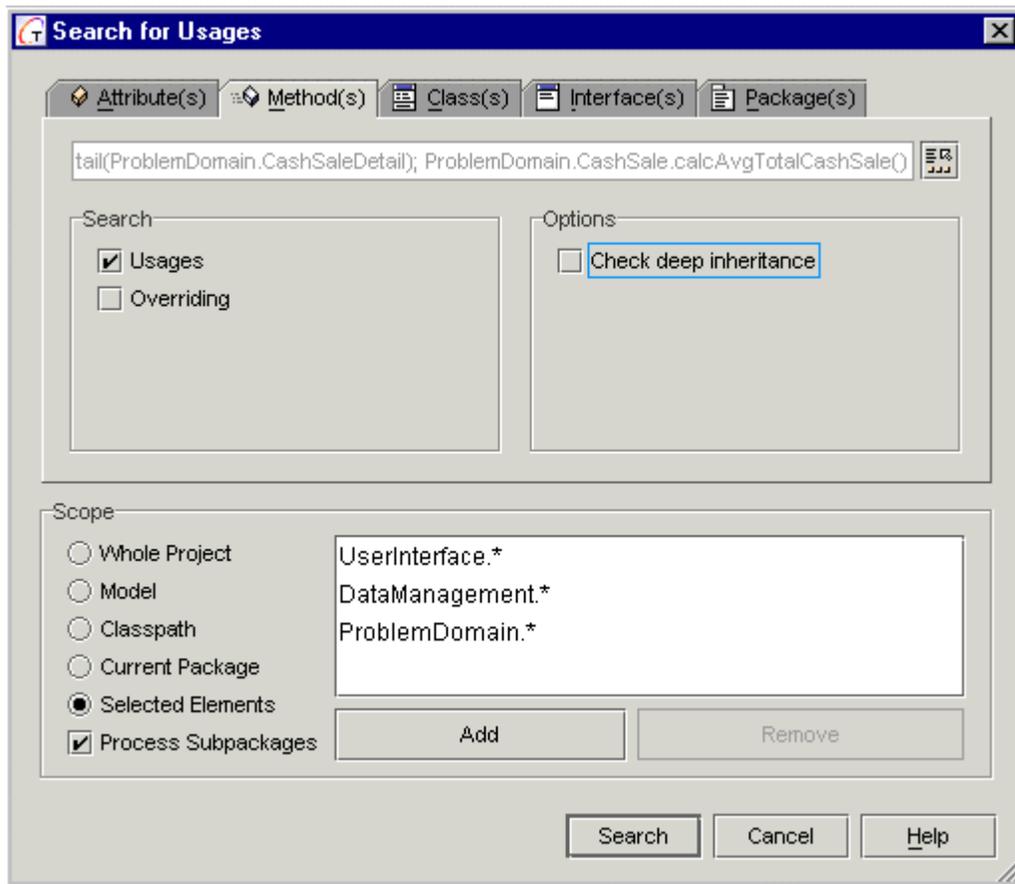
Usages of fields finds all places in the source code, where the variables of this class are used.

Derived classes finds all classes that extend the selected class.

Method overriding searches for the overridden methods of the selected class.

Refer to the Package tab section for the Options and the Scope of search.

Method tab



Search

Usages finds all places in the source code, where the method is invoked.

Overriding finds all methods that override (implement, for an abstract method) the selected method

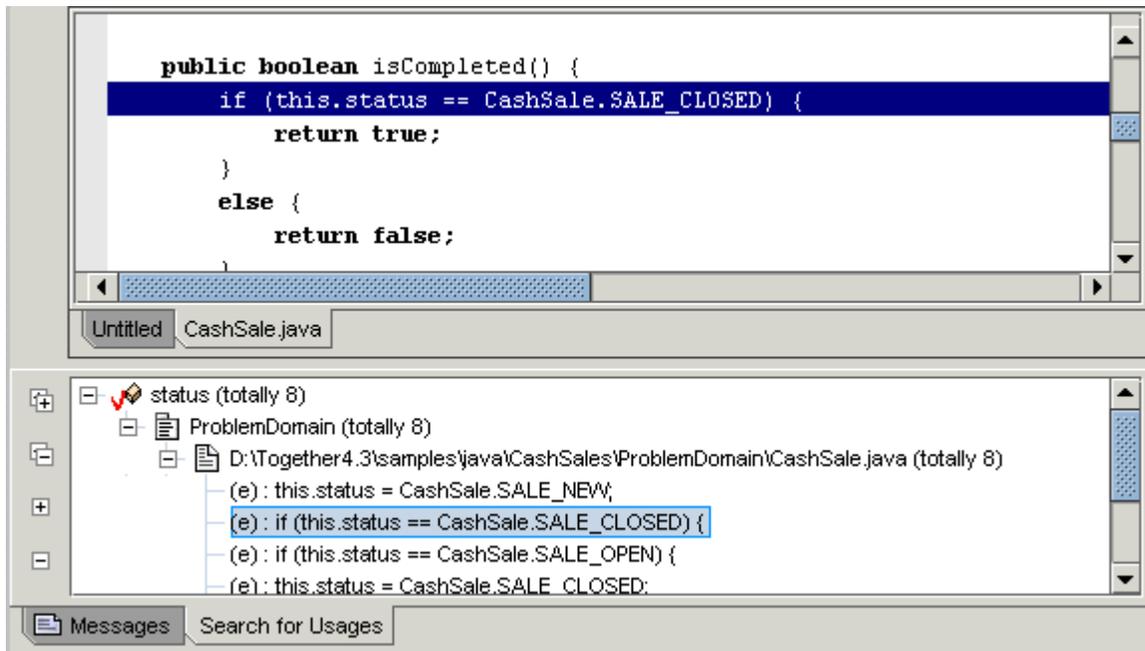
Refer to the Package tab section for the Options and the Scope of search.

There are no special search types and options for the attributes. The module scans the source code and finds all places, where the selected variable is used. Search is performed within the specified Scope.

Search Results show up in the **Search Result** tab of the Message pane. If you select a line on the list of search results, the relevant class displays in the Editor pane, highlighting the usage example.

Attribute tab

Attribute tab provides search for usages in the same Scope. The following screenshot demonstrates search results for an attribute.

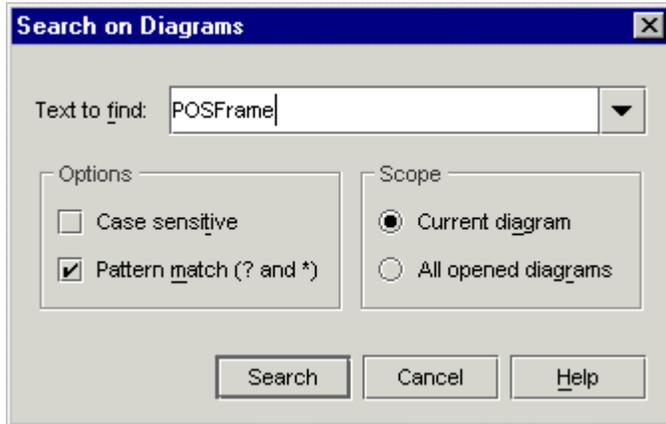


See also

Selection Manager dialog

Search on Diagrams

Use *Search on Diagrams* command on the Search menu to invoke this dialog.



Text to find

Enter the string you want to locate in the current diagram and click *Search*. The history of previously entered strings is available from the dropdown list.

Options

Case sensitive: When checked, search operation is case sensitive. When cleared, character case is ignored.

*Pattern match (? and *)*: when checked enables using wildcards. All matches to the specified pattern show up in the *Search Results* tab of the Message pane.

Scope

Use radio buttons to choose searching through the current diagram only, or through all opened diagrams.

Buttons

Search: starts the search with the specified parameters and closes the dialog.

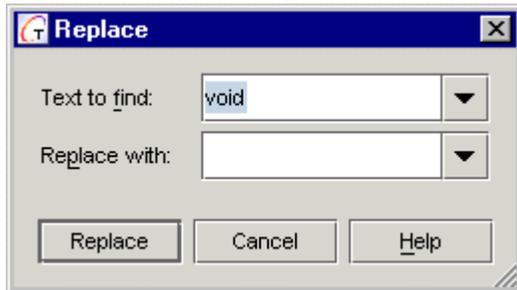
Cancel: discards all changes and closes the dialog.

Help: displays this topic

Search / Replace Dialogs

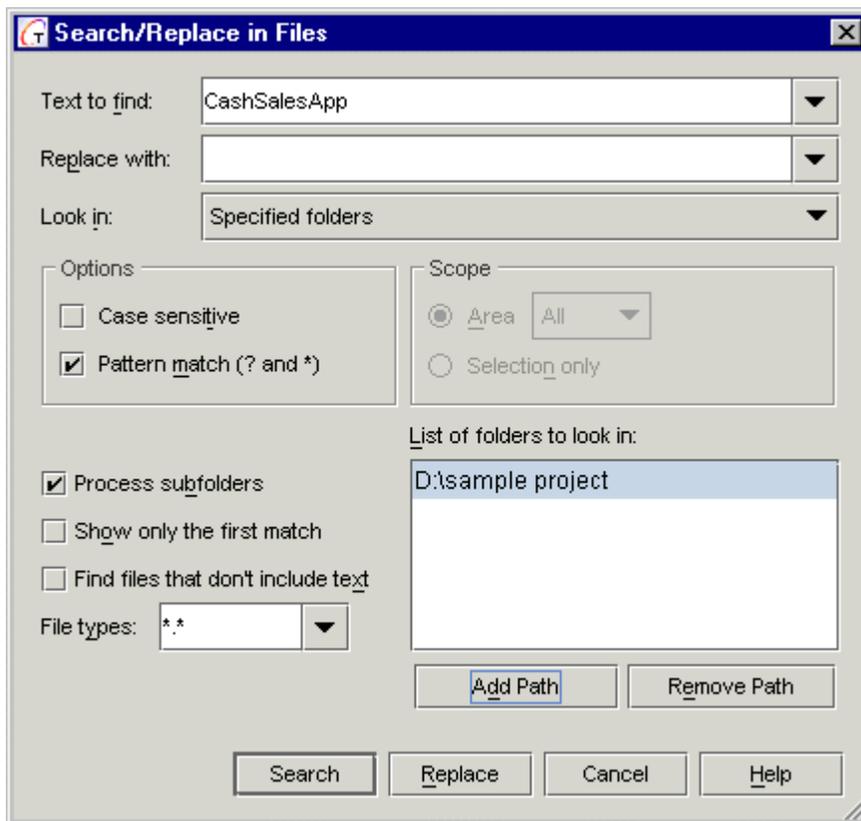
Search/replace function is available in two forms: reduced and complete.

The reduced form is invoked by *Search* (CTRL+F) and *Replace* (CTRL+H) commands on the Search menu. *Search/Replace* commands seek the current file in the Editor pane for the specified string, and replace the occurrences with the specified replacement string.



F3 finds next occurrences; SHIFT+F3 finds previous.

Complete form is invoked from the Search menu using *Search/Replace in Files* command, or CTRL+SHIFT+R shortcut.



Results of search display in the Search Results tab of the Message pane as a treeview. You can navigate through the search results, which is immediately responded in the Editor pane. Re-invoking a dialog starts new search.

Find/Replace strings

Text to find

Enter the string you want to locate in the specified scope or select it from the drop-down history of recently used search strings.

Replace with

Enter the replacement string. Enter the required value or select from the drop-down history of the recent replacements.

Look in

This control provides a drop-down list of available files/paths: current file, all opened files, all project source files or specified folders only. In the latter case the adjacent field *List of folders to look in*, with its buttons become enabled.

Options

Case sensitive: When checked, search operation is case sensitive. When cleared, character case is ignored.

Patterns match: When checked, enables using wildcard patterns in the search string.

Scope

This section section is only enabled if Current file is selected from the *Look in* drop-down list. Choose an option to control the search scope within the current text.

Area: Choose search area and direction from the drop-down list. It is possible to search through the entire current file, or upward / downward from the current cursor position.

Selection only: Limits the search to a block of text currently selected in the target file. Disabled if no text is selected.

The states of flags below depend on the selected look-in range.

Process subfolders: enabled when *Specified folders* option is selected. Allows recursion into the nested folders.

Show only the first match: always enabled. Terminates the search after the first occurrence of the search string.

Find files that don't include text: enabled unless *Current file* is selected. This flag provides inverse search in the selected range.

List of folders to look in

Use *Add Path* / *Remove Path* buttons to make up the list of paths to be searched.

Buttons

Search: Starts search with the current settings.

Replace: Starts search with the current settings. At each occurrence of the search string displays confirmation dialog. You can replace each particular occurrence, replace all occurrences in the entire search range, or in the current file only.

Cancel: discards all changes and closes the dialog.

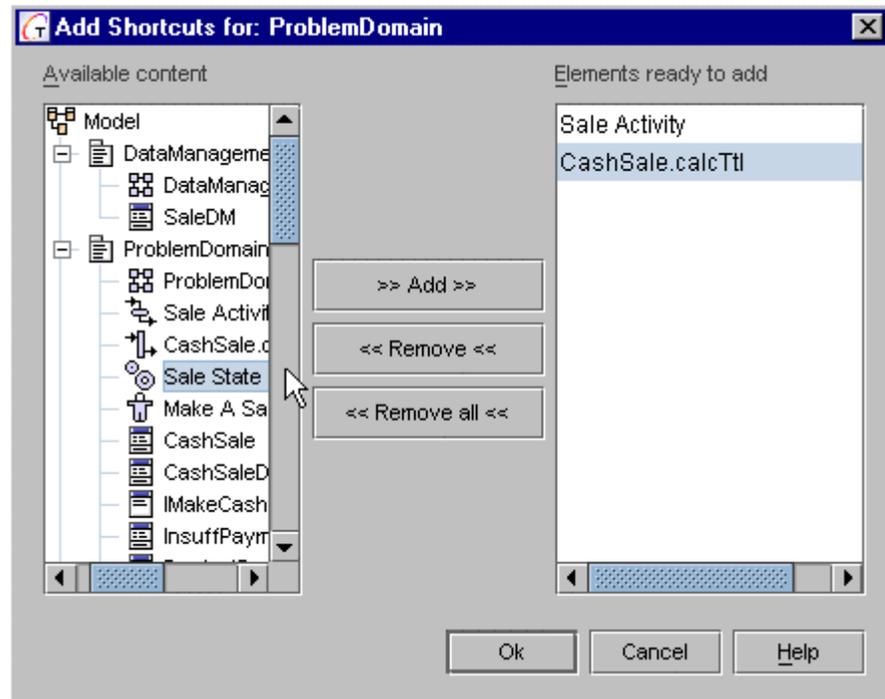
Help: invokes this dialog.

See also

Keyboard Shortcuts

Selection Manager dialogs

This is a general type of dialog that allows to select desired elements from the available contents, and add them to a certain destination scope. All Selection Manager dialogs have similar structure and varying title strings. Such dialogs are invoked by *Shortcut* command on the diagram speedmenu, F1 button in the Editor pane, or the browse buttons (for example, in the Search for Usages command).



Example of Selection manager dialog

Dialog title

The dialog's title varies depending on the way it is invoked. It displays the string that corresponds to the invoking object or property.

Available content pane

This pane displays all of the content available in your project on the search paths specified in Project Properties. It can use the explorer to navigate to the element and select it for inclusion in the pane of values returned by the dialog to the invoking object.

Existing and/or ready to add

This pane displays two kinds of data:

- Value(s) already existing and passed from the invoking object (if any)
- Values of the selection(s) you have added from the left-hand pane (if any)

Add

Enabled when an element is selected in the left-hand pane. Adds the selected element to the right-hand pane.

Remove

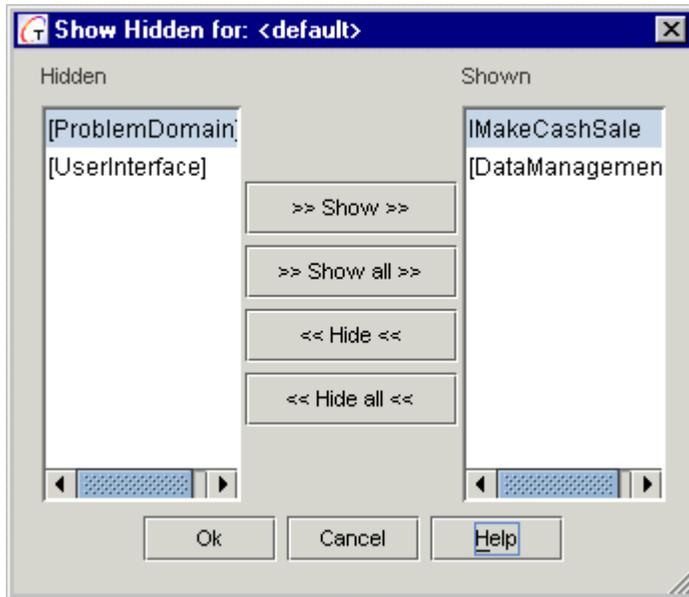
Enabled when you select an item in the right-hand pane. Removes the selected item from the pane. All removed values or objects are removed from the invoking property or diagram upon clicking OK.

Remove All

Enabled when items are present in the right-hand pane. Removes all items from that pane. All removed values or objects are removed from the invoking property or diagram upon clicking OK.

Show Hidden Dialog

Show Hidden dialog is invoked from the diagram speedmenu and allows to hide/show diagram elements as required.



The dialog window contains two panes that display the lists of hidden and shown elements, and a set of buttons that allow to move selected elements between the two lists.

Show: Moves selected element from the Hidden pane to the Shown pane.

Show all: Moves all elements from the Hidden pane to the Shown pane.

Hide: Moves selected element from the Shown pane to the Hidden pane.

Hide all: Moves all elements from the Shown pane to the Hidden pane.

OK: Completes operation and closes the dialog.

Cancel: Discards all changes and closes the dialog.

Help: Displays this topic.

Synchronize with Default Printer Dialog

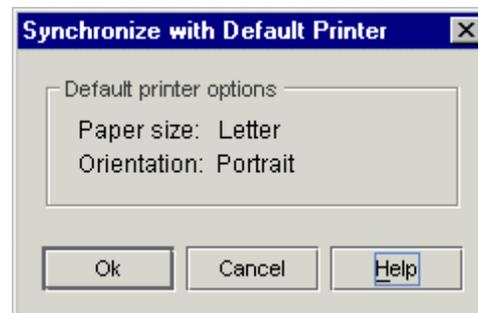
This dialog is invoked from the Print tab of the Options dialog, and is used whenever you need to return from the custom printer settings to the default settings. This helps avoid slowing down, which happens when the printer is polled automatically.

Buttons

OK: restores default printer settings of the paper size and orientation, and closes the dialog.

Cancel: Preserves current custom settings of the paper size and orientation, and closes the dialog.

Help: displays this page.



Version Control interaction dialogs

When version control interaction is enabled, the Version Control command appears on speedmenus of source-generating visual elements in Class diagrams and the Model tab of the Explorer. This command has a submenu with commands that launch dialogs that enable you to complete the selected operation (get, add, etc.)

Dialogs are presented for these version control operations:

- Add
- Get
- Update
- Check-in
- Check-out
- System

The System dialog is essentially a visual client for your CVS or SCC version control system and which enables all the basic interactions with that system.

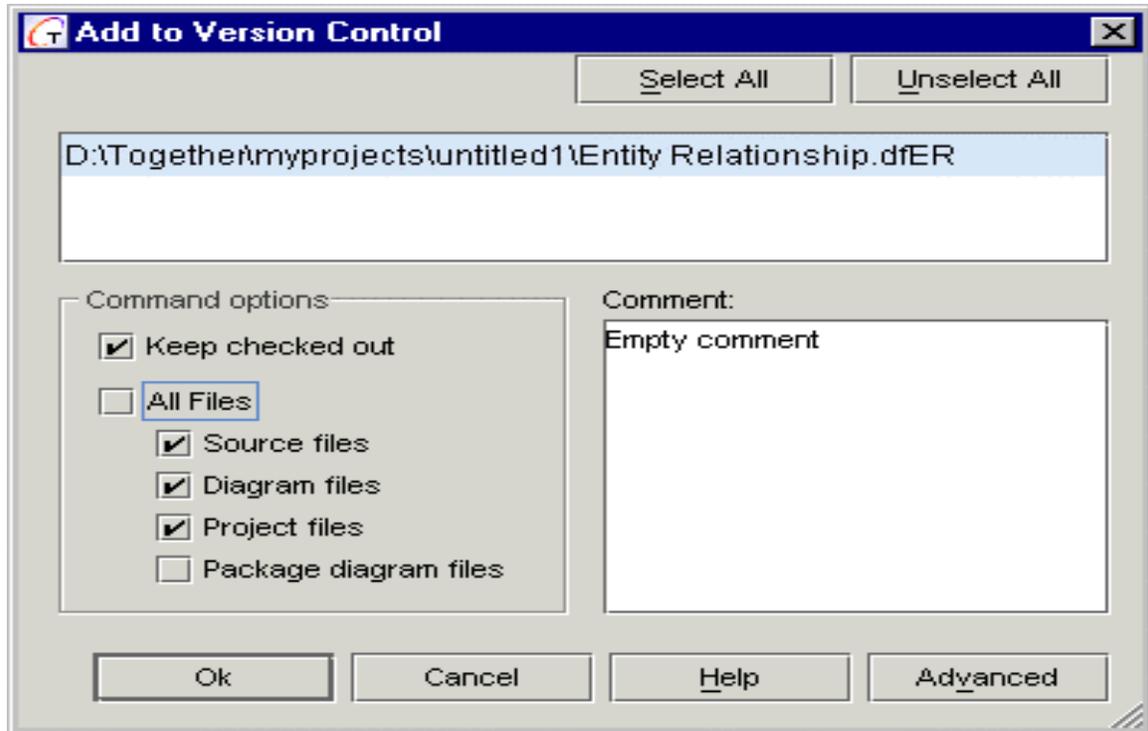
Dialog content and layout

The underlying implementation is the same for each VCS dialog, but what each one contains can vary according to:

- the function (add, checkout, etc.)
- the VCS you are using (i.e., CVS-LAN, CVS Client Server, or an SCC-compliant system)
- the content you are working with... a file or a directory.

Add dialog

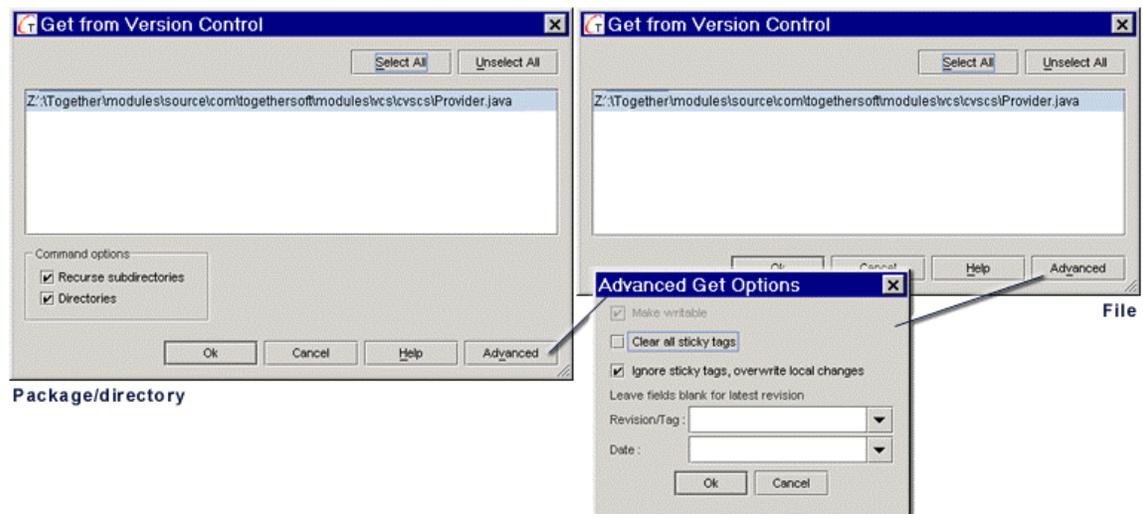
The Add dialog is essentially identical for all supported Version Control Systems. You can add files one at a time or several at once, depending on what is selected when you invoke the Add dialog. If multiple items are selected, you can limit the file types that the Add operation will act upon using the *All Files* checkbox and those beneath (see figure below for an example).



The Add dialog is essentially the same for CVS and SCC systems

Get dialog

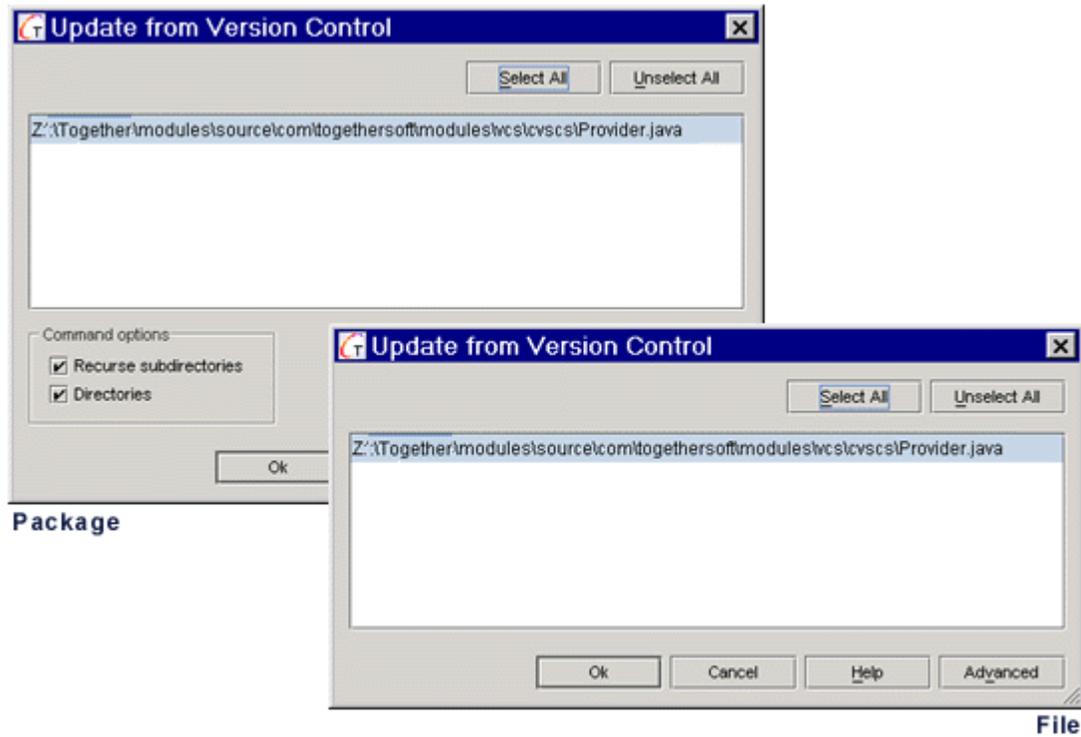
The Get dialog differs slightly depending on whether it was invoked on a package/directory or on a single file. The *Advanced* button may or may not be enabled depending on what version control system is integrated. If it is enabled, it launches a dialog with additional system-specific *get* options.



Variant states of the Get dialog showing Advanced dialog for CVS

Update dialog

Update in CVS Client-Server means synchronization of local files with the respective version on the CVS server. For CVS-CS integrations, the Update command is available on relevant speedmenus, and the Update dialog is presented on invocation. The dialog varies slightly depending on whether you invoke it on a directory or on a file (or several files).

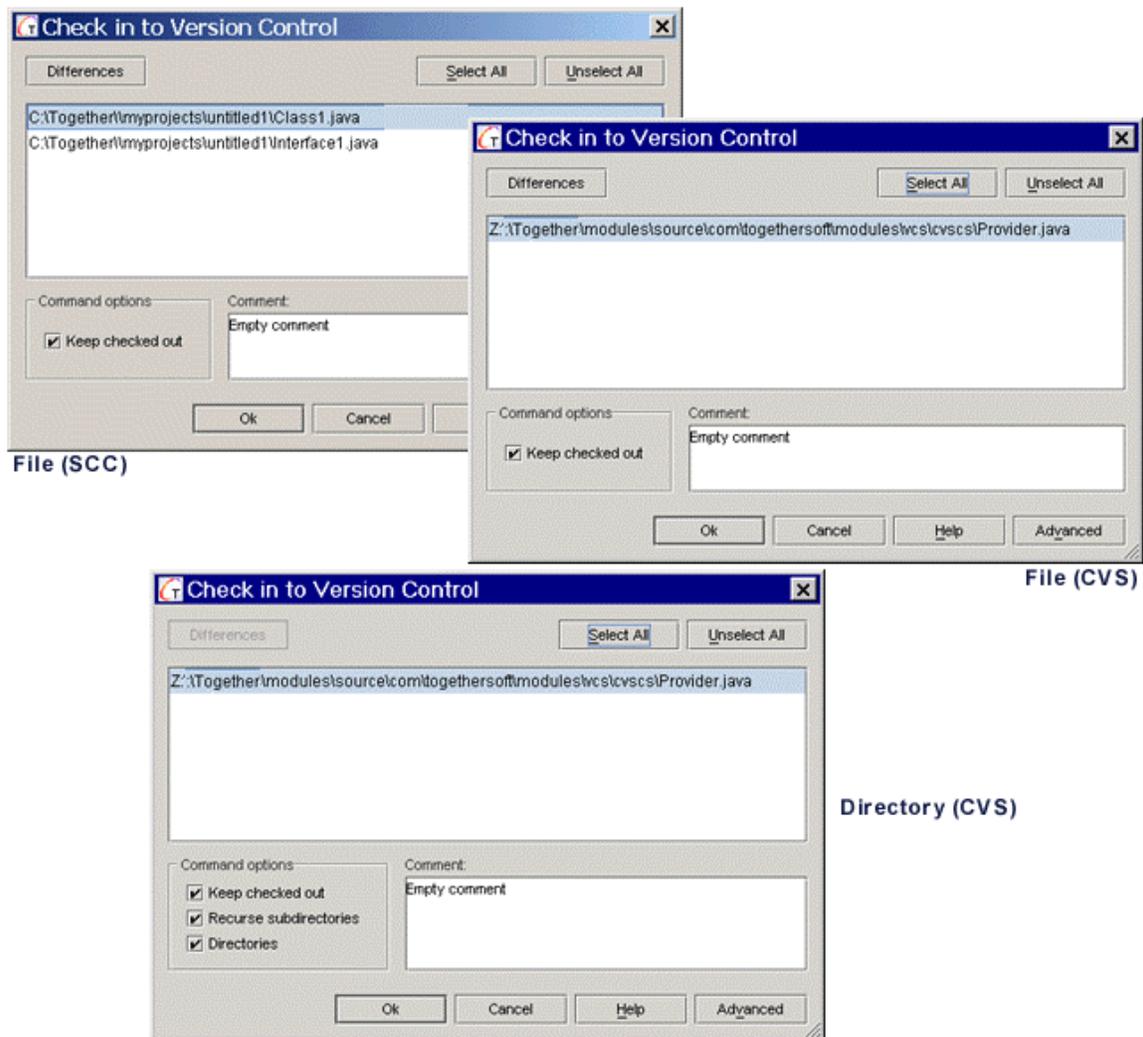


Variant states of the VCS Update dialog (for CVS Client-Server only)

Check-in dialog

The Check-in dialog varies slightly depending on whether you invoked it on a directory/package or a file. For directories, the dialog contains "Directories" and "Recurse subdirectories" options. *Directories* means that that command is applied on the selected directory. Unchecking displays a list of files in the directory which can be individually selected.

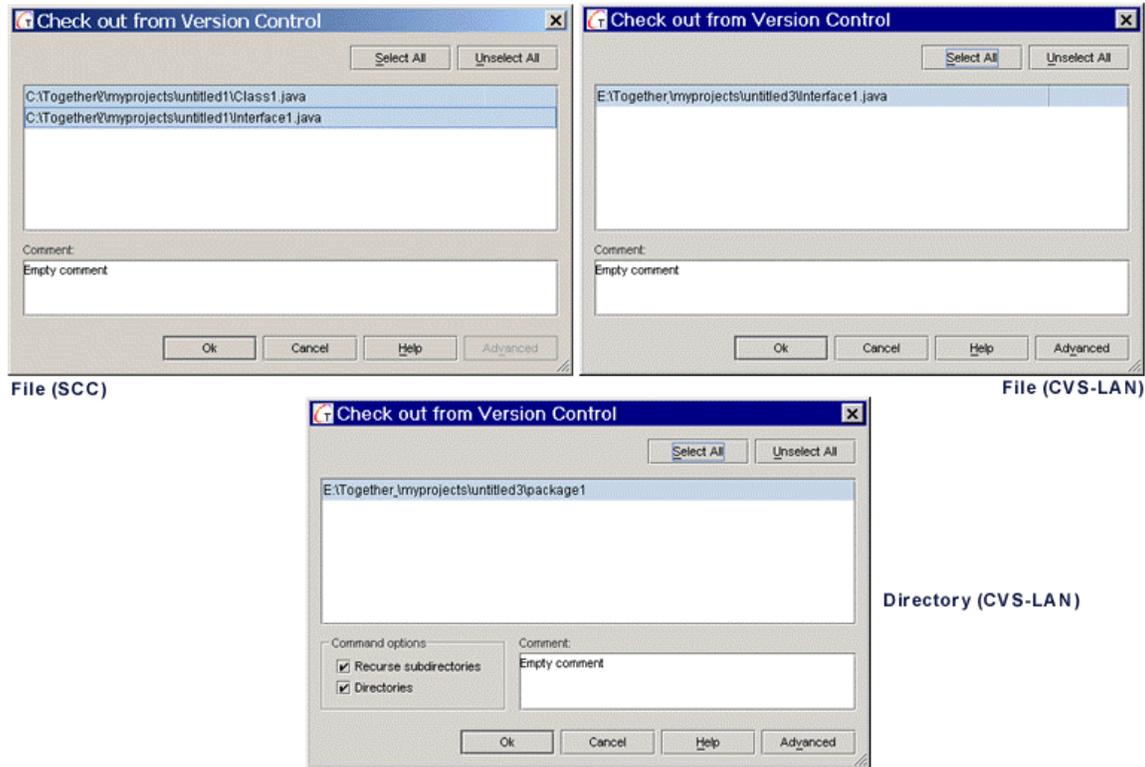
When invoked for file(s), the *Differences* button is enabled. You can use it to run your VCS's *diff* operation on the file(s). The button is disabled for directories. In CVS client-server mode "Keep checked out" is selected and disabled. For single files, the dialogs are same.



Variant states of the VCS Check-in dialog

Check-out dialog

The Check-out dialogs for SCC and CVS LAN differ in same way as Check-in dialogs: in CVS, directories are shown in the listbox by default, plus the options *Directories* and *Recursive*. CVS client-server does not show the Check-out dialog.



Variant states of the VCS Check-out dialog

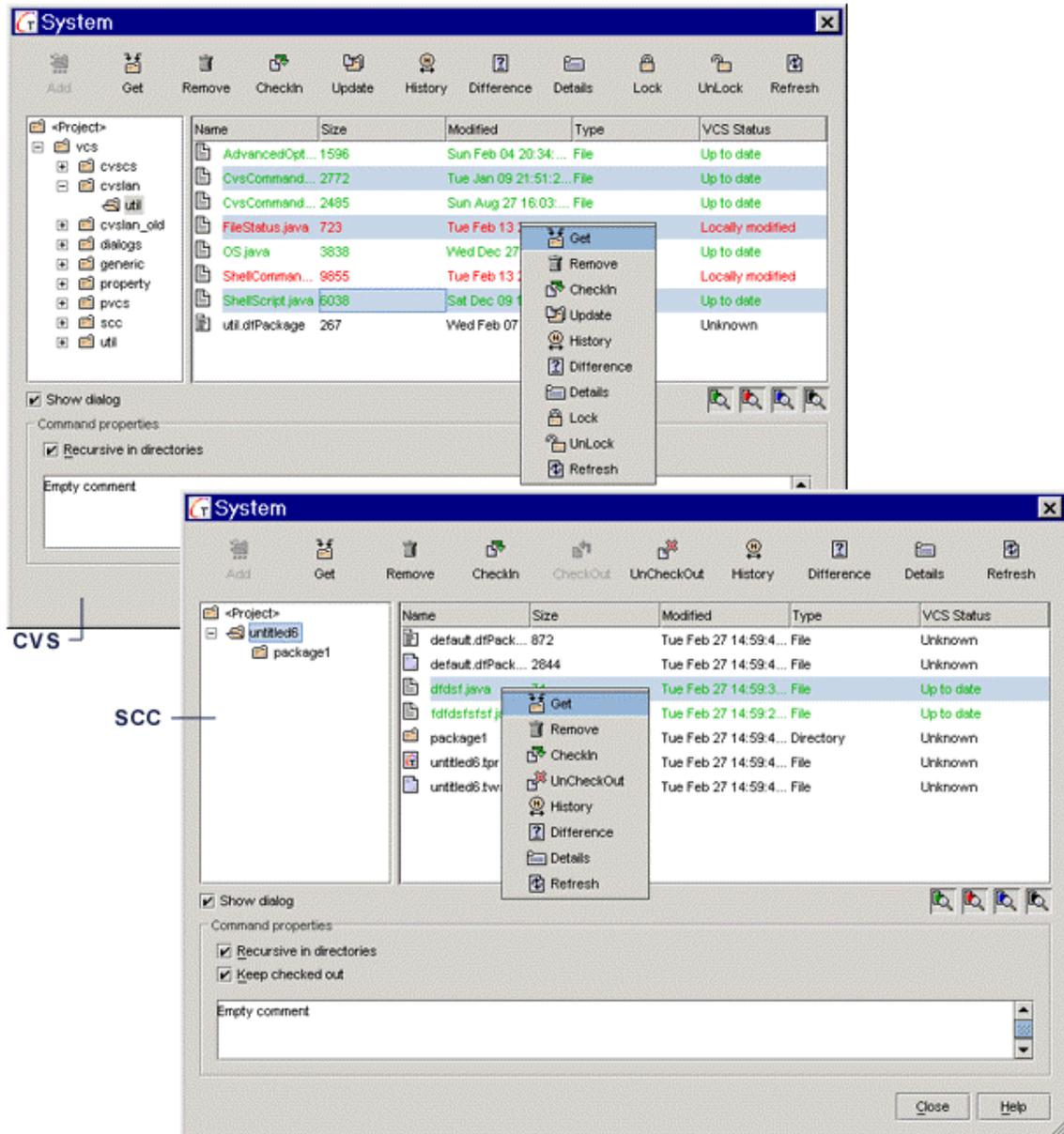
System dialog

The System dialog is essentially a client for the version control system you have integrated with Together. It supports most of the most commonly needed version control functions and enables you to visually navigate around on your system and perform different version control options on different elements in different locations. You can also see the status files under version control (with some exceptions as noted below). You should use the client for your VCS to perform any functions not provided by the System dialog (advanced or administrative tasks for example). The System dialog should provide enough functionality for most developers most of the time, however.

The composition of toolbars and menus in this dialog varies somewhat according to the specific provider you are using. That is to say, there could be some variations not only between CVS and SCC, but among different SCC providers as well.

Caveats

- The System dialog for SCC systems does not display "new on server" files... that is, files present on the server but not present locally.
- Some SCC providers do not support the distinction between "locally modified" and "modified on server" status of files. For such providers, the System dialog only shows locally modified status. Files that are modified on server show as up to date.



System dialog for CVS Client-Server and SCC (Visual Source Safe provider)

Web Services Expert

This topic provides a detailed description of the server-specific pages for the most popular platforms: Apache-SOAP (<http://xml.apache.org/soap>), and BEA WebLogic 6.1 beta application server (<http://e-docs.bea.com/wls/docs61>).

Working with the Web Services Expert, you have to describe what's to be done. Each page of the Web Services Expert contains a number of tasks. After you have filled in all necessary fields and clicked *Finish*, the Web Services Expert starts executing the selected tasks.

Web Services Expert for Apache-SOAP

Process options

Compile classes from the currently selected diagram

Check this box if you want to compile the classes.

Generate Deployment Descriptor(s)

Check this box to create Deployment Descriptor(s).

Open XML editor for the generated Deployment Descriptor(s)

Check this box to open XML Editor, where you can edit the generated Deployment Descriptor(s).

Register WebServices on the server

Check this field to register Web Services on the application server using Apache-SOAP's tool.

"Common Properties" page

Name	Value
Temporary folder	c:\temp
JDK 1.3 root directory	c:\jdk1.3
Apache-SOAP home directory	C:\Together5.5\bundled\tomcat\webapps\soap

The fields on this page are quite self-explanatory. However, here is a brief description for the Apache-SOAP home directory:

Apache-SOAP home directory

Tomcat, that comes bundled with Together, is pre-configured to support Apache-SOAP. If you are going to create a client or a service, the library `TGH/bundled/tomcat/soap` should be added to the project path.

If another version of the Tomcat is used, the users should add Apache-SOAP to Tomcat manually.

"Run-time Registering Web Services" page

Name	Value
Server host name	localhost
Server port number	8080

Server host name

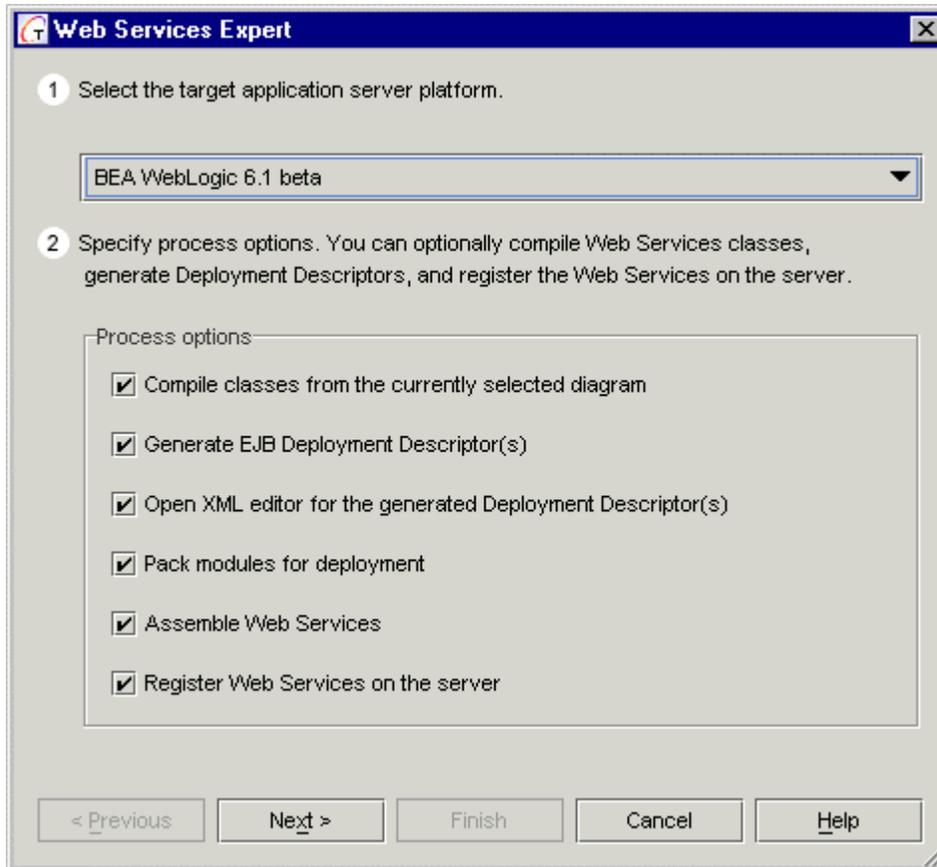
As a rule, the server host is assigned to `localhost`, since this Web Services Expert uses the file system access to the application server with Apache-SOAP, and therefore the server runs on the same computer.

Server port number

The default server port is 8080.

Web Services Expert for BEA WebLogic 6.1 beta Application Server

Process options



Compile classes from the currently selected diagram

Check this box to compile the classes from the selected diagram and to pack them in a JAR file. The new archive is stored in a temporary folder. Standard Sun tools (javac.exe and jar.exe) are used for compilation and packing.

Generate EJB Deployment Descriptor

Check this box to create a Deployment Descriptor for all RPC Web Services and build.xml file.

Open XML editor for the generated Deployment Descriptor and build.xml

Check this box to open XML Editor, where you can edit the generated Deployment Descriptor and build.xml file, which are created during this process.

Pack modules for deployment

Check this box to create a deployable *.jar (standard JAR for deployment of an RPC Web Service Session EJB).

Assemble Web Services

Check this option to generate a build.xml file, and then to generate a deployable *.ear using Java Ant.

Register Web Services on the server

Check this field to register RPC-style or message-style Web Services on the target application server (deployment of the generated *.ear).

"Common properties" page

The fields on this page are quite self-explanatory and don't require any comments.

"Run-time Registering Web Services" page



Name	Value
Server host name	localhost
Server port number	7001
Web Service context	/myContext
Protocol	http
Server root directory	D:\bea\wlserver6.1_beta\config\mydomain

Server host name

This field contains the name of the host that is running the WebLogic Server hosting the Web Service. As a rule, the server host is assigned to `localhost`, since this Web Services Expert uses the file system access to WebLogic 6.1 application server, and therefore the server runs on the same computer.

Server port number

This field contains the port number of WebLogic Server. The default server port is 7001.

Web Service context

This field contains the context root of the Web Services. You can use this value to access the deployed Web Service with a URL.

Protocol

This field contains the protocol that clients use to access the Web Service. There are two possible values: *http* or *https*. The default value is *http*.

Server root directory

This field contains the path to the directory that includes the WebLogic startup file, config files for your applications, etc.

See also

Deployment using the Web Services Expert

Index

Apache	99
audits.....	74
blueprint	15
breakpoints.....	12
classpath	63
code	15
configuration.....	68
Context Help Dialog	21
copyright.....	8
customization	68
DBMS config.....	22
dialogs 10, 12, 13, 15, 21, 22, 27, 29, 33, 35, 39, 41, 43, 46, 48, 62, 63, 68, 69, 72, 76, 83, 89, 91, 92, 93	
documentation	
Together	5
DTD	27
expert	15, 22, 35, 43, 48
export	27
extract method	32
find	83
generate	39, 41
HTML	41
Inspector	
inspector property builder.....	46
JDBC	22, 35, 43
link.....	10
metrics.....	76
model	91
options.....	68
pattern	13
printing	69, 70, 72
profile	19
refactor	13
refactoring	32
rename	80
replace	83
search	83
template	15
trademark.....	8
version control.....	93
view management.....	92
web service	99