

# Starting and Leaving the Debugger

This chapter covers the following topics:

- Preparing to Use the Debugger
  - Starting the Debugger
  - Restarting the Debugger
  - Leaving the Debugger
- 

## Preparing to Use the Debugger

To exploit the full functional scope of the debugger, you must set the parameter `SYMGEN` to "ON". You can set this parameter in one of the following ways:

- dynamically when starting Natural,
- only for the current session by changing the session parameters, or
- in your parameter file using the Configuration Utility.

When you catalog or stow an object and `SYMGEN` is set to "ON", a symbol table is generated as part of the generated program. Since this table contains the information relevant to the variables active for this object, variables cannot be accessed without `SYMGEN` being specified, although it is still possible to debug the object.

### Note:

It is not necessary to set the parameter `SYMGEN` when debugging in a SPoD environment on a mainframe.

## Starting the Debugger

The debugger can be used with stowed or cataloged Natural programs and dialogs. It can be used in the local environment and in the remote environment.

See also the description of the system command `DEBUG`.

### To start the debugger

1. Open the editor for the object that is to be debugged.

Or:

Select the object in the library workspace.

2. From the **Debug** menu, choose **Start**.

Or:

Press CTRL+F7.

Or:

When the Debug toolbar is shown, choose the following toolbar button:



Or:

When you have selected an object in the library workspace, invoke the context menu and choose **Debug**.

When the editor for the selected object has not yet been opened, it is opened now.

For a dialog, the dialog source is now shown in a separate window.

When the debugger has been started, additional elements are available in the Natural Studio window. See *Elements of the Debugger* for further information.

## Restarting the Debugger

When you restart your debugging session, the debugger repositions to the beginning of the application while all your current settings for breakpoints, watchpoints and watchvariables are kept. Thus, restarting a debugging session is useful if you want to rerun your application without having to specify the settings relevant for debugging again.

### ▶ To restart the debugger

- From the **Debug** menu, choose **Restart**.

Or:

Press CTRL+SHIFT+F7.

Or:

When the Debug toolbar is shown, choose the following toolbar button:



## Leaving the Debugger

The debugger is terminated automatically if the application ends without an error. You can also stop the debugger before it terminates automatically; see the description below.

### **Note:**

Closing the editor window does not stop the debugger.

When the debugger is terminated or stopped, your breakpoint, watchpoint and watchvariable settings are automatically stored. All these settings will be restored the next time you start the debugger.

In the case of an error, the corresponding source is displayed and the trace position indicates the line which caused the error. A message window appears with the appropriate error message and a choice to either continue or end the debugging session. Continuing the debugging session may be useful, for example, if your application contains any error processing (including error transactions) or if you want to display any variables before you end your debugging session.

▶ **To stop the debugger**

- From the **Debug** menu, choose **Stop**.

Or:

Press SHIFT+F7.

Or:

When the Debug toolbar is shown, choose the following toolbar button:



The debugging session is terminated and control is returned to Natural.