Hello World!

This chapter contains the following exercises:

- Creating a Program
- Running a Program
- Correcting Program Errors
- Stowing a Program
- Setting the Workspace Options

Creating a Program

You will now write your first short program which displays "Hello World!". It will be stored in the library you have created previously.

To create a new program

1. In the library workspace, select the library named `TUTORIAL`.

2. From the context menu, choose **New Source > Program**.

   Or:
   Choose the following toolbar button:

   The program editor appears. It is currently empty.

3. Enter the following code in the program editor:

   ```
   * The "Hello world!" example in Natural.
   *
   DISPLAY "Hello world!"
   END /* End of program
   ```

   Comment lines start with an asterisk (*) followed by at least one blank or a second asterisk. When you forget to enter the blank or second asterisk, Natural assumes that you have specified a system variable; this will result in an error.

   If you want to insert empty lines in your program, you should define them as comment lines. This is helpful, if you want to access your program from different platforms (Windows, mainframe, UNIX or OpenVMS). With the mainframe version of Natural, for example, the default is that empty lines are automatically deleted when you press ENTER.

   You can also insert comments at the end of a statement line. In this case, the comment starts with a slash followed by an asterisk (/*).

The text that is to be shown in the output is defined with the `DISPLAY` statement. It is enclosed in quotation marks.

The `END` statement is used to mark the physical end of a Natural program. Each program must end with `END`.

Running a Program

The system command `RUN` automatically invokes the system command `CHECK` which checks the program code for errors. If no error is found, the program is compiled on the fly and then executed.

Notes:

1. The system commands are also available with the mainframe version of Natural. Under Windows, they are invoked by choosing the corresponding command from the **Object** menu.
2. `CHECK` is also available as a separate command in the **Object** menu.
3. Natural also provides the system command `EXECUTE` which uses the stowed version of your program (stowing a program is explained later in this tutorial). In contrast to this, the `RUN` command always uses your latest modifications to the program.

To run a program

1. From the **Object** menu, choose **Run**.

   Or:
   Choose the following toolbar button:

   When your code is syntactically correct, the output contains the text you have defined.

```
Page      1                                                   09-06-30  12:07:25


Hello world!
```

2. Press ENTER to return to the program editor.

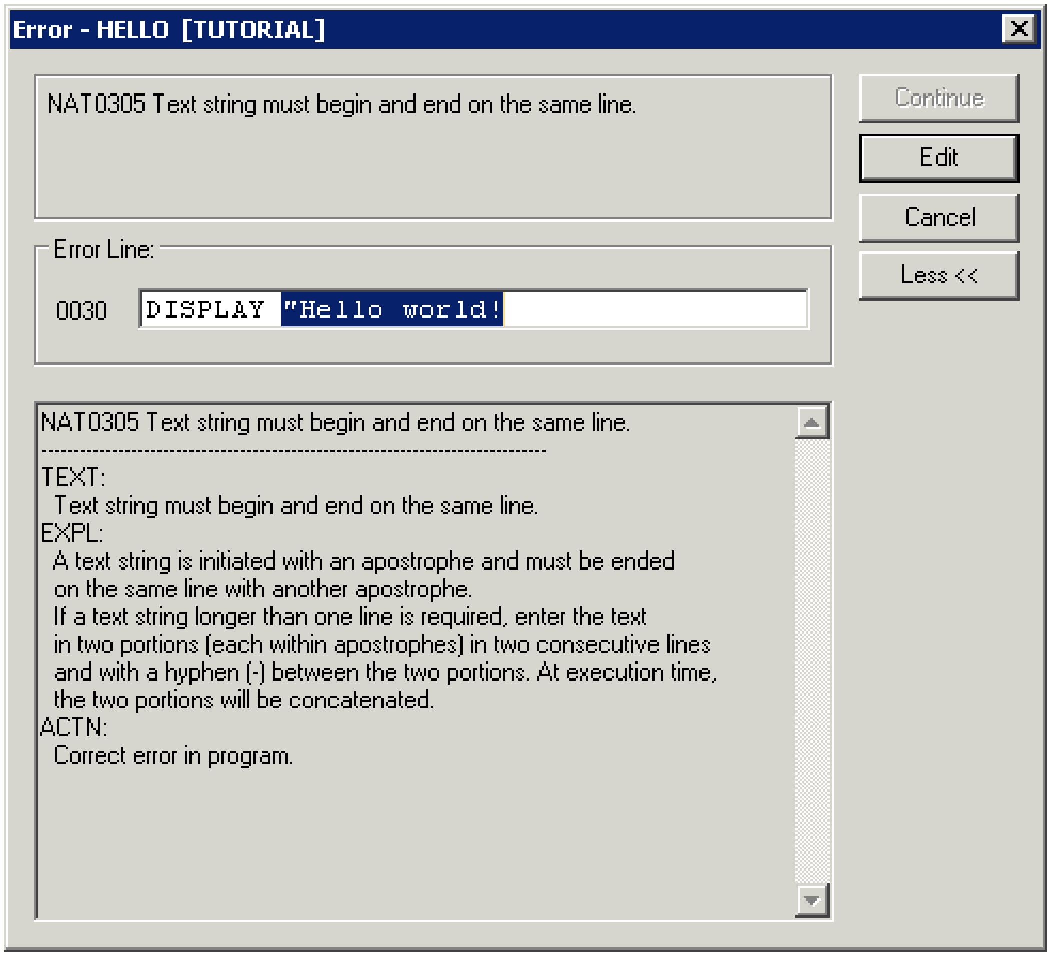
Correcting Program Errors

You will now create an error in your Hello World program and then run the program once more.

To correct an error

1. Delete the second quotation mark in the line containing the `DISPLAY` statement.
2. Run the program once more as described above.

   When the error is found, a dialog box appears, providing information on the error.



Error - HELLO [TUTORIAL]

NAT0305 Text string must begin and end on the same line.

Error Line:

```
0030 DISPLAY "Hello world!
```

```
NAT0305 Text string must begin and end on the same line.
-----------------------------------------------------------------------
TEXT:
  Text string must begin and end on the same line.
EXPL:
  A text string is initiated with an apostrophe and must be ended
  on the same line with another apostrophe.
  If a text string longer than one line is required, enter the text
  in two portions (each within apostrophes) in two consecutive lines
  and with a hyphen (-) between the two portions. At execution time,
  the two portions will be concatenated.
ACTN:
  Correct error in program.
```

3. Correct the error in the dialog box, that is: insert the missing quotation mark at the end of the line.
4. Choose the **Continue** button to find the next error.

   In this case, no more errors are found and the output is shown.

5. Press ENTER to return to the program editor.

Note:
Instead of choosing the **Continue** button, it is also possible to choose the **Edit** button. The dialog box is then closed and you can correct the error directly in the program editor.

Stowing a Program

When you stow a program, it is compiled and both source code and a generated program are stored in the Natural system file.

Like the `RUN` command, the system command `STOW` automatically invokes the `CHECK` command. A program is only stowed when it is syntactically correct.

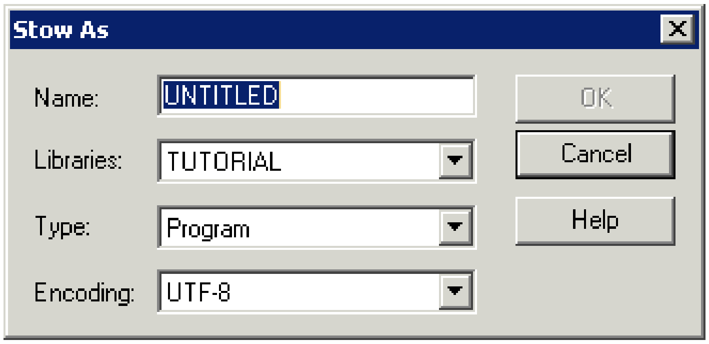
Note:
If you want to save the changes to your program, even if the program contains a syntactical error (for example, if you want to suspend your work until the next day), you can use the system command `SAVE` which can be invoked from the **Object** menu.

To stow a program

1. From the **Object** menu, choose **Stow**.

   Or:
   Choose the following toolbar button:

   Since your program has not yet been saved, the **Stow As** dialog box appears.

   Name: UNTITLED
   Libraries: TUTORIAL
   Type: Program
   Encoding: UTF-8

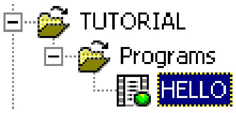
   The name of the currently selected library is automatically provided in the corresponding drop-down list box.

2. Specify the name "HELLO" in the **Name** text box.

3. Choose the **OK** button.

   A message is now shown, informing you that stowing was successful. This message is either shown in the status bar or in a dialog box, depending on the setting of a specific workspace option (see below).

In the library workspace, a new node named **Programs** appears below the **TUTORIAL** node. This node contains the program you have just stowed.



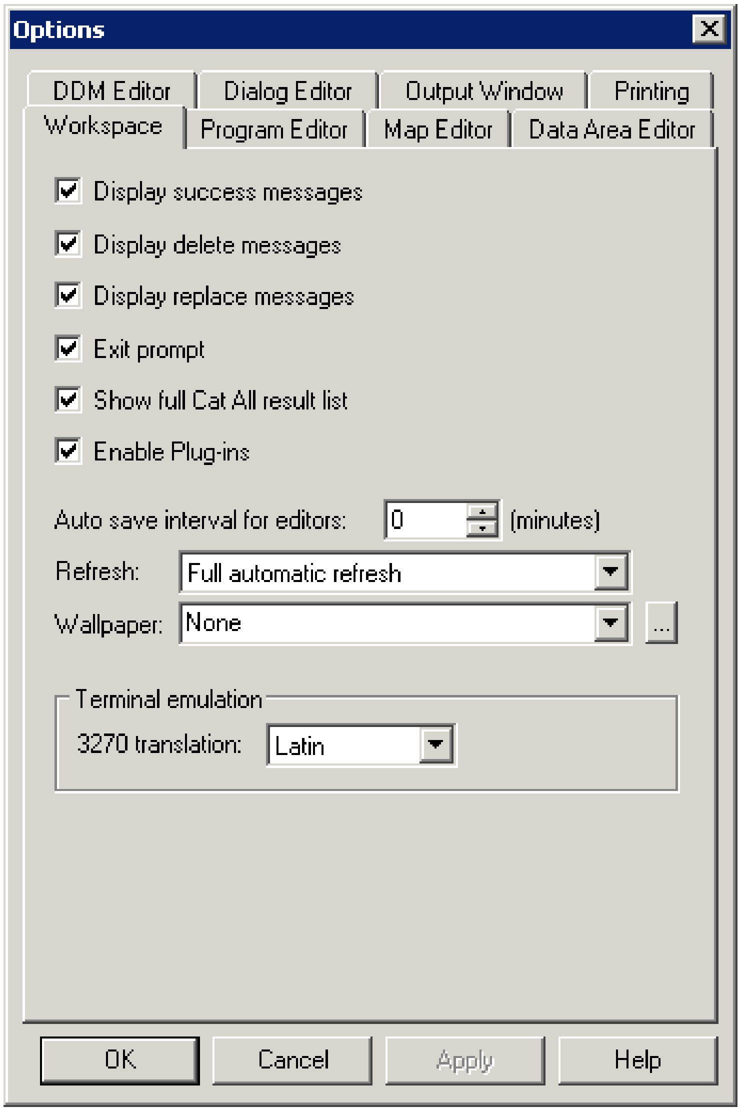
The green dot on the program icon indicates that both source code and a generated program are available for the object.

Setting the Workspace Options

You will now check the settings of your workspace options.

▶ **To check the workspace options**

1. From the **Tools** menu, choose **Options**.
2. In the resulting **Options** dialog box, display the **Workspace** page.



☑ Display success messages

☑ Display delete messages

☑ Display replace messages

☑ Exit prompt

☑ Show full Cat All result list

☑ Enable Plug-ins

Auto save interval for editors: 0 (minutes)

Refresh: Full automatic refresh

Wallpaper: None

Terminal emulation

3270 translation: Latin

3. When success messages are to be shown in a dialog box, make sure that the corresponding check box is selected.

   Note:
   Whether line numbers are shown in the program editor is controlled by an option on the **Program Editor** page.

4. Choose the **OK** button to save your changes and to close the dialog box.

You can now proceed with the next exercises: *Database Access*.