

Debugging a Module



Access

To begin debugging a module, press **Enter** in the [Start ILE Adelia debugger \(HA#STRIDBG\)](#) screen.

The module shown is the entry procedure module of the program being debugged.

Note: If the source code of the program being debugged is found in the first QDBGSRC file found in the session's list of libraries, the 4GL source for the module's verification list of the program is displayed.

Otherwise, the module debug screen will appear without the source.

```
Debug a module
Program . : WRKCUS      Library . . : TSTADELIA8      Module : WRKCUS
1.00 * WORK WITH CUSTOMERS
2.00 *-----
3.00 INITIALIZE 01
4.00 PROCESS 01
5.00 *-----
6.00 TRANSACTION 01
7.00 IF *F03 OR *F12
8.00     TERMINATE
9.00 END
10.00 VERIFY 01
11.00 INITIALIZE 02
12.00 PROCESS 02
13.00 *-----
14.00 TRANSACTION 02
+
Debug
F3=Exit  F4=Prompt  F6=Add/Clear breakpoint  F9=Retrieve  F10=Step
F11=Display variable  F12=Resume  F13=Work with breakpoints  F24=More keys
```

Function keys

Note: Function keys F6 and F11 are only available for modules that have source.

F3 Stop the source debug program and return to previous screen. If the program is running, it will be stopped. However, the breakpoints set in the module will be stored and the session will continue in Debug mode.

F4 Prompt on the "Debug" field.

F6 Add/clear a breakpoint.

If there are no breakpoints on the line of the source where the cursor is placed, an unconditional breakpoint will be added. When you add a breakpoint, the corresponding line number is highlighted.

If the line already contained a breakpoint, it is instead deleted.

Note: You can also add and remove breakpoints directly, using the HA#BREAK and HA#CLEAR commands, respectively.

F9 Retrieves the last command used.

If the "Debug" field is empty, pressing F9 recalls the last validated command in the field.

If the "Debug" field is not empty, pressing F9 lets you scroll up the list of the 99 most recent commands.

F10 Step by step execution of the displayed program.

F11 The "Debug" field will show the current value of the variable over which the cursor is placed.

Note: You can also display the value of a variable by entering the HA#EVAL command directly.

- F12** Stop the source's debug program and return to previous screen. If a program is already running, it will continue. The breakpoints set in the module will be stored and the session will continue in Debug mode.
- F13** [Work with the module's breakpoints.](#)
- F14** [Work with modules](#) in order to select the current module.
- F15** [Display the module's data list](#) (variables and views).
If you select a data item, it will be inserted on the "Debug" line, at the cursor's current position.
- F16** Starting at the current cursor position, searches for the string displayed in the "Debug" line.
- F17** Moves to the top of the source.
- F18** Moves to the bottom of the source.
- F21** Displays a [window for entering commands](#).
- Enter** Validate data entry.
- Roll Up /Dn**

Displayed data

Program

Name of the program being debugged.

Library

Name of the library containing the program being debugged.

Module

Name of the current module.

Line

Line numbers for which a breakpoint has been set, are shown highlighted.

The line where the program stopped is shown highlighted.

Debug

When **F11** is pressed, the values of any variables will be shown on this line, provided the data to be displayed are not too long.

Data entry

Debug

Debug command to be run.

Press **F4** to display the list of available commands.