

[Next](#) [Up](#) [Previous](#) [Index](#)

**Next:** [How do I use](#) **Up:** [Getting started with ECLiPSe](#) **Previous:** [How do I run](#) [Index](#)

## Subsections

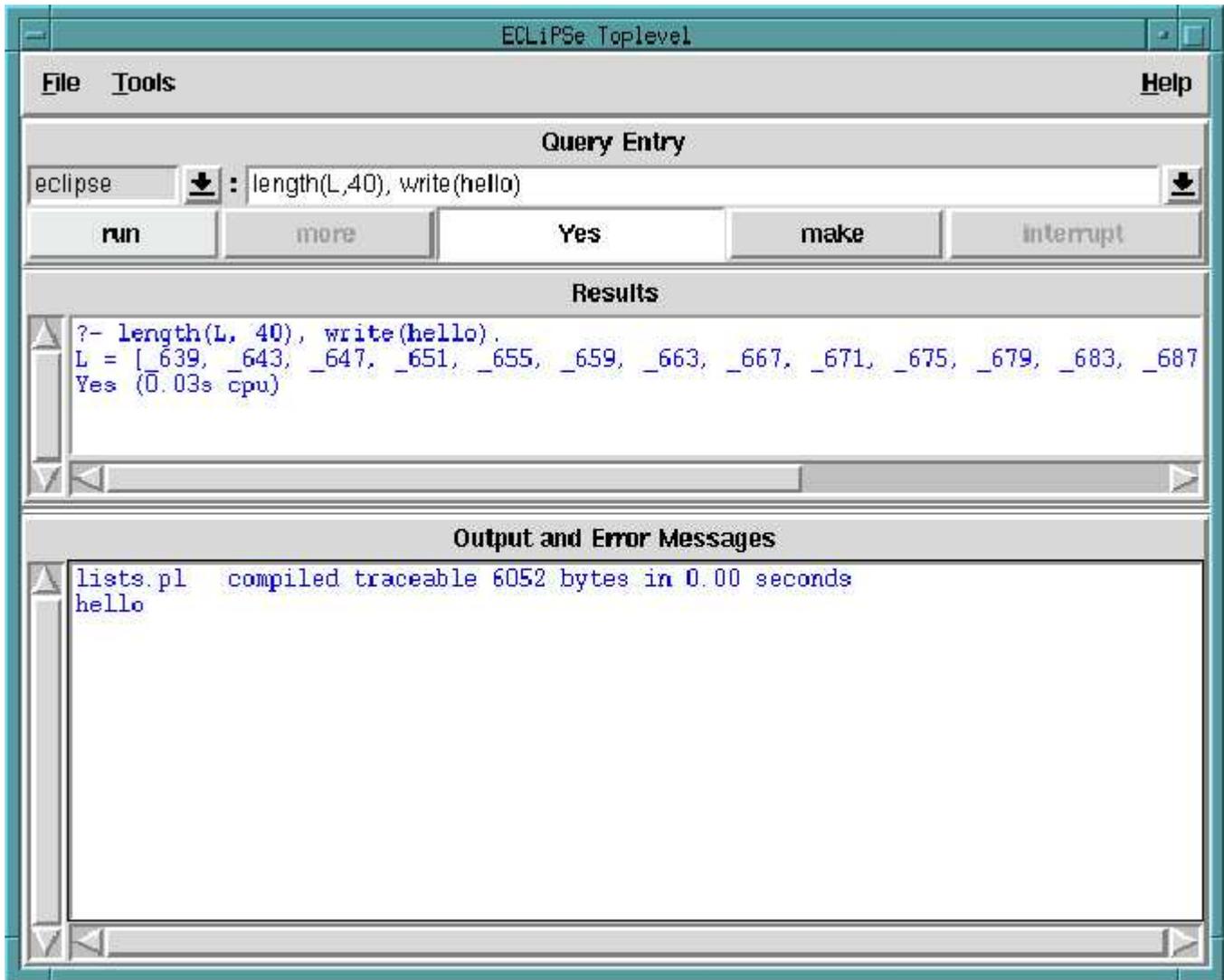
- [Getting started](#)
- [Compiling a program](#)
- [Executing a query](#)
- [Editing a file](#)
- [Debugging a program](#)
- [Getting help](#)
- [Other tools](#)
  - [Compile scratch-pad](#)
  - [Source File Manager](#)
  - [Predicate Browser](#)
  - [Source Viewer](#)
  - [Delayed Goals](#)
  - [Tracer](#)
  - [Inspector](#)
  - [Global Settings](#)
  - [Statistics](#)
  - [Simple Query](#)
  - [Library Help](#)
- [Preference Editor](#)

---

# How do I use tkeclipse?

## Getting started

To start TkECLiPSe, either type the command `tkeclipse` at an operating system command-line prompt, or select TkECLiPSe from the program menu on Windows. This will bring up the TkECLiPSe top-level, which is shown in Figure [3.1](#).

**Figure: TkeCLiPSe top-level**

Note that help on TkeCLiPSe and its component tools is available from the `Help` menu in the top-level window. If you need more information than can be found in this manual, try looking in the `Help` menu.

## Compiling a program

From the `File` menu, select the `Compile ...` option. This will bring up a file selection dialog. Select the file you wish to compile, and click on the `Open` button. This will compile the file and any others it depends on. Messages indicating which files have been compiled and describing any errors encountered will be displayed in the bottom portion of the TkeCLiPSe window (`Output and Error Messages`).

If a file has been modified since it was compiled, it may be recompiled by clicking on the `make` button. This recompiles any files which have become out-of-date.

For more information on program compilation and the compiler, please see chapter [6](#).

## Executing a query

To execute a query, first enter it into the `Query Entry` text field. You will also need to specify which module the query should be run from, by selecting the appropriate entry from the drop-down list to the left of the `Query Entry` field. Normally, the default selection of `eclipse` will be fine; this will allow access to all ECLiPSe built-ins and all predicates that have not explicitly been compiled into a different module. Selecting

another module for the query is only needed if you wish to call a predicate which is not visible from the `eclipse` module, in which case you need to select that module. (For more information about the module system, please see chapter [7](#).)

To actually execute the query, either hit the `Enter` key while editing the query, or click on the `run` button. `TkECLiPSe` maintains a history of commands entered during the session, and these may be recalled either by using the drop-down list to the right of the `Query Entry` field, or by using the up and down arrow keys while editing the `Query Entry` field.

If `ECLiPSe` cannot find a solution to the query, it will print `No` in the `Results` section of the `TkECLiPSe` window. If it finds a solution and knows there are no more, it will print it in the `Results` section, and then print `Yes`. If it finds a solution and there may be more, it will print the solution found as before, print `More`, and enable the `more` button. Clicking on the `more` button tells `ECLiPSe` to try to find another solution. In all cases it also prints the total time taken to execute the query.

Note that a query can be interrupted during execution by clicking on the `interrupt` button.

## Editing a file

If you wish to edit a file (e.g. a program source file), then you may do so by selecting the `Edit ...` option from the `File` menu. This will bring up a file selection dialog. Select the file you wish to edit, and click on the `Open` button.

When you have finished editing the file, save it. After you've saved it, if you wish to update the version compiled into `ECLiPSe` (assuming it had been compiled previously), simply click on the `make` button.

You can change which program is used to edit your file by using the `TkECLiPSe` Preference Editor, available from the `Tools` menu.

## Debugging a program

To help diagnose problems in `ECLiPSe` programs, `TkECLiPSe` provides the tracer. This can be invoked by selecting the `Tracer` option from the `Tools` menu. The next time a goal is executed, the tracer window will become active, allowing you to step through the program's execution and examine the program's state as it executes.

The tracer displays the current call stack and a trace log. By using the left mouse button in the `Call Stack` region of the tracer window, you can bring up a menu of additional operations you can perform on that goal, such as inspecting it, or setting a spy point on the predicate in question. Selecting `Configure filter ...` from the `Options` menu of the tracer will launch the conditional filter. This filter allows you to specify conditions on which the tracer should stop at a debug port. This can be very useful for skipping over unwanted debug ports.

For more information on using the tracer, please see the online help, available by selecting `Tracer Help` from the `Help` menu.

Other `TkECLiPSe` tools which are useful while using the tracer are:

- the predicate browser (available by selecting the `Predicate Browser` option from the `Tools` menu), which is useful for setting or removing spy points on predicates, or for setting the `start_tracing` flag which activates the tracer when a particular predicate is called for the first time; and
- the term inspector (available by double left clicking on a term from the stack window, or by selecting the `Inspector` option from the `Tools` menu), which is useful for examining and browse the arguments of a term in detail.

- the delayed goals browser (available by selecting the `Delayed Goals` option from the `Tools` menu), which allows you to inspect the current list of delayed goals.
- the display matrix (available either from calls in user's code, or by interactively selecting terms to be observed from the inspector, tracer or delay goals tools), which allows you to monitor any changes to a term and its arguments.

More information about debugging in ECLiPSe may be found in chapter [14](#).

## Getting help

More detailed help than is provided here can be obtained online for all the features of TkeCLiPSe. Simply select the entry from the `Help` menu on TkeCLiPSe's top-level window which corresponds to the topic or tool you are interested in.

## Other tools

TkeCLiPSe comes with a number of useful tools. Some have been mentioned above, but here is a more complete list. Note that we only provide brief descriptions here; for more details, please see the online help for the tool in question.

### Compile scratch-pad

This tool allows you to enter small amounts of program code and have it compiled. This is useful for quick experimentation, but not for larger examples or programs you wish to keep, since the source code is lost when the session is exited.

### Source File Manager

This tool allows you to keep track of and manage which source files have been compiled in the current ECLiPSe session. You can select files to edit them, or compile them individually, as well as adding new files.

### Predicate Browser

This tool allows you to browse through the modules and predicates which have been compiled in the current session. It also lets you alter some properties of compiled predicates.

### Source Viewer

This tool attempts to display the source code for predicates selected in other tools.

### Delayed Goals

This tool displays the current delayed goals, as well as allowing a spy point to be placed on the predicate and the source code viewed.

### Tracer

As discussed in section [3.3.5](#), the tracer is useful for debugging programs. See also chapter [14](#).

### Inspector

This tool provides a graphical browser for inspecting terms. Goals and data terms are displayed as a tree structure. Sub-trees can be collapsed and expanded by double-clicking. A navigation panel can be launched

which provides arrow buttons as an alternative way to navigate the tree.

Note that while the inspector window is open, interaction with other TkECLiPSe windows is disallowed. This prevents the term from changing while being inspected. To continue TkECLiPSe, the inspector window must be closed.

## Global Settings

This tool allows the setting of some global flags governing the way ECLiPSe behaves. See also the documentation for the [set\\_flag/2](#) and [get\\_flag/2](#) predicates.

## Statistics

This tool displays some statistics about memory and CPU usage of the ECLiPSe system, updated at regular intervals. See also the documentation for the [statistics/0](#) and [statistics/2](#) predicates.

## Simple Query

This tool allows the user to send a simple query to ECLiPSe even while ECLiPSe is running some program and the Toplevel Query Entry window is unavailable. Note that the reply is shown in EXDR format (see the ECLiPSe Embedding and Interfacing Manual).

## Library Help

This tool allows you to browse the online help for the ECLiPSe libraries. On the left is a tree display of the libraries available and the predicates they provide.

- Double clicking on a node in this tree either expands it or collapses it again.
- Clicking on an entry displays help for that entry to the right.
- Double clicking on a word in the right-hand pane searches for help entries containing that string.

You can also enter a search string or a predicate specification manually in the text entry box at the top right. If there is only one match, detailed help for that predicate is displayed. If there are multiple matches, only very brief help is displayed for each; to get detailed help, try specifying the module and/or the arity of the predicate in the text field.

## Preference Editor

This tool allows you to edit and set various user preferences. This include parameters for how TkECLiPSe will start up, e.g. the amount of memory it will be able to use, and a initial query to execute; and parameters which affects the appearance of TkECLiPSe, such as the fonts TkECLiPSe uses.

---

[Next](#) [Up](#) [Previous](#) [Index](#)

**Next:** [How do I use](#) **Up:** [Getting started with ECLiPSe](#) **Previous:** [How do I run](#) [Index](#)

*Warwick Harvey*  
2003-09-20