Appendix

C Host Control

Executing Scripts and Commands

Session allows a host computer to execute an entire script or almost any individual TermTalk command using the host control facility.

The following escape sequence executes the script specified, which must be compiled previously either by using the `compile` command or by executing the script while running Session interactively.

```
ESC & o x <compiled script file name> <CR>
```

To compile and execute a TermTalk object under NewWave, the following sequence is used.

```
ESC & o X <script object name> <CR>
```

To execute an individual command, you send the text of the command to Session in an escape sequence. Session then compiles and executes the command. The following escape sequence executes the command specified, but does not return command completion status to the host program. The host must assume that the command was performed successfully.

```
ESC & o C <command> <CR>
```

The following escape sequence executes the command specified, and returns completion status to the host program.

```
ESC & o B <command> <CR>
```

Session indicates success or failure of the command by returning the standard HP terminal completion codes of `S` (for successful) or `F` (for failed) followed by a carriage return. Failure to perform a command could be caused by a syntax error in the command, use of variables or restricted commands in the escape sequence, or the command could fail for its own reasons (file not found, for example). The completion codes are returned using a type 3 block transfer (with handshaking). See "Testing the Status" below.

The text of the command must evaluate to a complete command expression. It must contain less than 256 characters. Partial commands are not allowed. The command cannot contain variable references. Procedure calls are not allowed, though `do script` commands can be used. Function calls are also not allowed, since they return a value. `if` statements and `while` loops are not allowed in host control escape sequences, since they are not contained in one line.

The following examples show how this facility can be used:

```
ESC & o B set baud to 1200 <CR>
```

This escape sequence sets the baud rate for the session to 1200. Session would return an `S` character to the host program indicating successful completion.

```
ESC & o B display "A host message" <CR>
```

This escape sequence displays the string on the terminal screen. Session would return an `S` character to the host program indicating successful completion.

```
ESC & o x "myscript.ttx" <CR>
```

This escape sequence executes the script specified. This allows the compilation of the host control string to be bypassed, directly performing the compiled script. This results in much better performance.

```
ESC & o X "My Script Object" <CR>
```

This escape sequence compiles and executes the NewWave script object specified.

File Transfer

To provide the host computer with control over NewWave Object and MS-DOS file transfer, the following escape sequences are used.

NewWave

To send a NewWave object to the host HP 3000:

```
ESC & o s <nn> h <mpe name> N
```

where <nn> is the length in characters of the MPE file name.

On receiving this sequence Session displays a dialog box asking the user to drag an object onto the Session window for transfer to the host. The user can move both the dialog box and the main Session window in case they are covering the object he wishes to upload; a serialized copy of the object is uploaded. The original object remains on the user's workstation on completion of the transfer.

To receive a NewWave object from the host HP 3000:

```
ESC & o r <nn> h <mpe name> N
```

where <nn> is the length in characters of the MPE file name. No user interaction is required during the execution of this sequence. The named MPE file is downloaded and imported as a serialized object file.

MS-DOS

To send an MS-DOS file to the host HP 3000:

```
ESC & o s <nn> h <mpe name> Z
```

where <nn> is the length in characters of the MPE file name. This sequence displays the normal MS-DOS file selection dialog, in which the user selects a file to be uploaded. Once the file has been selected the Send File dialog box is displayed. The only difference is that the user cannot change the MPE file name displayed.

To receive an MS-DOS file from the host HP 3000:

```
ESC & o r <nn> h <mpe name> Z
```

or

```
ESC & o r <mm> w <msdos name> <nn> h <mpe name> Z
```

where <nn> is the length in characters of the MPE file name, and <mm> is the length of the MS-DOS file name text <msdos name>. On receiving one of

these sequences, Session starts up the Tymlink host program, then prompts the user for the MS-DOS file name into which the MPE file will be transferred.

The second form of the sequence specifies the name of the MS-DOS file as well as that of the MPE file. The file selection dialog is displayed with this file name shown as the default.

Extended versions of the send and receive escape sequences provide the ability to override default file conversion options:

```
ESC & o r <o> c <nn> h <mpe name> Z
```

or

```
ESC & o s <o> c <nn> h <mpe name> Z
```

The `<o>` parameter specifies the conversion option to be used and may be one of the following:

0 = Default according to file type

1 = Text

2 = Binary

3 = Wordwrap

4 = Backup

5 = Restore

When overriding the default conversion option, take care not to specify a conversion type which is unavailable for the type of file transferred. Specifying an incompatible conversion option will cause the transfer to terminate without sending any data.

The following chart shows the options available for various file types; the asterisked option is the default. "Backup" type files are created using the `backup` conversion option.

Option	Text	Backup	Other
`text`	Yes*	No	No
`binary`	Yes	Yes	Yes
`wordwrap`	Yes	No	No
`backup`	Yes	No	Yes*
`restore`	No	Yes*	No

When using host control file transfer sequences in HP Desk scripts, please note the following:

1. The script will need to swallow the `RUN TYMLINK.PUB.SYS` command line, including its carriage return, sent in response to the escape sequence. If the user cancels the transfer action before this point, the string `CANCEL` will be received by the script instead. The script should check the string and either invoke Tymlink on Session's behalf or exit the script if `CANCEL` was sent.
   Note that Session checks to ensure that the run string is reflected to the terminal as it is typed. The function used to read the terminal data should therefore be a normal read line with echo activity that terminates on receipt of the carriage return.

2. The script can use the `Esc&oS` status sequence (see below) to find out the name of the uploaded object or MS-DOS file, and can use this as the item name when importing the MPE file into HP Desk.

Printing Spooled Data

To print a spooled log:

```
ESC & o pP
```

This has the same effect as selecting the **Print Spooled** command from the Log menu. It has no effect if logging is not already underway. It does not stop logging.

Testing the Status

To check the result of any of the above sequences:

```
ESC & o S
```

A string terminated by a carriage return is returned to the host. The first character indicates whether or not the previous `Esc&o` action was successful with one of the following codes.

S Successful

F Failed

C Cancelled by the user

X `ESC & o` sequence invalid

For file transfer sequences, this status character is followedby the MS-DOS file name or NewWave object name; for example, "July Market Assessment" or `EXPENSE.XLS`.

To determine what terminal model is in use, returning the same result produced by the TermTalk `identity()` function, use the following sequence:

```
ESC * s 811 ^
```