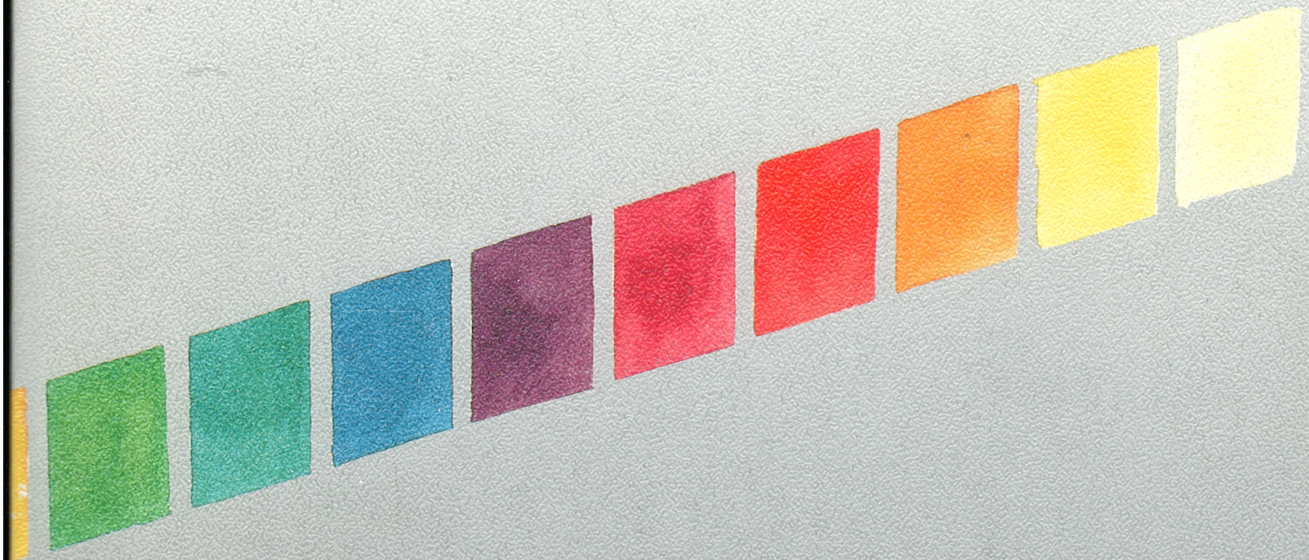


# TELEFIX
# Reference Manual

ND-30.040.2 EN

Norsk Data

# TELEFIX
# Reference Manual

ND-30.040.2 EN

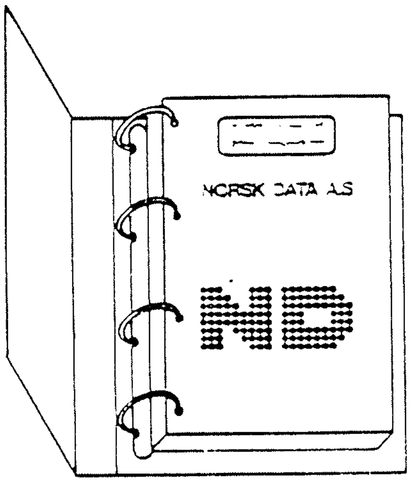
# NOTICE

The information in this document is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this document. Norsk Data A.S assumes no responsibility for the use or reliability of its software on equipment that is not furnished or supported by Norsk Data A.S.

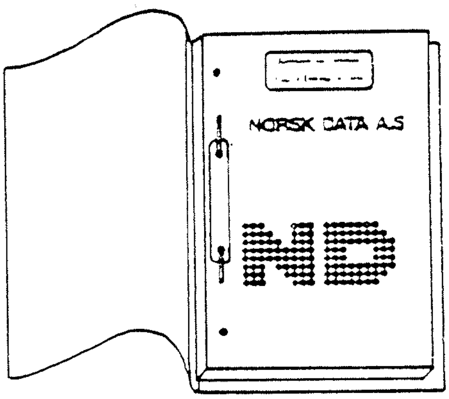
This manual is in loose-leaf form for ease of updating. Old pages may be removed and new pages easily inserted if the manual is revised.

The loose-leaf form also allows you to place the manual in a ring binder (A) for greater protection and convenience of use. Ring binders with 4 rings corresponding to the holes in the manual may be ordered in two widths, 30 mm and 40 mm. Use the order form below.

The manual may also be placed in a plastic cover (B). This cover is more suitable for manuals of less than 100 pages than for large manuals. Plastic covers may also be ordered below.



A: Ring Binder



B: Plastic Cover

Please send your order to the local ND office or (in Norway) to:

**Norsk Data A.S**
Graphic Center
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

## ORDER FORM

I would like to order

...... Ring Binders, 30 mm, at nkr 20,- per binder

...... Ring Binders, 40 mm, at nkr 25,- per binder

...... Plastic Covers at nkr 10,- per cover

Name ..........................................................................................................

Company ....................................................................................................

Address ......................................................................................................

..................................................................................................................

City ............................................................................................................

# PRINTING RECORD

| Printing | Notes |
|---|---|
| 09/84 | VERSION 1 EN |
| 10/85 | VERSION 2 EN |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

TELEFIX Reference Manual
Publ.No. ND-30.040.2 EN

Norsk Data

Norsk Data A.S
Graphic Center
P.O.Box 25, Bogerud
0621 Oslo 6, Norway

Manuals can be updated in two ways, new versions and revisions. New versions consist of a complete new manual which replaces the old manual. New versions incorporate all revisions since the previous version. Revisions consist of one or more single pages to be merged into the manual by the user, each revised page being listed on the new printing record sent out with the revision. The old printing record should be replaced by the new one.

New versions and revisions are announced in the Customer Support Information (CSI) and can be ordered as described below.

The reader's comments form at the back of this manual can be used both to report errors in the manual and to give an evaluation of the manual. Both detailed and general comments are welcome.

These forms and comments should be sent to:

Documentation Department
Norsk Data A.S
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

Requests for documentation should be sent to the local ND office or (in Norway) to:

Graphic Center
Norsk Data A.S
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

# PREFACE

## THE PRODUCT

The TELEFIX system consists of the following products:

| | | |
|---|---|---|
| 1) | TELEFIX | ND 210772C |
| (replaces | TELEFIX-MODEM | ND 10772B) |
| 2) | TELEFIX-MINI | ND 210773C |
| (replaces | TELEFIX-MINI | ND 10773B) |
| 3) | TELEFIX-LOCAL | ND 210775C |
| (replaces | TELEFIX-REMOTE | ND 10775B) |
| 4) | TELEFIX FILES FOR USER SITES | ND 210375C |
| (replaces | TELEFIX-END-USER | ND 10375B) |

These products are internal ND tools for remote diagnosis and maintenance of ND hardware and software. They may also be used by product developers and testers.

## THE READER

The reader should be someone from ND involved in support, maintenance, development or testing of ND hardware and software.

## PREREQUISITE KNOWLEDGE

The user of this manual should be familiar with SINTRAN III and also the hardware and software to be tested.

## THE MANUAL

This manual describes the hardware and software installation needed to run Telefix, the various modes in which the Telefix programs may be run, and all of the Telefix commands. It covers the products listed above, with the exception of TELEFIX-MINI, which is described in a separate manual (ND-60.239 EN).

## RELATED MANUALS

TELEFIX-MINI Reference Manual ND-60.239 EN
All manuals for the systems you want to test.

## CHANGES FROM THE PREVIOUS VERSION

Areas in this manual which describe changes in Telefix between Releases B and C are framed by borders like the ones framing this note.

# TABLE OF CONTENTS

SECTION PAGE

Section Page

# LIST OF FIGURES

# CHAPTER 1

## INTRODUCTION

# 1 INTRODUCTION

Telefix is ND's remote diagnostic system. It gives you the ability to trace, diagnose, and correct faults in ND computers at any remote site, via the public telephone network.

Among the advantages offered by the use of Telefix are rapid response time to customer inquiries and enhanced access to expertise.

To make the connection possible, the remote site must be equipped with an ND 103750 Telefix Adaptor, connected to a modem attached to the public telephone network. A person at the site must call the ND Telefix center and establish the correct modem and ND 10375 connections.

This manual explains how to install a Telefix center, set up and communicate with the remote sites, and use the system in manual and automatic modes.

# CHAPTER 2

## *THE TELEFIX VERSIONS*

## 2 THE TELEFIX VERSIONS

The product "Telefix" consists of two related programs, TELEFIX (the main program, formerly called TELEFIX-MODEM) and TELEFIX-LOCAL (a subset of TELEFIX, formerly called TELEFIX-REMOTE).

The choice of program and mode (see pages 13-17 and page 53) will depend on the nature and magnitude of the problem to be investigated, or of the task to be carried out. The different programs and modes also represent a sort of escallation sequence - a natural progression to more powerful, and more complex tools. The typical uses of the programs are as follows:

TELEFIX

This is the version including all commands, and designed to work with asynchronous modem lines.

TELEFIX-LOCAL

This version is designed to work with internal SINTRAN Telefix access devices instead of physical lines. It allows you to run a remote process in the same system as Telefix runs. It can be installed in a remote system to be used either locally, or from a connected TELEFIX system, with the REMOTE-MODE command. The advantage of TELEFIX-LOCAL is that it permits the running of long automatic modes with no communication line connected during this time. The activated remote system is the one where this Telefix program runs itself.

It does not include the commands:

ACCOUNTING-ON-OFF
PRINT-ACCOUNTING
DEFINE-SINTRAN-PROMPT
SEND-FILE
GET-FILE
PLACE-BINARY
RELAY-MODE
REMOTE-MODE
CONTROL-MODE

No RT program is needed, nor the internal device, but SINTRAN must be generated with Telefix access devices.

This version may run with terminals of type 2 (paper-printer terminals). In this case, the following commands become unavailable:

```
LOG-ANALYZER
INITIALIZE-TERMINAL
DESCRIPTION-EDITOR
SCRIPT-EDITOR
```

TELEFIX-MINI

This is a stripped down version allowing you to run only a manual mode without a log file, and to transfer files. It is not discussed here, as a special manual (ND-60.239 EN) exists for it.

TELEFIX FILES
FOR USER SITES

(formerly
TELEFIX-END-USER)

This is the part of Telefix to be installed at the customer site by the customer. It includes two programs, one receiver and one transmitter program, and a mode file to be used when installing the programs.

> Note: The program TELEFIX-LINK is not supported in this Release of Telefix (Release C).

# CHAPTER 3

## INSTALLING A TELEFIX SYSTEM

# 3 INSTALLING A TELEFIX SYSTEM

## 3.1 GENERAL INSTALLATION CONSIDERATIONS

To be able to use Telefix for service purposes from any Telefix office to any customer, it is necessary to set up Telefix centers following some standard rules. Telefix includes some facilities to link centers, and to use any one as a relay to the next one.

The user who needs to be connected, via Telefix, to some experts has only to call the Telefix center closest to his/her site. If this local center is unable to provide the requested service, the connection already established should be left open. The local office can then forward the communication to another center, where support can be found.

To allow this transfer of calls and connections, a local Telefix center must be installed as follows:

1) The TELEFIX program is installed following the instructions found in this manual.

2) At least two asynchronous communication interfaces (terminal lines) are available, each connected to one modem. One line is generated as an RT terminal line and is used for the connection with customer systems. If it is not possible to handle all transmission speeds with the same modem, this line has to be duplicated for all the necessary modem types. We call this/these line(s) the Telefix line(s), connected to the Telefix modem(s).

3) The second line is generated as a terminal line with background and is used for the connection with other Telefix centers. It runs a modem compatible with the ones used by the other ND centers. We call this line the relay line, connected to the relay modem.

The relay line has several uses:

- To allow another center to log in on the Telefix machine in order to transfer files, or to start the Telefix program in remote mode.

- To transfer a remote system connection to another Telefix center (relay mode).

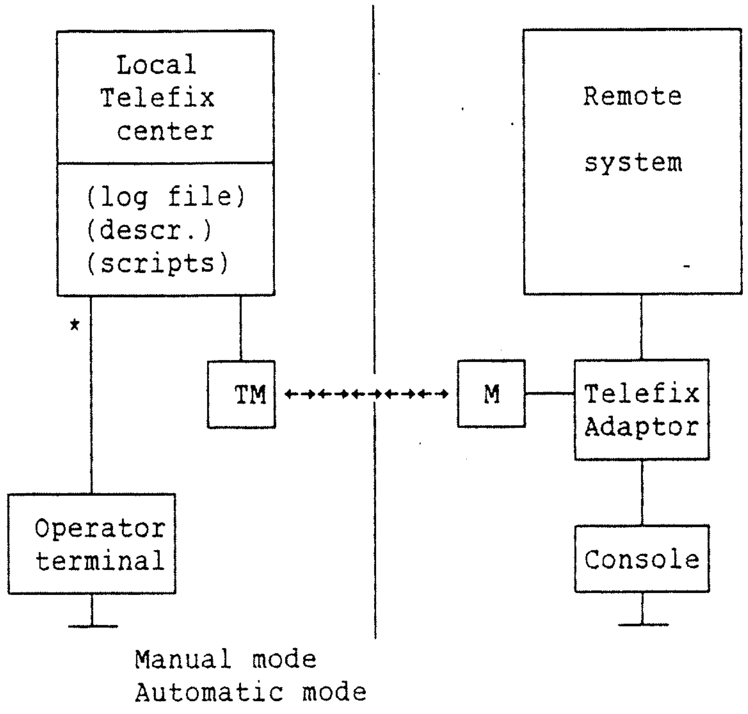
- As a follow-up terminal to the other center, which can then follow the operations.

In addition to these physical links between Telefix centers through asynchronous modems connected to the public networks, you can use already installed COSMOS connections to log in to one center from another one. But, care must be taken when using COSMOS to run Telefix.

The packet switching networks (X.25) are not suitable for such use, as Telefix works in an "always break, no echo" mode. In general, packet switching communications are not usable for interactive modes.

TELEFIX-LOCAL, not working against asynchronous interfaces but with internal SINTRAN Telefix access devices, may run in "local mode" to run the Telefix machine also as the remote system, or in "remote mode" called from a Telefix center. One of the interests for service departments is to install it at ND-500 configurations: the automatic mode to run ND-500 tests will be handled from the attached ND-100 and not from the Telefix center, in order to cut the communication line during long runs. In this case, the log files and all other needed files are located at the remote system. The automatic modes may also be started locally, on site, to run diagnostics.

Pages 13-17 show the possible connections between remote sites and Telefix centers.



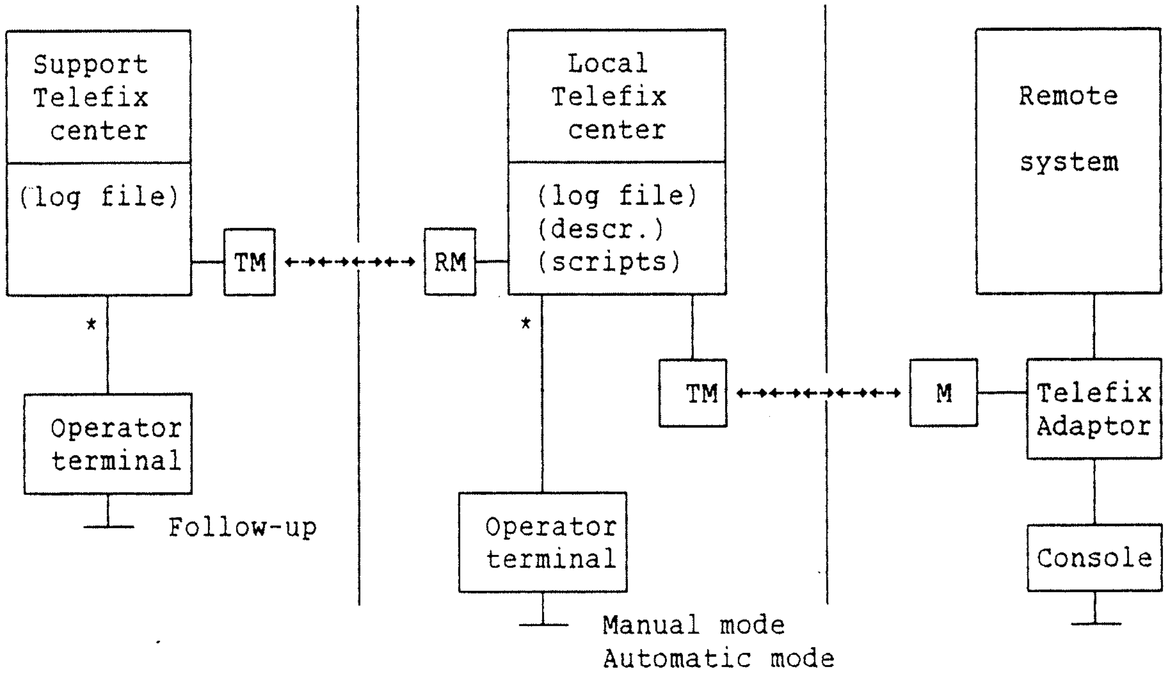
Key

M = Modem
TM = Telefix Modem
descr. = description

* = The terminal line(s) from which the Telefix program is activated

Fig. 1. Telefix remote - run from support center

The local Telefix center works from the operator terminal to the remote system, either in manual or automatic mode.



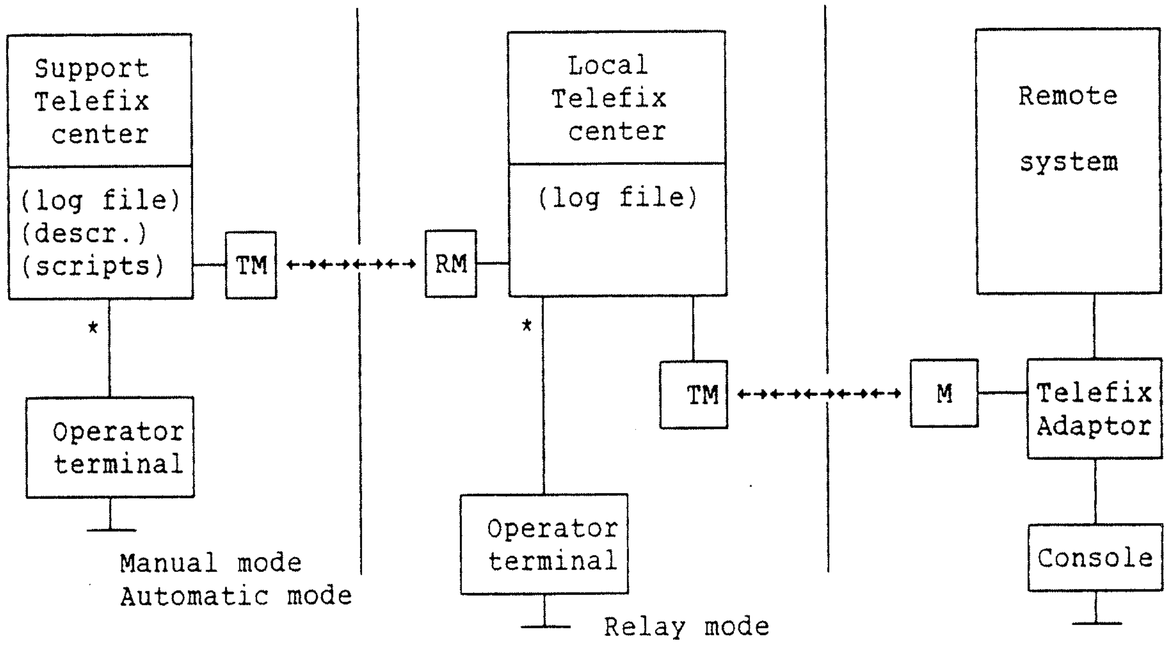
Key

M = Modem
TM = Telefix Modem
RM = Relay Modem
descr. = description

* = The terminal line(s) from which the Telefix program is activated

Fig. 2. Telefix support center follow-up

The local center has established the connection between its relay modem and the support-center Telefix modem. The operator at the local center has given his/her relay-modem line number as the follow-up terminal. The operator at the support center is in manual mode with his Telefix modem line, and follows on his/her terminal screen the operations running between the local center and the remote site.



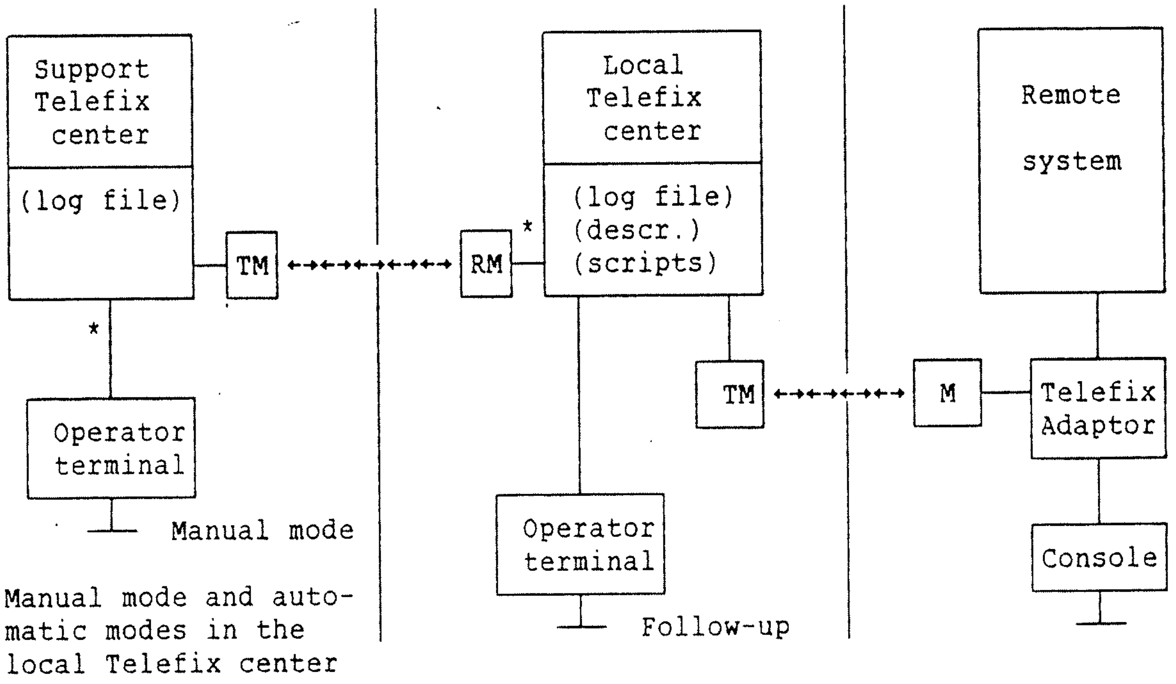
Key

M = Modem
TM = Telefix Modem
RM = Relay Modem
descr. = description

* = The terminal line(s) from which the Telefix program is activated

Fig. 3. Telefix relay mode

The local-center operator has stopped the follow-up function and entered relay mode on his relay-modem line number, to give the remote connection to the support center. The support-center operator terminal is directly connected to the remote system and may run manual and automatic modes. The local-center operator follows the line traffic between the support center and the customer on his/her terminal screen.



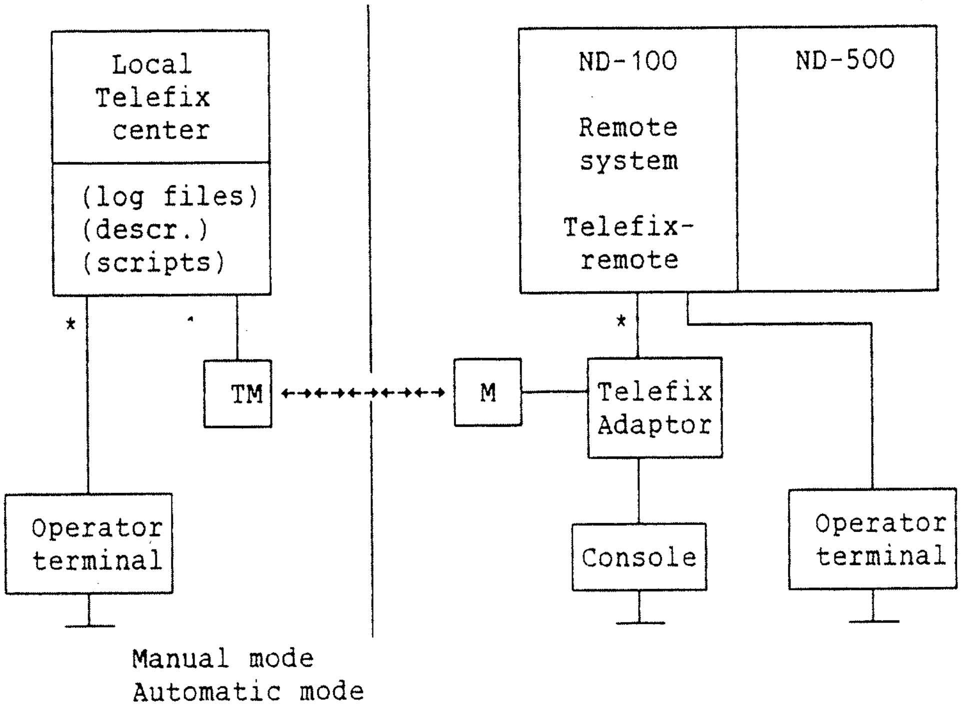
Key

M = Modem
TM = Telefix Modem
RM = Relay Modem
descr. = description

* = The terminal line(s) from which the Telefix program is activated

Fig. 4. Telefix local center follow-up

The support-center operator needs to use the DESCRIPTIONs and SCRIPTs present in the local center. For that purpose, he/she uses his/her own TELEFIX command REMOTE-MODE to log in to the local center through the relay-modem line, and start the TELEFIX program in remote mode. The CONNECT-LINE command is given with the line number corresponding to the Telefix modem still connected to the remote site. The local-center operator terminal is specified as the follow-up terminal to allow the viewing of the operations. The remote system runs against the local Telefix, which is activated from the support-center Telefix.



Key

M = Modem
TM = Telefix Modem
RM = Relay Modem
descr. = description

* = The terminal line(s) from which the Telefix program is activated

Fig. 5. TELEFIX-LOCAL - run at customer site

The ND-500 remote system is installed with the TELEFIX-LOCAL program. Automatic jobs for the ND-500 may be started by using the local ND-100 machine, either from a local operator terminal or from a Telefix center starting the TELEFIX-LOCAL with the REMOTE-MODE command. In this last case, the operator terminal on the remote site may be used as a follow-up terminal.

## 3.2 TELEFIX CENTER INSTALLATION

### 3.2.1 HARDWARE INSTALLATION FOR TELEFIX

The following are required:

- an ND-100 computer with asynchronous line interfaces (ND 102710 or ND 102720) switched to V24 mode.

- asynchronous modems, corresponding to the required transfer speed, and connected to the public network and the ND-100 interfaces.

- terminals from which to run the system. Any VDU terminal is suitable, but it is better to use the ones handling video attribute modes if possible. In any case, a terminal must have a working speed greater than the corresponding modem transfer speed.

The necessary number of modems and terminals is dèfined according to the expected work load. One Telefix work station uses one terminal and one modem. In addition, an extra modem should be available as a relay modem to another center.

### 3.2.2 HARDWARE INSTALLATION FOR TELEFIX-LOCAL

No other hardware than the operator terminal(s) is needed, as there is no communication running outside the Telefix system. The operator terminal may be a paper-printer terminal.

### 3.2.3 SOFTWARE INSTALLATION FOR TELEFIX AND TELEFIX-LOCAL

> Please refer to the Telefix Program Description Sheets for information concerning software installation for TELEFIX and TELEFIX-LOCAL.

### 3.2.4 INSTALLATION OF TELEFIX ACCESS DEVICES IN SINTRAN

> Please refer to the Telefix Program Description Sheets for information concerning installation of Telefix access devices in SINTRAN.

## 3.3 REMOTE SITE INSTALLATION

### 3.3.1 HARDWARE INSTALLATION

Make the physical connections as follows:

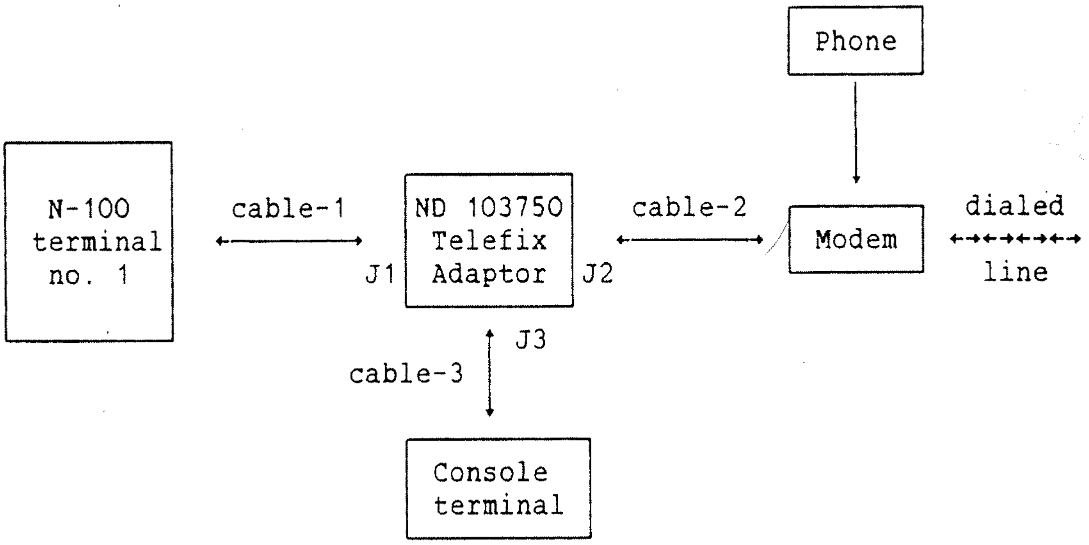


Fig. 6. Installation of the ND 103750 Telefix Adaptor

For more information on installation of the ND 103750 Telefix Adaptor, please refer to the Service Handbook, volume II, page II-1-51-1.

Use of the CONTROL-MODE command demands special hardware and connections which are not described here.

The connection may be established through any asynchronous terminal line, but the command PLACE-BINARY works only if the line is the terminal number 1 (system console). So, it is recommended that terminal line 1 is the one used for all remote systems, which in addition gives the possibility of using OPCOM commands.

### 3.3.2 SOFTWARE INSTALLATION

> Please refer to the Telefix Program Description Sheets for information concerning remote-site software installation

## 3.4 SOFTWARE PROTECTION

The software key system is now used for protection of the Telefix software. Both versions (TELEFIX and TELEFIX-LOCAL) are protected, with the following exception:

The TELEFIX-LOCAL version is <u>not</u> protected when running on a ND-500 system, except for the commands:

```
CREATE-SCRIPT
REGENERATE-SCRIPT
SCRIPT-EDITOR
DESCRIPTION-EDITOR
PRINT-SCRIPT
PRINT-DESCRIPTION
```

# CHAPTER 4

## TELEFIX START-UP

# 4 TELEFIX START-UP

The two Telefix versions (TELEFIX and TELEFIX-LOCAL) may be started in either local mode or remote mode. Local mode is the normal one when the Telefix program is recovered from a terminal (the operator terminal) directly connected to the Telefix system. Remote mode is entered when started from another Telefix system (see the REMOTE-MODE command).

Telefix may also be started as a @MODE or @BATCH job, if the output goes to a VDU terminal known to the Virtual Terminal Manager (VTM). In such a case, it always starts in local mode, and the ESCAPE key may be used to abort the @MODE job. The commands CREATE-SCRIPT and CREATE-MASTER-SCRIPT are not allowed.

It is now possible (in Release C), when running Telefix as a MODE/BATCH job, to go back into Telefix with the SINTRAN command @CONTINUE, without losing the parameter values specified in the previous run. This is possible only if the last command of the run was not EXIT.

To use the TELEFIX version, the RT program RTTELE must be active. It can be started either from user SYSTEM or RT by typing the SINTRAN command: @RT RTTELE. The message:

```
*** RT-TELEFIX-C READY TO OPERATE ***
```

is printed at the system console.

> Note: The Telefix command processor prompt character has been changed from > to <.

As the program is terminal type oriented, it will start by asking you to enter the terminal type if it is not able to recognize the currently defined type. It is mandatory to use a VDU known by the VTM. If possible, use one which can handle video attribute modes, so as to benefit from the better display forms. TELEFIX-LOCAL may also use terminal type 2.

The program TELEFIX may be recovered from any terminal running through any user. Error messages will be generated if the internal device needed by TELEFIX is not accessible, or if it cannot be reserved in the correct way.

# CHAPTER 5

# THE TELEFIX COMMANDS

# 5 THE TELEFIX COMMANDS

The following list shows the syntax of all Telefix commands, and indicates which Telefix version(s) will accept each command (T for TELEFIX and L for TELEFIX-LOCAL, and l for TELEFIX-LOCAL not using terminal type 2):

| | |
|---|---|
| HELP <command> | (T,L) |
| @<SINTRAN command> | (T,L) |
| EXIT | (T,L) |
| ACCOUNTING-ON-OFF | (T) |
| PRINT-ACCOUNTING <output file name> | (T) |
| PRINT-LINES-TABLE <output file name> | (T,L) |
| CONNECT-LINE <device number/Telefix access device> | (T,L) |
| TEST-MODEM | (T,L) |
| SET-TRANSFER-SPEED <speed> | (T) |
| STATUS | (T,L) |
| DISCONNECT-LINE | (T,L) |
| OPEN-LOG-FILE <log file name> | (T,L) |
| APPEND-LOG-FILE <log file name> | (T,L) |
| CLOSE-LOG-FILE | (T,L) |
| LOG-ANALYZER | (T,l) |
| INITIALIZE-TERMINAL <terminal type> | (T,l) |
| DEFINE-SINTRAN-PROMPT | (T) |
| CHANGE-EXIT-CHARACTER | (T,L) |
| FOLLOW-UP-TERMINAL <device number> | (T,L) |
| MESSAGE | (T,L) |
| RELAY-MODE <device number> | (T) |
| REMOTE-MODE <log-in user parameters> <Telefix name> | (T) |
| CONTROL-MODE <password> <control-table file name> | (T) |
| MANUAL-MODE | (T,L) |
| CONVERT-OCTAL-TO-ASCII-MANUAL-MODE | (T,L) |
| SEND-FILE <destination> <source> <from page> | (T) |
| GET-FILE <destination> <source> <from page> | (T) |
| PLACE-BINARY <BPUN file name> | (T) |
| LOAD-PROGRAM <file name> | (T) |
| CREATE-SCRIPT <script file name> | (T,L) |
| CREATE-MASTER-SCRIPT <script file name> | (T,L) |
| REGENERATE-SCRIPT <dest. script file> <source script> <ch.wait> | (T,L) |
| REJECT <characters string> | (T,L) |
| SCRIPT-EDITOR <script name> | (T,l) |
| DESCRIPTION-EDITOR <description name> | (T,l) |
| PRINT-SCRIPT <script name> <output file name> | (T,L) |
| PRINT-DESCRIPTION <description name> <output file name> | (T,L) |
| LIST-SCRIPT-FILES <script file name> <output file name> | (T,L) |
| LIST-DESCRIPTION-FILES <description file name> <output file> | (T,L) |
| LIST-SCRIPT-LIST <script file name> <output file name> | (T,L) |
| LIST-DESCRIPTION-LIST <description file name> <output file name> | (T,L) |
| MUTING-ON-OFF | (T,L) |
| INVERSE-DEFAULT-PARAMETERS | (T,L) |
| DEBUG-MODE-ON-OFF | (T,L) |
| RUN-SCRIPT <script name> <character wait time> | (T,L) |
| AUTOMATIC-MODE <description name> | (T,L) |

> Note: The DUMP-INPUT-BUFFER command is not supported in Release C of Telefix. The TRACE command has been replaced by a related command in the debug mode.

The command name and the parameter(s) are separated by one space or one comma. The included command processor allows you to edit inputs by using certain control codes. If parameters are missing from a command name, they are asked for automatically.

It is not possible to break the TELEFIX program with the ESCAPE key, but the execution of some commands may be interrupted with ESCAPE.

Line editing

The following control codes may be used for line editing (they are the same as when using the DESCRIPTION/SCRIPT editor):

| | |
|---|---|
| HOME | Cancel command. |
| CURSOR LEFT | Move to previous character. |
| CURSOR RIGHT | Move to next character. |
| CTRL+F char. | Move forward to specified character.<br>char.= CTRL+F move beyond last character. |
| CTRL+R char. | Move backwards to specified character.<br>char.= CTRL+R move to beginning of line. |
| CTRL+A or DELETE | Delete one character. |
| CTRL+D char. | Delete up to and including cursor position.<br>char.= CTRL+D delete the entire line.<br>char.= CR delete the rest of the line.<br>char.= CTRL+R delete from beginning of the line. |
| CTRL+K or CTRL+Q | Delete the whole line. |
| CTRL+E | Set/reset expand mode. |
| CTRL+O | Next is a control code (except CR; may be DEL). |
| CR | Terminate the entry. |

In addition, if CR is given directly in answer to the prompt character, the last command given to the command processor is displayed and ready for editing.

TELEFIX-LOCAL used with terminal type 2 does not use the Telefix command processor for input, but rather the SINTRAN command buffer, so the control codes to be used are the ones from SINTRAN.

The Telefix commands are described in the following sections.

## 5.1 HELP

Format: HELP

This command displays the list of all available commands, and their syntax.

## 5.2 EXIT

Format: EXIT

Telefix is finished and the terminal returns to the SINTRAN command processor. The line is disconnected and the log file is closed, if it was not done previously.

## 5.3 ACCOUNTING-ON-OFF

Format: ACCOUNTING-ON-OFF

The internal Telefix accounting system is turned ON or OFF.

If the accounting is ON:

Any time a line is connected, an identification (32 characters length maximum) is requested. When the line is disconnected, the following information is appended to the file (SYSTEM)TELEFIX-ACCOUNT:DATA :

The terminal number running Telefix.
The user name running Telefix.
The communication line number used.
The date and time of the connect-line command.
The date and time of the disconnect-line command.
The identification entered by the operator.

If the accounting is OFF:

The accounting system is not used, and no identification is requested for the connect-line command. If it is turned OFF between the connect-line and the disconnect-line commands, no accounting information is stored. If it was OFF at the connect-line command time and turned ON before the disconnect-line command, incomplete information is stored.

## 5.4 PRINT-ACCOUNTING

Format: PRINT-ACCOUNTING <output file name>

The contents of the accounting file are printed on the specified output file. After the printout is done, it is possible to reset the accounting file.

## 5.5 PRINT-LINES-TABLE

Format: PRINT-LINES-TABLE <output file name>

A table giving the correspondence between terminal numbers, SINTRAN logical device numbers, and hardware IOX device numbers is printed to the file corresponding to <output file name>. The default printout device is the operator terminal.

For TELEFIX-LOCAL:

The table indicates only the logical unit numbers corresponding to the SINTRAN Telefix access devices.

## 5.6 CONNECT-LINE

Format: CONNECT-LINE <device number/Telefix access device>

Unless this command is executed, it is not possible to communicate with any remote system. The parameter <device number> is the SINTRAN logical unit number corresponding to the asynchronous line to be used for communication. The default radix is decimal value.

> Note: The modem test function which was previously (Release B) part of the CONNECT-LINE command, is now found in a separate command called TEST-MODEM (see the next page).

The program checks that the carrier signal from the modem is present at the asynchronous interface input. If it is not, a message indicates that the program will wait up to 5 minutes for the missing signal in order to give you the necessary time to correct the physical connection. You may break this wait by typing ESCAPE.

When the carrier is present, the DTR signal from the interface is locked by transferring a dummy character (CR) in test mode.

The corresponding SINTRAN input buffer is not cleared by this command.

If the accounting has been started, the operator is requested to enter an identification string, maximum 32 characters long.

For TELEFIX-LOCAL:

If the Telefix access device number to be connected is omitted, the first free one is allocated. There is no accounting system, so there is no identification to provide.

## 5.7 TEST-MODEM

Format: TEST-MODEM

This is the modem test function previously included in the CONNECT-LINE command, when running TELEFIX.

This test assumes that all characters sent to the communication line are returned. This may be done by:

- starting a program running at the remote system side which does a simple echo of all received characters.

- setting the remote system modem in test mode, if this possibility exists on this modem type.

Information is displayed at the operator terminal during the test run. The SINTRAN input buffer for the communication line is cleared when performing the modem test.

## 5.8 SET-TRANSFER-SPEED

Format: SET-TRANSFER-SPEED <speed>

The connected line is programmed in the hardware to transfer at the speed specified in the parameter <speed> (baud rate). If the given value is illegal, the possible legal values are printed at the operator terminal.

Remember that the communication-line transfer speed must be lower than that of the operator terminal, to prevent the risk of losing characters when displaying the line traffic.

## 5.9 STATUS

Format: STATUS

The following information is displayed at the operator terminal:

The running mode: local or remote.
The device number for the connected communication line.
The modem status: carrier signal present or not.
The value for the exit character.
The value for the remote system SINTRAN prompt.
The file name for the log file in use, with the number of records.
The accounting status (ON or OFF).
The follow-up terminal number.
The muting for automatic and relay mode status (ON or OFF).
The default parameters for automatic mode (normal or inverse).
The number of characters currently in the input buffer.
The transfer speed, if the SET-TRANSFER-SPEED command was used.
The number of retries done during the last file-transfer command execution.

## 5.10 DISCONNECT-LINE

Format: DISCONNECT-LINE

The asynchronous line previously specified in the CONNECT-LINE command is not known any more by the program, and a hardware clear is issued in the interface (TELEFIX only).

The corresponding SINTRAN input buffer is not cleared by this command.

If the accounting is started, the accounting information is appended to the accounting file.

## 5.11 OPEN-LOG-FILE

Format: OPEN-LOG-FILE <log file name>

During the execution of certain commands, the line traffic may be recorded onto a log file. This command opens the file corresponding to the parameter <log file name>, to make it available for writing. Further, any time the MANUAL-MODE, the RELAY-MODE, the REMOTE-MODE, or the AUTOMATIC-MODE is entered, the line traffic is logged on to this file, in the same way as it appears at the operator terminal. A title, including time, date and the related communication line number, is also recorded when entering or leaving communication with the remote system.

When writing messages to the log file, Telefix issues "log-file format" printouts which are recognized by the LOG-ANALYZER program designed to inspect log files. The log file format identification used by Telefix is TFX.

The default file type for the parameter <file name> is :LOGS. The previous contents of the log file are lost.

## *5.12 APPEND-LOG-FILE*

Format: APPEND-LOG-FILE <log file name>

This command is the same as the OPEN-LOG-FILE command, but the new data is appended to the existing data. The previous contents of the log file are not lost.

## *5.13 CLOSE-LOG-FILE*

Format: CLOSE-LOG-FILE

The log file previously opened with the command OPEN-LOG-FILE or APPEND-LOG-FILE is closed. A message tells how many sessions have been recorded during the time the log file was opened.

Further, it is possible to use the LOG-ANALYZER program which gives the possibility of extracting any desired information.

## *5.14 LOG-ANALYZER*

Format: LOG-ANALYZER

The command EXIT is executed and the LOG-ANALYZER program started. If a log file was in use, the LOG-ANALYZER recovers it automatically and gives a short log status.

## *5.15 INITIALIZE-TERMINAL*

Format: INITIALIZE-TERMINAL <terminal type>

The operator terminal is initialized as the specified type. If <terminal type> is missing, the terminal is re-initialized with the type unchanged. This may be useful after a manual mode sequence, where a program in the remote system has also used the VTM and reset the terminal options.

## 5.16 DEFINE-SINTRAN-PROMPT

Format: DEFINE-SINTRAN-PROMPT

To execute the transfer file commands (see pages 36-39) or the REMOTE-MODE command, Telefix has to log in to the remote system. To know if the log-in is successful, the prompt character coming from the SINTRAN command processor is looked at. It may happen that the prompt string is redefined with a string where the last character is not the usual @. In such a case, the log-in procedure fails unless this new prompt character is specified by this command.

## 5.17 CHANGE-EXIT-CHARACTER

Format: CHANGE-EXIT-CHARACTER

When online with the remote system through the manual mode, any character typed at the operator keyboard is transmitted via the communication line, except one which is known as the exit character and is used to return to the Telefix command processor. This exit character is preset to the value 0 (CTRL+@). This command allows you to change to a new character.

## 5.18 FOLLOW-UP-TERMINAL

Format: FOLLOW-UP-TERMINAL <device number>

The terminal corresponding to <device number> is used as a follow-up display for all future commands. If <device number> is 0 or omitted, the follow-up function is reset. The terminal used may also be a modem, but TELEFIX only checks the hardware status for the carrier signal.

The executed Telefix commands and control-mode commands are displayed at the follow-up terminal, as well as the line traffic during manual and automatic modes, including the automatic-mode Telefix messages.

The follow-up terminal may be used for dialog with the operator terminal (see MESSAGE command). When the person sitting at the follow-up terminal wants to begin communication with the one at the opêrator terminal, he/she has to hit ESCAPE on his/her keyboard. A message indicating that there is a request from the follow-up terminal is delivered to the operator terminal as soon as this one is in command mode.

## 5.19 MESSAGE

Format: MESSAGE

This command allows you to send a message to the follow-up terminal and to get an answer back if wanted, or to receive a message from the follow-up terminal after its ESCAPE key was activated.

## 5.20 RELAY-MODE

Format: RELAY-MODE <device number>

The Telefix center is used as a relay to connect the remote system to another Telefix system. The connected line is directly forwarded to a second line, the relay line, corresponding to <device number>. This is a two-way connection. During relay mode, and if the muting function is not ON, the line traffic from the remote system to the relay line is also displayed at the operator terminal. The relay mode is terminated when the ESCAPE key is activated.

## 5.21 REMOTE-MODE

Format: REMOTE-MODE <log-in user parameters> <Telefix name>

An automatic log-in is performed in the remote system following the <log-in user parameters> (same format as for the transfer file commands). The Telefix program in the remote system (Default= TELEFIX-LOCAL) is started in remote mode, and the first calling Telefix enters a manual mode. The commands in the remotely started Telefix are directly available. This special manual mode in the first system (called remote mode) allows you to send the exit character to the second one in order to be able to exit from a manual mode in this remote Telefix. The remote mode in the calling Telefix is aborted with two consecutive exit characters. See CHANGE-EXIT-CHARACTER and MANUAL-MODE for explanations about the exit character. Any Telefix can be started either in local mode or in remote mode. It is started in remote mode by another Telefix with this REMOTE-MODE command, and is initialized to work with the same terminal type and same exit character as in the calling Telefix.

## 5.22 CONTROL-MODE

Format: CONTROL-MODE <password> <control-table file name>

This command is implemented for use with a hardware control module on the remote system side. It is available only in the TELEFIX version directly connected to the hardware control module, and not passing through relays. The control commands themselves are not implemented in Telefix, but in the control module, and the CONTROL-MODE command is only in charge to put the operator terminal in contact with this module. A control-table file (default: TELEFIX-CNTTB:DATA) must be available at the Telefix system. It includes all the necessary information for Telefix to interface with the control commands implemented in the control module. The descriptions of the available control commands can be found in the documentation concerning the hardware control module. A HELP command and an EXIT command are always implemented. The EXIT command returns to the Telefix command processor.

## 5.23 MANUAL-MODE

Format: MANUAL-MODE

The operator terminal is directly connected to the communication line, and so to the remote system. The line traffic displayed at the operator terminal is also recorded on the log file, if it was previously opened. The manual mode finishes when the operator types the exit character. If characters have already been received in the input buffer when entering the manual mode, a message is displayed and asks whether or not you want to clear those characters.

## 5.24 CONVERT-OCTAL-TO-ASCII-MANUAL-MODE

Format: CONVERT-OCTAL-TO-ASCII-MANUAL-MODE

This is a special way of running MANUAL-MODE. Every time a string of characters which looks like it is a six digit octal number is received from the remote system, it is replaced by the two ASCII characters corresponding to this number. Control codes which are not CR (carriage return) or LF (line feed) are displayed as a period.

An example of where this mode might be used is to dump an area of the remote-system memory containing text strings, using OPCOM commands.

## 5.25 SEND-FILE

Format: SEND-FILE <destination> <source> <from page>

The file <source> in the Telefix system is transferred to the remote-system file corresponding to <destination>.

The transfer begins with the page given in the <from page> parameter. The default value is page number 0, i.e. the beginning of the file.

The receiver program in the remote system will be used, and so must be available. Any file type of any length may be transferred, and the transfer progress is continuously displayed at the operator terminal. The transfer may be aborted with ESCAPE.

When the SEND-FILE command is aborted, the page number of the last correctly transferred page is written to the operator terminal, in order that the transfer may be restarted at this point later on.

The parameter <destination> has the following format:

Log-in user parameters.<Destination file>

The <destination file>, as well as the <source>, follows the standard rules, i.e.:

(Directory:User)File name:Type;version

Telefix has to log in to the remote system in order to start the receiver program and to perform the transfer. The log-in user parameters specify which user will be entered. If it is not specified, the user name appearing in the <destination file> will be entered. If no user at all is specified, FLOPPY-USER will be entered as default. What is happening during the log-in procedure is displayed at the operator terminal in order to show that there are no problems, for example, no transmission error during the user name/password/project password and recover command transfers. Remember to specify the SINTRAN prompt character with the command DEFINE-SINTRAN-PROMPT if the remote-system prompt character is not the usual @. The log-in user parameters have to be specified as follows:

User(password:project password)

Examples of SEND-FILE parameters:

- SYSTEM(XYZ:PROJ)."(SYSTEM-PACK:RT)TEST:PROG",(ND:F-U)MOVER:PROG,,

The file MOVER:PROG, owned by user F-U on directory ND, is transferred in the remote system to user RT on directory SYSTEM-PACK, to a new file called TEST:PROG. To perform the operation, the user SYSTEM will be logged in at the remote system, with XYZ as the password, and PROJ as the project password if the accounting is active.

- PACK-TWO:RT(:PROJ).TEST:PROG,(ND:F-U)MOVER:PROG,,

The file is transferred to the existing file TEST:PROG. As user RT on main directory PACK-TWO will be entered in remote system (without a password, but with PROJ as a project password if needed), the file is expected to be owned by RT (or SYSTEM).

- (RT)TEST:PROG,(ND:F-U)MOVER:PROG,,

The file is transferred to the existing file TEST:PROG owned by user RT. As there is no log-in user specified, the user RT will be entered without a password or project password in any case.

- "TEST:PROG",(ND:F-U)MOVER:PROG,,

The file is transferred to a new file named TEST-PROG. As no user at all is specified, user FLOPPY-USER will be entered as the owner for this new file.

The default file type for source and destination file names is :SYMB.

The communication protocol handles transmission errors by doing retries. Any time an error is detected, the packet size is reduced to about half of its previous value. It will be increased again only if five consecutive packets have been transmitted without error and the total number of retransmissions has not exceeded a value which is also dependent on the last packet size used. So, the packet size is automatically adjusted according to the communication line quality. The program allows a maximum of four retries for each packet. When transmission errors are detected, the operator terminal bell is activated. It is, of course, necessary that both sides work with the same configuration for the number of data/stop/parity bits to avoid having characters rejected by the SINTRAN terminal drivers. The receiver program working in the remote system is normally aborted in any case of command termination. But, if this program hangs up, it is possible to force it to stop by sending three consecutive ESCAPEs on the communication line by using the manual mode.

During the file transfer, the SINTRAN command @TERMINAL-STATUS at the remote system gives the name of the file being received and written.

## *5.26 GET-FILE*

Format: GET-FILE <destination> <source> <from page>

The file corresponding to <source> in the remote system is transferred to the specified <destination> file in the Telefix system.

The transfer begins with the page given in the <from page> parameter. The default value is page number 0, i.e. the beginning of the file.

The transmitter program in the remote system will be used, and so must be available. Any file type of any length may be transferred, and the transfer progress is continuously displayed at the operator terminal. The transfer may be aborted with ESCAPE.

When the GET-FILE command is aborted, the page number of the last correctly transferred page is written to the operator terminal, in order that the transfer may be restarted at this point later on.

The parameter <source> has the same format as the parameter <destination> for the SEND-FILE command:

Log-in user parameters.<source file>

See SEND-FILE command for the log-in mechanism description.

An example of GET-FILE parameters:

- "TEMP",F-U(:TEST).(SYSTEM)PATCH:OUT,,

The file PATCH:OUT owned by user SYSTEM in the remote system is transferred to the Telefix system to a new file named TEMP:SYMB owned by the user currently running Telefix. In the remote system, to activate the transmitter program, the user F-U is entered with no password and with TEST as the project password if required.

The use of the ND 103750 Telefix Adaptor makes a permanent connection between the output channel of the system console interface and the printer for the console terminal, which is usually a paper-printer terminal. During the GET-FILE command execution, all characters sent to the line (file content plus several control codes) are also received by the console printer, and it is advisable to turn the console off-line during the transfer to prevent long printouts of rubbish.

During the file transfer, the SINTRAN command @TERMINAL-STATUS at the remote system gives the name of the file being read and sent.

## *5.27 PLACE-BINARY*

Format: PLACE-BINARY <BPUN file name>

This command is used to load a stand-alone program into the remote system in stop mode. The transfer may be aborted with ESCAPE. It works only if the connection goes into the remote system through the terminal 1 interface. The parameter <file name> must specify a binary file (type :BPUN). The remote machine is expected to be in stop mode, and the command executes the following operations:

A small program taken from the file TELEFIX-A00-BOOT:BPUN is loaded with OPCOM to remote system memory address 0, and started. This program waits for characters on the console line to be loaded from memory address 177500. A receiver program taken from the file TELEFIX-A00-BPUN:BPUN is loaded through the bootstrap program. This is the program which handles the communication procedure. It will start at the end of the load only if no transmission error occurred (self checksum and hardware status verification), and will establish the communication with Telefix. Then, the program corresponding to <file name> is sent with the same protocol characteristics as for the SEND-FILE or GET-FILE command. The communication is assumed to work with seven data bits, one even parity bit, two stop bits (which is the default configuration value for the SINTRAN terminal driver), and no XON/XOFF control. The receiver program working stand-alone in the remote system may be interrupted by sending three consecutive ESCAPEs on the communication line.

After the binary file transfer and load is finished, the remote system stops and the PLACE-BINARY command is terminated. Then, it is possible to enter the MANUAL-MODE to start the placed program. The remote CPU B register was loaded with the program start address, and the P register points to the instruction JMP ,B. A simple continue command (!) starts the program execution.

## 5.28 LOAD-PROGRAM

Format: LOAD-PROGRAM <file name>

This command is used to send a program to the TPE-MONITOR. The specified file (default type :TEST) is sent on the line using the same communications protocol as the commands SEND-FILE, GET-FILE, and PLACE-BINARY. The TPE-MONITOR should already be active in the remote system, and waiting for a command.

## 5.29 CREATE-SCRIPT

Format: CREATE-SCRIPT <script file name>

The manual mode is entered with a clear for the communication-line input buffer. The line traffic is analyzed and recorded to the specified SCRIPT, together with commands and parameter default values. The SCRIPT creation finishes when the operator types the exit character.

It is recommended that this command be used only on a system where the load is small enough to not interrupt the Telefix program very often or for long periods, due to the operating system time sharing. It is also necessary to not type ahead too much from the operator terminal keyboard, in order to get the correct sequences concerning command echo and answers.

The CREATE-SCRIPT command starts the SCRIPT generation with several lines corresponding to the default values of several parameters:

```
TYPING   : The typing speed is set to 1 character ahead.
ECHO ERR : On echo error  : CONTINUE.
ANSW ERR : On answer error: CONTINUE.
```

These three parameters are not written again. Two others are recorded when needed:

```
MAX WAIT : Initial value = 5 seconds. Recorded when changed since
           last time, with 5 seconds as the minimum value.
DELAY    : Only recorded when equal to or greater than 5 seconds.
```

### *5.30 CREATE-MASTER-SCRIPT*

Format: CREATE-MASTER-SCRIPT <script file name>

Same as the CREATE-SCRIPT command, but the answers are not recorded to the SCRIPT. Each answer sequence is replaced by a command line ANSWER NO CARE. It is possible to execute such a "master SCRIPT", and it will not produce any answer error.

### *5.31 REGENERATE-SCRIPT*

Format: REGENERATE-SCRIPT <destination script file name>
<source script name> <character wait time>

This is the same as the CREATE-SCRIPT command, but the COMMANDs are not taken from the operator keyboard, but from the source SCRIPT. DELAYs which may appear in the source SCRIPT are taken into consideration and forwarded to the destination SCRIPT. The ANSWERs and MAX WAITs from the source are used to recognize the supposed timings for new ANSWERs.

The parameter <character wait time> is used to simulate the CH. WAIT value when running a SCRIPT, and is used as a time-out to find the end of the received answers compared with the ones specified in the source script. The default value is 3 seconds.

The source SCRIPT is normally a "master SCRIPT". If it is a normal script, all DO, POS SEQU and IF sequences are skipped. Some commands have no effect and are only copied to the destination.

This is a list of all SCRIPT codes with the corresponding action during regeneration:

Action on SCRIPT codes during regeneration

COMMAND: Used to take characters to be sent to the remote system. Before sending the first character, ANSWERs may be received up to the CH. WAIT time-out.

ECHO: Skipped.

ANSWER: Used to check if at this time one answer may arrive within the MAX WAIT time. If the subcode is NO CARE: all answers up to the CH. WAIT time-out are accepted.

ALT ANSW: Skipped.

POS ANSW: Same as for ANSWER. A wait up to the MAX WAIT time is performed to receive ANSWERs for each list of possible answers found.

POS SEQU: Skipped up to END POS.

END POS: See POS SEQU.

MAX WAIT: The MAX WAIT time is adjusted for the next expected answers. It is not copied to the destination but regenerated with the new value corresponding to the new ANSWER, when different from the previous value.

DELAY: Stops to send COMMANDs during the specified time. Copied to the destination.

CLEAR: Clears the input line and forces the next COMMAND to be sent immediately.

OP. WAIT: Only copied to destination.

OP. BELL: Only copied to destination.

PROG. EX: Only copied to destination.

SET FLAG: Only copied to destination.

RESET FL: Only copied to destination.

DO: Skipped up to the END DO.

END DO: See DO.

CONTINUE: If placed as the first command in the SCRIPT: no clear for the input line from remote system when starting the regeneration. Copied to destination.

TYPING: Typing speed for all next COMMANDs. Copied to destination.

ECHO ERR: Only copied to destination. If subcode is DO or DO + REP: not copied and skipped up to the END DO.

ANSW ERR: Only copied to destination. If subcode is DO or DO + REP: not copied and skipped up to the END DO.

PLACE-BI: Only copied to destination.

REJECT: Specified characters are used for the reject function on ANSWERs, up to the end of SCRIPT. Copied to destination.

ERR MESS: Only copied to destination.

IF TRUE: Skipped up to corresponding ENDIF.

IF FALSE: Skipped up to corresponding ENDIF.

IF ANSW: Skipped up to corresponding ENDIF.

| | |
|---|---|
| IF CONST: | Skipped up to corresponding ENDIF. |
| ELSE: | Skipped. |
| ENDIF: | Skipped. |
| COMMENT: | Only copied to destination. |
| STORE: | Only copied to destination. |
| PROTOCOL: | Only copied to destination. |
| LENGTH: | Only copied to destination. |
| SHIFT: | Only copied to destination. |
| LOG ADD: | Only copied to destination. |
| LOG SUB: | Only copied to destination. |
| LOG AND: | Only copied to destination. |
| LOG OR: | Only copied to destination. |
| LOG XOR: | Only copied to destination. |
| DEC ADD: | Only copied to destination. |
| DEC SUB: | Only copied to destination. |
| OCT ADD: | Only copied to destination. |
| OCT SUB: | Only copied to destination. |
| COPY: | Only copied to destination. |
| LABEL: | Only copied to destination. |
| LOAD-PRO: | Only copied to destination. |
| WAIT FOR: | Only copied to destination. |
| GO TO: | Skipped. |
| CALL: | Skipped. |
| ROUTINE: | Skipped up to corresponding END ROUT. |
| END ROUT: | See ROUTINE. |

## 5.32 REJECT

Format: REJECT <characters string>

During a SCRIPT creation (CREATE-SCRIPT, CREATE-MASTER-SCRIPT, REGENERATE-SCRIPT), the characters specified in <characters string> coming from the remote system will be rejected. These may be any characters, including control codes, but cannot be the Carriage Return code. They may be placed in any order in <characters string>, and up to 60 characters may be specified. If <characters string> is missing, the reject function is reset.

## 5.33 SCRIPT-EDITOR

Format: SCRIPT-EDITOR <script name>

The default file type for a SCRIPT file is :TSCR. The SCRIPT name may be omitted if no file is to be retrieved.

An interactive editor is entered. A help command is available. It is possible to re-edit any line by using the cursor positioning keys to access the wanted fields. The code and subcode names are shown in inverse video boxes if the terminal handles video attributes. The ASCII strings are displayed normally, but the possible included control codes are shown in inverse video by printing the visible character corresponding to the control code +100B. (For example: CTRL+A is displayed as an inverse video A.) If the terminal does not handle video attributes, the control codes are replaced with & and the JOKER codes with #. The file is written when the store command is given.

## 5.34 DESCRIPTION-EDITOR

Format: DESCRIPTION-EDITOR <description name>

The default file type for a DESCRIPTION file is :TDES. The DESCRIPTION name may be omitted if no file is to be retrieved.

An interactive editor is entered. A help command is available. It is possible to re-edit any line by using the cursor positioning keys to access the wanted fields. The code and subcode names are shown in inverse video boxes if the terminal handles video attributes. The ASCII strings are displayed normally, but the possible included control codes are shown in inverse video by printing the visible character corresponding to the control code +100B. (For example: Control+A is displayed as an inverse video A). If the terminal does not handle video attributes, the control codes are replaced with & and the JOKER codes with #. The file is written when the store command is given.

## 5.35 PRINT-SCRIPT

Format: PRINT-SCRIPT <script name> <output file name>

The specified SCRIPT is printed on the specified output.

If answer or command strings include control codes, the corresponding printable code is shown, and the character & is printed under it on the next line.

If answer strings include JOKER codes, the corresponding code number is shown, and the character # is printed under it at the next line.

## 5.36 PRINT-DESCRIPTION

Format: PRINT-DESCRIPTION <description name> <output file name>

The specified DESCRIPTION is printed on the specified output file.

## 5.37 LIST-SCRIPT-FILES

Format: LIST-SCRIPT-FILES <script file name> <output file name>

The available SCRIPT files which have names matching <script file name> are listed to the specified output file name. The default <script file name> is all files owned by the user running Telefix. The default <output file name> is the operator terminal.

## 5.38 LIST-DESCRIPTION-FILES

```
Format: LIST-DESCRIPTION-FILES <description file name>
                               <output file name>
```

The available DESCRIPTION files where the names match with <description file name> are listed to the specified output file name. Default <description file name> is all files owned by the user running Telefix. Default <output file name> is the operator terminal.

## 5.39 LIST-SCRIPT-LIST

Format: LIST-SCRIPT-LIST <script file name> <output file name>

The full file name corresponding to <script file name> is listed, together with the included area names. The output goes to <output file name> (the default is the operator terminal).

## 5.40 LIST-DESCRIPTION-LIST

```
Format: LIST-DESCRIPTION-LIST <description file name>
                              <output file name>
```

The full file name corresponding to <description file name> is listed, together with the included area names. The output goes to <output file name> (default: operator terminal).

## 5.41 MUTING-ON-OFF

Format: MUTING-ON-OFF

The muting function for automatic mode is turned ON/OFF.

When the muting function is ON, there is no display at the operator terminal screen during the execution of AUTOMATIC-MODE, RUN-SCRIPT, and RELAY-MODE commands. But the log file is still written.

## 5.42 INVERSE-DEFAULT-PARAMETERS

Format: INVERSE-DEFAULT-PARAMETERS

The state for the default values concerning the ANSW ERR, ECHO ERR, ALT ANSW when starting the AUTOMATIC-MODE or the RUN-SCRIPT is changed (normal or inverse). The STATUS command gives the state of the default parameters.

In normal mode: The AUTOMATIC-MODE starts by default with the ANSW ERR enabled, the ECHO ERR enabled, and the ALT ANSW enabled. The RUN-SCRIPT starts with the ANSW ERR disabled, the ECHO ERR disabled, and the ALT ANSW disabled.

In inverse mode: The AUTOMATIC-MODE starts by default with the ANSW ERR disabled, the ECHO ERR disabled, and the ALT ANSW disabled. The RUN-SCRIPT starts with the ANSW ERR enabled, the ECHO ERR enabled, and the ALT ANSW enabled.

## 5.43 DEBUG-MODE-ON-OFF

Format: DEBUG-MODE-ON-OFF

This command turns the debugger program (also called "debug mode") on and off. For an explanation of debug mode, see page 85.

## 5.44 RUN-SCRIPT

Format: RUN-SCRIPT <script name> <character wait time>

This command executes an automatic mode with the specified SCRIPT and without any DESCRIPTION control. The echo and answer errors and the alternative answers are all disabled if the default parameters are in the normal state. If they are in the inverse state, these three conditions are all enabled (see the INVERSE-DEFAULT-PARAMETERS command). So, a run SCRIPT cannot report about echo and answer errors if the default parameters are not inverse, but executes all COMMANDs in the same way as the AUTOMATIC-MODE. The parameter <character wait time> gives the value for CH. WAIT which is defined in the DESCRIPTION when using the AUTOMATIC-MODE command, and which cannot be defined elsewhere. The default value is three seconds.

## 5.45 AUTOMATIC-MODE

Format: AUTOMATIC-MODE <description name>

The DESCRIPTION corresponding to <description name> is opened and checked for validity. The automatic mode starts by executing the first specified SCRIPT. The activity and messages are displayed at the operator terminal, if the muting function is not ON, and recorded to the log file, if opened. The automatic mode may be aborted by ESCAPE.

The directory and user names corresponding to the running DESCRIPTION will be used as default directory and user names for the SCRIPT file(s) and possible later DESCRIPTION files to be opened. This allows running DESCRIPTIONs and SCRIPTs from any directory and user.

The SCRIPT execution may be interrupted by the following conditions.

- The end of the SCRIPT is reached or an EXIT is executed within an END DO or END POS command: control returns to the DESCRIPTION at the line following the RUN-SCRIPT command.

- An error exit is executed due to an ERROR EX subcode within an END DO or END POS command, or due to an echo error, an answer error or unexpected answer, or due to a max time repeat command reached on echo or answer error: control returns to the description to display the error message if one was defined, and to do the specified action, either abort, run a new script, link to another description, or continue the description execution. It is possible to specify inside the same description the error exit condition DISPLAY, together with one of the conditions ABORT, LINK TO, or SCRIPT. These last three mutually exclusive, and only the last one specified is valid. If DISPLAY is the last defined condition, it cancels any other one.

- A programmable exit is issued: control goes to the corresponding entry point in the DESCRIPTION or, if the entry point is not defined, returns at the line following the RUN-SCRIPT command.

- An exit is issued due to an operator break (ESCAPE) or a SINTRAN file system error: the corresponding message is displayed and the automatic mode is aborted.

The automatic mode starts with several default values. The time-out is set to 50%. The character wait time is set to three seconds. The echo error, answer error, and alternative answers are enabled if the default parameters are in the normal state. If they are in the inverse state, these three conditions are all disabled (see the INVERSE-DEFAULT-PARAMETERS command). All constants and flags are reset. When entering a DESCRIPTION, the error exit is preset to ABORT and the error exit message (DISPLAY) is reset. The definitions for time-out, character wait, echo and answer error, alternative answer enable, constants and flags are the ones previously defined. When entering a SCRIPT execution, the MAX WAIT time is preset to five seconds without adjustment according to the TIME OUT, the typing speed to one character ahead, and the CONTINUE condition on echo and answer error. The reported error messages are ON. The terminal bell for OP. WAIT is set to active = 1 second and silent = 0 seconds. The communication line input buffer is cleared if the SCRIPT does not start with a CONTINUE command.

Checks are performed on the command sequences inside a script. After any command string, an echo condition command is expected; if no echo is specified, it is assumed that no echo is returned from the command string. If no answer is specified between two commands, it is expected that no characters will be received from the remote system. Characters other than the echo characters following the first command string will be considered as unexpected answers (if the ANSW ERR condition is enabled).

The character wait time is used when the communication line input buffer is read. It works as a time-out to look at characters arriving from the remote system and to decide if the answers or echoes are terminated, in the following cases (this time-out restarts when a character arrives before the end of time):

- Before sending a new command, if the last answer was wrong or if it is not supposed to receive an answer since the previous command, and if the CLEAR command was not just used. If characters arrive within the character wait time, they are treated as unexpected answer characters, until the time-out is reached. This check is suppressed if the ANSW ERR condition is disabled, and in such a case the possible characters already received in the communication buffer are not read at all and will be treated as an answer later on.

- When reading unexpected answers, to detect when no more of those unexpected answers arrive.

- When reading echo on command string, in case of missing characters.
- When reading the expected answers, to detect the end of answer when it is not terminated with a CR.
- When displaying answers without check, for NO CARE condition, and for NO CHECK condition up to a received CR.

Error messages, telling what was the expected answer, print the character & in place of control codes other than CR or LF, and print the character # in place of JOKERs. The alternative answers are not mentioned in the error messages. If the error concerns an answer not found in a possible list of answers, the error message does not mention the possible answers' contents, but the SCRIPT line where the list starts.

# CHAPTER 6

# THE AUTOMATIC-MODE COMMANDS

# 6 THE AUTOMATIC-MODE COMMANDS

## 6.1 OVERVIEW

The MANUAL-MODE allows the operator to send commands to the remote system, either to run test programs, to work in stop mode, to issue SINTRAN commands or to start user programs. To check what is happening after any command string was sent from the keyboard, the operator has to follow the terminal printout received back on the communication line. Using a log file allows you to go back over the different steps in the investigation afterward, to see exactly what happened.

The AUTOMATIC-MODE sends the command strings to the remote system, taken from disk files called SCRIPTs, and following programmable rules. The answers are displayed together with possible messages produced by an online analysis (line traffic and messages are also recorded to the log file in use). This analysis is performed with additional information from the SCRIPTs. The SCRIPT files may be created by recording the line activity during a manual mode session by using the CREATE-SCRIPT command. A SCRIPT EDITOR allows you to look at the SCRIPTs' contents, to edit the included data or commands and parameters, and to create new SCRIPTs directly. The way the SCRIPTs are executed during the AUTOMATIC-MODE is described in another disk file family called DESCRIPTIONs.

The DESCRIPTIONs hold the necessary information to run SCRIPTs. They have two main parts: 1) the SCRIPT names, in the sequence they have to be executed, and 2) general commands, such as the time-out after delay on expected answer, the action to take in case of error exit from a script execution, etc. The DESCRIPTION EDITOR is used to create or edit the DESCRIPTION files.

The editors will work with any terminal, but ideally you should use one which can handle video attribute modes (terminal type 36, for example).

Each line in a SCRIPT is a command, and includes three fields: the command code, a subcode, and a parameter. The command code is always present, but the subcode and parameter are dependent on the command type and may be nonexistent. The maximum length for the parameter field is 60 characters.

The SCRIPT execution starts with the first command string to be sent to the remote system. It may include any code. The echo from the command string is checked according to the associated parameters. Next comes a parameter indicating the delay time within which an answer is assumed to be received in response to the command string. This is followed by the expected answer, the other alternative possible answer(s), and parameters telling how to check the answer and decide if it is acceptable or not. A list of possible answers may also be specified with sequences to execute in case of a match. A delay may separate this first part from the next one starting again with a command string or with extra expected answers. Other parameters give what actions to perform in the event of a detected error. These parameters may have been changed from those originally specified by the execution of intervening commands.

When the SCRIPT specifies that an answer is expected from the remote system, it must arrive before the end of the currently defined MAX WAIT time. The answer is assumed to be fully received when:

- A CR code is received.
- The expected number of characters is received, if the main expected answer is not supposed to end with a CR.
- 60 characters have been received.

The received answer is checked against the specified expected answer from the SCRIPT and, if no match occurs, it is then checked against the alternative answers if some are present (or against possible answers if such a list is present). The comparison is done character by character on the lower 7 bits, and taking care of the possible JOKER codes (see page 72). For answers terminated with CR, any LFs which follow are skipped, without checking how many of them there are.

When checking for unexpected answers, before sending a new command, and during delays, it is taken into consideration that no more answer characters will arrive when the CH. WAIT time (i.e. the time-out when waiting for the next character) is reached.

If the muting function is not turned on, the automatic mode displays the line traffic at the operator terminal, together with some comments and error messages. A log file may be also used. The messages issued by Telefix follow the "log file format" recognized by the LOG-ANALYZER program. The displays on the operator terminal screen are not in phase with the real times concerning character traffic, as Telefix buffers echoes and answers for analysis before displaying and recording to the log file.

## *6.2 THE MULTIAREA FILE CONCEPT*

The DESCRIPTIONs and SCRIPTs are stored on DESCRIPTION files (type :TDES) and SCRIPT files (type :TSCR). These files use a multiarea file concept in order to save disk space and the number of files for one user. This file organization may be considered as an option, because it is possible to forget about it and to use them as single area files.

A file cannot be empty and always includes the first area referred to as the "default area". The default area may be empty only if there is at least one other nonempty area. The other areas are referred to by an area name with a 16 character maximum length, and standard SINTRAN abbreviation rules.

The syntax to refer to an area is: File name/Area name, which gives, with respect to the SINTRAN file system, the full definition:

(DIRECTORY:USER)FILE NAME:TYPE;VERSION/Area name

If "/Area name" is missing, the default area is assumed. All Telefix commands having read access to a file may use "/Area name", but the ones having write access (CREATE-SCRIPT, CREATE-MASTER-SCRIPT) cannot use it and create only the default area. The editors include the commands which are able to generate the multiarea files, either directly or by merging several single area files. The REGENERATE-SCRIPT command may also generate multiareas.

The multiarea files are sequential files without any space lost between areas. They are protected by internal checksums and can be edited only by the Telefix editors. If one area is corrupted, the whole file is destroyed.

In sections 6.3 to 6.7.19 below, DESCRIPTION file name or SCRIPT file name means a file name without the "/Area name" specification. DESCRIPTION name or SCRIPT name refers to an area name and so to a file name with associated "/Area name" if the area is not the default area.

## *6.3 THE SCRIPT COMMANDS*

Each line in a SCRIPT is a command which includes these three fields:

- command code
- subcode
- parameter

Some commands do not require a subcode and/or a parameter; in such cases the subcode and/or parameter field(s) are left blank. A summary of all of the SCRIPT commands, together with their subcodes and parameters where applicable, is given in the following table. Detailed descriptions of the commands are given in the sections immediately following this table.

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| COMMAND | STRING | Any string without control codes. Ends with CR. |
| COMMAND | ANY | Any characters. May include control codes. |
| ECHO | FULL+ LF | -------- |
| ECHO | FULL | -------- |
| ECHO | NOT LAST | -------- |
| ECHO | NONE | -------- |
| ECHO | NO CHECK | -------- |
| ANSWER | STRING | Any string without control codes. Ends with CR. |
| ANSWER | ANY | Any characters. May include control codes. |

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| ANSWER | NO CHECK | -------- |
| ANSWER | NO CARE | -------- |
| ALT ANSW | STRING | Any string without control codes. Ends with CR. |
| ALT ANSW | ANY | Any characters. May include control codes. |
| POS ANSW | STRING | Any string without control codes. Ends with CR. |
| POS ANSW | ANY | Any characters. May include control codes. |
| POS SEQU | -------- | -------- |
| END POS | RETURN | -------- |
| END POS | CONTINUE | -------- |
| END POS | EXIT | -------- |
| END POS | ERROR EX | -------- |
| MAX WAIT | SECONDS | Decimal value. |
| MAX WAIT | MINUTES | Decimal value. |
| DELAY | SECONDS | Decimal value. |
| DELAY | MINUTES | Decimal value. |
| CLEAR | -------- | -------- |
| OP. WAIT | -------- | Message. |
| OP. BELL | ACTIVE | Number of seconds. |
| OP. BELL | SILENT | Number of seconds. |
| PROG. EX | Entry No | -------- |
| SET FLAG | Flag No | -------- |
| RESET FL | Flag No | -------- |
| DO | -------- | Decimal value. |
| END DO | CONTINUE | -------- |
| END DO | RETURN | -------- |
| END DO | EXIT | -------- |
| END DO | ERROR EX | -------- |
| END DO | GO TO | Any label name. |
| CONTINUE | -------- | -------- |
| TYPING | FULL | -------- |
| TYPING | CHAR/SEC | Number of characters per second. |
| TYPING | AHEAD | Number of characters typed ahead. |
| ECHO ERR | ERR EXIT | Number of errors before exit. |
| ECHO ERR | CONTINUE | -------- |
| ECHO ERR | NEXT CMD | -------- |
| ECHO ERR | REPEAT | Maximum number of repeats. |
| ECHO ERR | DO | -------- |
| ECHO ERR | DO + REP | Maximum number of repeats. |
| ANSW ERR | ERR EXIT | Number of errors before exit. |
| ANSW ERR | CONTINUE | -------- |
| ANSW ERR | NEXT CMD | -------- |
| ANSW ERR | REPEAT | Maximum number of repeats. |
| ANSW ERR | DO | -------- |
| ANSW ERR | DO + REP | Maximum number of repeats. |
| PLACE-BI | -------- | The name of a file containing a binary program. |
| REJECT | -------- | Any characters. May include control codes. |
| ERR MESS | ON | -------- |
| ERR MESS | OFF | -------- |
| IF TRUE | Flag No | -------- |
| IF FALSE | Flag No | -------- |
| IF ANSW | -------- | Any string. May include control codes. |
| IF CONST | Const No | Any string. May include control codes. |
| ELSE | -------- | -------- |
| ENDIF | -------- | -------- |
| COMMENT | LOCAL | The comment string, without control codes. |

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| COMMENT | DISPLAY | The comment string, without control codes. |
| STORE | Const No | -------- |
| PROTOCOL | ON | |
| PROTOCOL | OFF | |
| LENGTH | Const No | -------- |
| SHIFT | Const No | Number of positions. |
| LOG ADD | Const No | Any string of characters. |
| LOG SUB | Const No | Any string of characters. |
| LOG AND | Const No | Any string of characters. |
| LOG OR | Const No | Any string of characters. |
| LOG XOR | Const No | Any string of characters. |
| DEC ADD | Const No | Any string of characters. |
| DEC SUB | Const No | Any string of characters. |
| OCT ADD | Const No | Any string of characters. |
| OCT SUB | Const No | Any string of characters. |
| COPY | Const No | Any string of characters. |
| LABEL | Const No | Any string of printable characters and spaces. |
| GO TO | -------- | Any label name. |
| LOAD-PRO | -------- | File name. |
| CALL | -------- | Any ROUTINE name. |
| ROUTINE | -------- | Any name consisting of printable characters and spaces. |
| END ROUT | -------- | -------- |
| WAIT FOR | ANY | Any characters, including control codes and JOKERs. |
| WAIT FOR | STRING | Any characters, including JOKERs. |

### 6.3.1 COMMAND

The parameter is the command string to be sent on the line. Subcode ANY is for a string including any code, displayable character or control character, except CR. Subcode STRING is for a string which does not include control codes, and which is terminated by the control character CR, which is part of the string. Since the maximum length for one string is 60 characters, the longer strings have to be split into several of the type ANY, except for the last one which may be of the type STRING.

Before sending the command string, an additional check is performed to see that no unexpected answer is present in the communication line input buffer, but only if the ANSW ERR condition is enabled. If it is, and characters are present, this unexpected answer is displayed. An error message is issued if the received string is not a message following the "log file format", sent from the program running into the remote system. In this last case, the message is directly displayed and recorded in the log file.

### 6.3.2 ECHO

No parameter. The subcode tells what echo is expected from the command string. The ECHO command immediately follows the command string, and if no ECHO is specified at the line after the COMMAND, no echo is expected (same as subcode NONE).

FULL+ LF : All characters are echoed and an additional Line Feed is expected.

FULL : All characters are echoed.

NOT LAST : All characters are echoed except the last command string character.

NONE : No echo, the possible returned characters are an answer.

NO CHECK : The echo is received and displayed, but no check is performed. The number of echoed characters should correspond to the command string, including the possible final CR/LFs.

### 6.3.3 ANSWER

The parameter is the answered string. The possible subcodes are ANY, STRING, NO CHECK and NO CARE. The subcode ANY indicates an answer including any character except CR. The subcode STRING indicates an answer without control codes but terminated with at least one CR, with possible extra LFs. In addition, both answer types may include JOKER codes (see page 72).

If subcode NO CHECK is specified, the single subsequent answer is received and displayed without any control, up to CR/LF or the CH. WAIT time-out.

If subcode NO CARE is specified, all the subsequent answers are received and displayed without any control, up to the CH. WAIT time-out.

### 6.3.4 ALT ANSW

ALT ANSW works in the same manner as ANSWER. The possible subcodes are ANY and STRING. It gives one possible alternative answer. There can be an unlimited number of ALT ANSW, but all of them must be placed in sequence after the ANSWER line.

If the received answer from the remote system is not the expected ANSWER, nor one of the ALT ANSWs, an error message is issued. It indicates that the received answer was not the string corresponding to the expected ANSWER, but the alternative possibilities are not mentioned.

If ANSWER and ALT ANSWs of the type ANY are specified, the longest string possibility must be the ANSWER, and not one of the ALT ANSWs, because the answer received from the remote system is truncated according to the number of characters found in the ANSWER command.

### 6.3.5 POS ANSW

The possible subcodes are ANY and STRING. POS ANSW specifies one possible answer which may be returned or not from the remote system. For more details, see "Processing lists of possible answers", on pages 74-75.

### 6.3.6 POS SEQU

Defines the entry point of a sequence to be executed if there is a match with the associated possible answer. For more details, see "Processing lists of possible answers", on pages 74-75.

### 6.3.7 END POS

Defines the end of a sequence executed after a match with a possible answer was detected. The subcode may be: RETURN, CONTINUE, EXIT, ERROR EX. For more details, see "Processing lists of possible answers", on pages 74-75.

### 6.3.8 MAX WAIT

The subcode is either SECONDS or MINUTES, and the parameter is a decimal value. It gives the maximum time the system will wait for the next answer. This command may appear at any place in the SCRIPT, and the system will always use the last defined time when waiting for an answer. Because of the time-sharing system Telefix runs under, it will always wait for a minimum of 5 seconds in any case.

The MAX WAIT time is adjusted according to the TIME OUT value.

### 6.3.9 DELAY

The subcode is either SECONDS or MINUTES, and the parameter is a decimal value. This is the time the system will wait before executing the next script line. It may be used to give delays between commands, for instance. During DELAYs, checks are continuously made for the presence of unexpected answers coming from the communication line, except if the ANSW ERR condition is disabled. If there are any answers, error messages are issued.

### 6.3.10 CLEAR

All received characters from the remote system which have not yet been processed are disregarded, and the next SCRIPT command which is supposed to check for unexpected answers (COMMAND or DELAY) will bypass this check. This command must be used with care. It is useful to force a COMMAND string to be sent in any case.

### 6.3.11 OP. WAIT

The SCRIPT execution is in a waiting state until the operator types a CR on his/her terminal keyboard. If the parameter field is not empty or is not equal to the string "NO BELL", a Telefix message is written to the terminal together with the parameter string. If the parameter field is empty, the wait is entered without any message to the terminal. If the parameter field is equal to the string "NO WAIT", the OP. WAIT command is bypassed. During the waiting time, the terminal bell is activated, following the instructions given by the command OP. BELL (see section 6.3.12). Instead of typing a CR, the operator may hit the ESCAPE key to abort the automatic mode.

Consequently, the OP.WAIT command has four different ways of working, depending on the message in the parameter field.

- No message: Wait without a message to the operator terminal, but with the bell activated.
- Message equal to "NO BELL": Wait without a message to the operator terminal and with the bell not activated.
- Message equal to "NO WAIT": Wait not performed at all.
- Message is something other than "NO BELL" or "NO WAIT": Wait with a message to the operator terminal and the bell activated.

### 6.3.12 OP. BELL

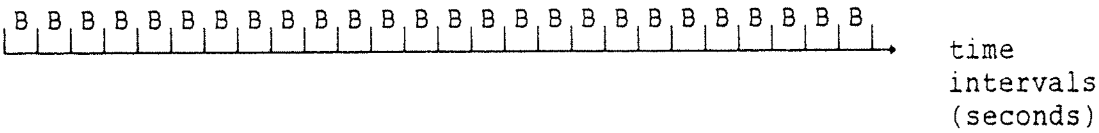
The subcode is ACTIVE or SILENT. The parameter is the time in seconds. This command determines the activity of the operator terminal bell for subsequent OP. WAIT commands.

The bell has alternating ACTIVE and SILENT periods. The duration of each (in seconds) is specified by the parameter in the OP. BELL command with the respective subcode.

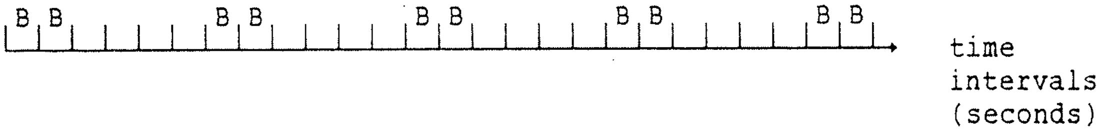
The default values when starting a SCRIPT execution are 1 second ACTIVE and 0 seconds SILENT, i.e. a continuous bell tone.

Diagrammatically, this may be pictured as follows: (B=bell tone)

```
Commands:     OP. BELL  ACTIVE  1   (the default situation)
              OP. BELL  SILENT  0
```



```
Commands:     OP. BELL  ACTIVE  2
              OP. BELL  SILENT  4
```



### 6.3.13 PROG. EX

This is a programmable exit command to stop the SCRIPT execution. The subcode is the DESCRIPTION entry-point number for return. See DESCRIPTION code ENTRY.

### 6.3.14 SET FLAG AND RESET FL

255 flags may be set or reset, and tested afterwards within the SCRIPT or DESCRIPTION. The subcode is the flag number. See the DESCRIPTION and script codes IF TRUE and IF FALSE.

### 6.3.15 DO

Loop command: the parameter value gives the number of times the following sequence up to the END DO line will be executed. If equal to zero, the loop is re-executed forever. Nested DO - END DO sequences are not allowed.

### 6.3.16 END DO

Used to terminate a conditional sequence following a DO loop sequence or a DO subcommand on ECHO or ANSWER error. The action to perform afterwards is given by the subcode.

CONTINUE: The SCRIPT execution continues on the next line following the END DO command, for an error DO sequence. This is the normal way to finish a DO loop sequence, to re-enter the loop, or to leave it if the specified number of loops are executed.

RETURN: The SCRIPT execution returns to the line following the one where the error was detected, or returns to repeat the command in the case of a DO + REP sequence entered on answer or echo error. For a normal DO loop sequence, the loop is re-executed forever, regardless of the specified number of loops.

EXIT: The SCRIPT is aborted, and control returns to the DESCRIPTION. (Not an error exit, but the same exit as for the end of script.)

ERROR EX: The SCRIPT is aborted and an error exit generated.

GO TO: When used to terminate a DO loop, after the loop has been executed the specified number of times, an unconditional jump to the specified LABEL takes place. When used to terminate an ANSW ERR or ECHO ERR handling sequence (with DO or DO + REP subcodes), an unconditional jump to the specified LABEL takes place. There is <u>no</u> repeat of the last command in the case of DO + REP.

The END DO EXIT and the END DO ERROR EX may be used outside any loop sequence to terminate the SCRIPT execution at any point, either with normal or error exit.

### 6.3.17 CONTINUE

This command has no subcode and no parameter. The different uses of the CONTINUE command are as follows:

| Position of CONTINUE: | Effect: |
|---|---|
| First SCRIPT command | No clearing of communication line input buffer |
| Within a DO loop | Forces jump to command after corresponding END DO |
| Within a POS SEQU | Forces jump to command after list of possible answers |
| Anywhere else | No effect |

The first mentioned use of the CONTINUE command is useful when the input buffer should not be cleared, as for example when it is necessary not to lose answers from a previous SCRIPT.

### 6.3.18 TYPING

Subcode= FULL : means the command string is sent at full speed.

Subcode= CHAR/SEC, parameter= decimal value : This command tells the system at what average speed the characters taken from the command strings must be sent on the line, in the number of characters per second.

Subcode= AHEAD, parameter= decimal value : The command string is sent at a controlled speed, so that the number of characters typed ahead is no more than the quantity specified by the parameter value.

Remember that, when you follow the dialog on the operator terminal screen, the characters displayed do not match the real speed.

### 6.3.19 ECHO ERR

The subcode gives the action to take in case the echo received back when sending a command string is not the expected one.

ERR EXIT: The SCRIPT execution is aborted, an error exit is issued. But, it is allowed to have the number of errors specified by the parameter value before issuing the error exit. For example: if the value is 0, it means no error at all is allowed and the first detected error will trigger the exit.

CONTINUE: No action is taken, the SCRIPT execution is continued.

NEXT CMD: The SCRIPT execution skips to the next command string to be sent, after the possible answers not yet processed are displayed.

REPEAT: The same command string is sent again, after the possible answers not yet processed are displayed. The parameter value gives the maximum number of times the command string may be repeated in case of an error before an error exit is issued.

DO: In the case of a detected error, the SCRIPT execution will branch to the next line, and the lines following the ECHO ERR line up to the first END DO command will be executed. When reading this command, a skip is performed to go to the line following the END DO command; but further, the control will return to the line starting the DO - END DO sequence in case of a detected and reported error. After the conditional sequence is executed in case of error, control goes according to the END DO subcode. Specifying another DO sequence inside the conditional sequence is not allowed.

DO + REP: This means do and repeat, and is a combination of the two subcodes DO and REPEAT. The action is the same as DO, but on the END DO command finishing the conditional sequence, and if the END DO subcode RETURN is specified, the execution returns to repeat the failing command. The parameter value specifies how many times it is allowed to repeat the same command, in the same way as for the ECHO ERR REPEAT.

The last definition for the echo error condition stays valid until another definition occurs. Practically, it means a new definition should be given inside a conditional sequence, otherwise the same sequence will be repeated forever in case of a fixed error. It is also possible to finish the conditional sequence with an EXIT or ERROR EX subcode placed in the END DO line, in order to leave the SCRIPT and to return to the DESCRIPTION control.

### *6.3.20 ANSW ERR*

The subcode gives the action to take in case the answer to the command string is not the expected one, nor one of the possible alternative answers, or in case of an unexpected answer. Same subcodes as for ECHO ERR, but in the case of detected unexpected answers, only subcode ERR EXIT works, while the other ones have the same effect as CONTINUE.

### 6.3.21 PLACE-BI

The PLACE-BINARY command is executed. The parameter is the file name for the binary program to be loaded in the remote system's main memory. Any error during the PLACE-BINARY function is considered a fatal error and aborts the automatic mode. See the description of the TELEFIX command PLACE-BINARY (on page 39) for more details concerning this function.

### 6.3.22 REJECT

The parameter is a string including all codes coming from the remote system which must be rejected. They may be displayable or control characters (excluding Carriage Return), specified in any order. The rejected codes are not displayed, but they restart the character wait time-out as the next character is awaited after a rejected one. A REJECT command with an empty parameter field resets the reject function.

### 6.3.23 ERR MESS

The subcode is ON or OFF, with no parameter. The error messages written to the operator terminal and the log file when detecting echo or answer errors may be suppressed or not by this command. It affects only the reported errors. The error detection conditions are not changed by this command. This command may be used if you wish to produce and use an error for programming purposes only, without giving any message. The default value when starting a SCRIPT execution is ON.

### 6.3.24 IF TRUE

This command is used to test the state of a programmable flag set or reset during the SCRIPT execution (see SET FLAG and RESET FL). The subcode is the flag number (1 to 255). There is no parameter.

If the condition is satisfied, the SCRIPT commands following the IF line are executed up to the ENDIF, or the ELSE, if present. If the condition is not satisfied, a skip after the ENDIF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### *6.3.25 IF FALSE*

This command is used to test the state of a programmable flag set or reset during the SCRIPT execution (see SET FLAG and RESET FL). The subcode is the flag number (1 to 255). There is no parameter.

If the condition is satisfied, the SCRIPT commands following the IF line are executed up to the ENDIF or the ELSE, if present. If the condition is not satisfied, a skip after the ENDIF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### *6.3.26 IF ANSW*

This command is used to test the contents of the last received answer. There is no subcode, and the parameter is the comparison string, which may include control codes and JOKERs. The comparison starts on the first character position and goes up to the last parameter string character position. The last received answer is the character string read from the remote system by the last executed ANSWER or POS ANSW command code. If the last executed ANSWER was ANSWER NO CHECK or ANSWER NO CARE and no answer was received, the previous last received answer is still valid for comparison.

If the condition is satisfied, the SCRIPT commands following the IF line are executed up to the ENDIF, or the ELSE, if present. If the condition is not satisfied, a skip after the ENDIF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### *6.3.27 IF CONST*

This command is used to test the contents of a constant (see "CONSTANTS as parameters", page 73). The subcode is the constant number and the parameter is the comparison string, which may include control codes and JOKERs.

If the condition is satisfied, the SCRIPT commands following the IF line are executed up to the ENDIF or the ELSE, if present. If the condition is not satisfied, a skip after the ENDIF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### *6.3.28 ELSE*

This command indicates that SCRIPT execution will continue with the statement immediately following the ELSE, in the event that the test condition for the associated IF TRUE, IF FALSE, IF ANSW, or IF CONST is <u>not</u> satisfied. There is no subcode and no parameter.

### 6.3.29 ENDIF

This command indicates that unconditional execution of the SCRIPT will be resumed with the statement after the ENDIF, following the execution of the test condition, and possibly also of intervening code. There is no subcode and no parameter.

### 6.3.30 COMMENT

The parameter is the comment string terminated by CR. If the subcode is LOCAL, then no action is taken; this is only a local comment in the script. If the subcode is DISPLAY, the comment is written on the operator terminal.

### 6.3.31 STORE

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. There is no parameter.

This command stores the last received answer in the specified CONSTANT.

### 6.3.32 PROTOCOL

The subcode is ON or OFF. There is no parameter.

This command turns ON or OFF a special communication protocol for message exchange between TELEFIX and the TPE-MONITOR, during AUTOMATIC-MODE execution. The default value is OFF when starting an AUTOMATIC-MODE.

### 6.3.33 LENGTH

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. There is no parameter.

The length of the last received answer (the number of characters, excluding the possible terminating CR) is stored in the specified CONSTANT as a two digit decimal number. The length may have a value between 00 and 60.

### 6.3.34 SHIFT

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is a decimal value.

The contents of the specified CONSTANT are shifted <u>left</u> by the number of positions specified by the parameter.

### 6.3.35 LOG ADD (LOGICAL ADDITION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The characters in the parameter field are added to the corresponding characters in the specified CONSTANT. The result is written back to the CONSTANT.

The addition concerns only the seven lower bits of the characters, and there is no carry between characters. The positions where a JOKER appears, either in the CONSTANT or parameter field, are skipped.

### 6.3.36 LOG SUB (LOGICAL SUBTRACTION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The characters in the parameter field are subtracted from the corresponding characters in the specified CONSTANT. The result is written back to the CONSTANT.

The subtraction concerns only the seven lower bits of the characters, and there is no carry between characters. The positions where a JOKER appears, either in the CONSTANT or parameter field, are skipped.

### 6.3.37 LOG AND (LOGICAL AND)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The operation "logical and" is performed between the characters in the parameter field and the corresponding characters in the specified CONSTANT. The result is written back to the CONSTANT.

The "logical and" is performed on only the seven lower bits of the characters. The positions where a JOKER appears, either in the CONSTANT or parameter field, are skipped.

### 6.3.38 LOG OR (LOGICAL INCLUSIVE OR)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The operation "logical inclusive or" is performed between the characters in the parameter field and the corresponding characters in the specified CONSTANT. The result is written back to the CONSTANT.

The "logical inclusive or" is performed on only the seven lower bits of the characters. The positions where a JOKER appears, either in the CONSTANT or parameter field, are skipped.

### 6.3.39 LOG XOR (LOGICAL EXCLUSIVE OR)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The operation "logical exclusive or" is performed between the characters in the parameter field and the corresponding characters in the specified CONSTANT. The result is written back to the CONSTANT.

The "logical exclusive or" is performed on only the seven lower bits of the characters. The positions where a JOKER appears, either in the CONSTANT or parameter field, are skipped.

### 6.3.40 DEC ADD (DECIMAL ADDITION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The ASCII decimal characters (60B to 71B) in the parameter field are added to the corresponding characters in the specified CONSTANT. The result is written to the CONSTANT. The operation is performed only if both characters are ASCII decimal characters.

There is a carry on the decimal addition, as long as the numeric field is unbroken. If it is broken, the decimal addition begins again at the next occurrence of a numeric field, but the carry is lost (i.e. there is no carry between isolated numeric fields, but only within them).

### 6.3.41 DEC SUB (DECIMAL SUBTRACTION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

The ASCII decimal characters (60B to 71B) in the parameter field are subtracted from the corresponding characters in the specified CONSTANT. The result is written to the CONSTANT. The operation is performed only if both characters are ASCII decimal characters.

Borrowing occurs on the decimal subtraction, as long as the numeric field is unbroken. If it is broken, the decimal subtraction begins again at the next occurrence of a numeric field, but there is no borrowing <u>between</u> isolated numeric fields, only within them.

### 6.3.42 OCT ADD (OCTAL ADDITION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

This command operates in the same way as DEC ADD, but with an octal radix.

### 6.3.43 OCT SUB (OCTAL SUBTRACTION)

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is any string of characters.

This command operates in the same way as DEC SUB, but with an octal radix.

### 6.3.44 COPY

The subcode of this command is the reference number of a CONSTANT in the DESCRIPTION governing the SCRIPT. The parameter is a string of characters to be placed in the specified CONSTANT.

This command copies a string of characters into the specified CONSTANT. (A sense of propriety probably keeps the developers from referring to the result of this operation as a <u>variable constant</u> - hence the choice of the word COPY.)

### 6.3.45 LABEL

This command has no subcode. The parameter may be any string of printable characters and spaces.

This command specifies a name to identify the next SCRIPT line, so that this line may be jumped to by a GO TO command.

### 6.3.46 GO TO

There is no subcode. The parameter is any LABEL name.

Upon execution of this command, an unconditional jump is made to the statement pointed to by the LABEL.

### 6.3.47 LOAD-PRO

There is no subcode. The parameter is the file name (with the default type :TEST).

This SCRIPT command executes the Telefix command LOAD-PROGRAM (see page 40). That command sends a program to the TPE-MONITOR. The specified file is sent on the line using the same communications protocol as the commands SEND-FILE, GET-FILE, and PLACE-BINARY. The TPE-MONITOR should already be active in the remote system, and waiting for a command.

### 6.3.48 CALL

There is no subcode. The parameter is any ROUTINE name.

A jump to the named ROUTINE is executed. ROUTINE calls may be nested up to a depth of 100.

### 6.3.49 ROUTINE

There is no subcode. The parameter is any name, which may include any printable characters and spaces.

This command identifies a SCRIPT line as the ROUTINE entry point.

### 6.3.50 END ROUT

There is neither subcode nor parameter.

This command identifies the end of a ROUTINE. The SCRIPT execution returns to the line immediately following the ROUTINE call.

### 6.3.51 WAIT FOR

The subcode may be ANY or STRING. If the subcode is ANY, the parameter consists of a string of any characters, including control codes and JOKERs. If the subcode is STRING, the parameter is a string of any characters, including JOKERs but not control codes.

Telefix waits for an answer matching the one specified in the parameter. All non-matching answers are displayed, as well as the matching one which stops the execution of the WAIT FOR command. The character wait time (CH. WAIT) is used to determine the end of each answer, and the MAX WAIT time is used as a time-out when no matching answer is found.

## 6.4 JOKER CODES

JOKER codes can be included in expected answers, alternative answers, possible answers, and strings for comparison in IF ANSW and IF CONST. They allow you to perform conditional checks. When using the SCRIPT editor, Each JOKER code is displayed on the screen as a special sign. They are replaced with the sign # if the terminal is not able to display the JOKER sign. When using the PRINT-SCRIPT command, JOKER codes are displayed as the corresponding code number, with the sign # printed on the line below.

JOKER 1 is displayed as a left arrow: replaces any character, and no check is performed on the corresponding character.

JOKER 2 is displayed as a right arrow: no check is performed from this position up to the end of the string.

**CAUTION:**

When using JOKER 2 for answers of type ANY, the answer characters are read up to the number of characters included in the script answer. In such a case, if the JOKER is the last character, the possible extra answer characters will not be read anyway. To force the reading of the extra characters, the JOKER must be followed by dummy characters, up to a number corresponding to the maximum number of expected answer characters. There is no such problem with answers of the type STRING, as the answer characters are read up to the first CR code (or CH. WAIT is reached).

JOKER 3 is displayed as an up arrow: no error if the corresponding character is greater than the character following the JOKER sign. In this case, the JOKER code needs two characters in the SCRIPT but corresponds to a single character in the string under check. The compare is performed on the unsigned binary values from the character's seven lower bits.

JOKER 4 is displayed as a down arrow: the same as for the JOKER 3, but the condition is that the character is less than the one following the JOKER sign.

JOKER 5 is displayed as the graphic sign bottom edge. The JOKER consists of the JOKER sign, followed by two characters, C1 and C2. This means that there is no error if the character is within the (inclusive) range C1:C2.

JOKER 6 is displayed as the graphic sign top edge. The JOKER consists of the JOKER sign followed by two characters, C1 and C2. This means that there is no error if the character is not within the (inclusive) range C1:C2.

## 6.5 CONSTANTS AS PARAMETERS

The SCRIPT parameter fields may be replaced, instead of specifying directly the character string, by a CONSTANT defined within the running DESCRIPTION. A CONSTANT is referred to by its number, which is in the range 1 to 255 (i.e. up to 255 constants may be defined at a time). The format for specifying a constant in a parameter field is:

```
*=<constant number>
```

For example, a command string is defined in the DESCRIPTION with the reference number 19. The corresponding command in the SCRIPT is:

```
COMMAND   STRING    *=19
```

When a constant is found in the SCRIPT, the parameter field is replaced by the constant definition only if it is currently defined. Otherwise, the parameter takes the value of the constant specification string (e.g., in this case, *=19). As this is likely to have unpredictable results, it is recommended that all constants used in SCRIPTs be defined in the DESCRIPTIONs which govern them.

Beginning with Release C of Telefix, if the parameter is something other than a pure numeric value, it is possible to specify more than one constant as follows:

```
CODE  (SUBCODE)   *=n1,n2(,n3...)
```

where n1, n2, etc. are reference numbers to CONSTANT definitions in the running DESCRIPTION. In contrast to the case where a single CONSTANT is specified, all specified CONSTANTs must be defined. CONSTANT numbers are separated by commas, without any spaces intervening.

When several CONSTANTs are specified, execution of the SCRIPT proceeds as if the code had been written several times on adjacent lines, with each instance having the next CONSTANT, reading from left to right.

Example

The following CONSTANT definitions occur in the DESCRIPTION:

```
CONSTANT      1     ENTER-DIRECTORY,
CONSTANT      2     MY-DIRECTORY-1
```

The following command in the SCRIPT:

```
COMMAND   STRING    *=1,2
```

is the same as the series of commands:

```
COMMAND   STRING    *=1
COMMAND   STRING    *=2
```

The result is that the command

```
ENTER-DIRECTORY,MY-DIRECTORY-1
```

is carried out.

## 6.6 PROCESSING LISTS OF POSSIBLE ANSWERS

A list of possible answers is made of consecutive lines of command POS ANSW. But, any POS ANSW may be followed by an associated possible sequence started with the command POS SEQU and ended with the command END POS.

Example: Processing a list of possible answers(POS ANSW)

```
--------
POS ANSW   STRING   Answer 1
POS ANSW   STRING   Answer 2
POS SEQU
--------
--------
END POS    (subcode)
POS ANSW   STRING   Answer 3
POS ANSW   STRING   Answer 4
POS SEQU
--------
--------
END POS    (subcode)
POS ANSW   STRING   Answer 5
--------
```

This list has five possible answers, and two of them (answers 2 and 4) have an associated possible sequence. The possible sequence cannot include another list of possible answers, and cannot be placed anywhere else than following a POS ANSW.

When entering a list of possible answers, the MAX WAIT time is waited to receive an answer. Several cases may arise:

- No answer is received: The SCRIPT execution continues with the command immediately following the possible list.
- An answer is received which matches none of the POS ANSWs: An error message is issued (if ANSW ERR is enabled) and action is taken depending on the defined answer error condition. In the case of CONTINUE on answer error, the SCRIPT execution continues with the command immediately following the possible list. If ANSW ERR is disabled, no message is issued and the SCRIPT execution returns to the beginning of the possible list.
- An answer is received which matches a POS ANSW not followed by a POS SEQU: The SCRIPT execution returns to the beginning of the possible list.
- An answer is received which matches a POS ANSW followed by a POS SEQU: The possible sequence is executed up to the END POS, if no exit from the SCRIPT occurs during this sequence execution. The END POS subcode gives the action to do afterwards:

  CONTINUE: The SCRIPT execution continues with the command immediately following the possible list.

  RETURN: The SCRIPT execution returns to the beginning of the possible list.

  EXIT: The SCRIPT execution is stopped and control returns to the DESCRIPTION.

  ERROR EX: The SCRIPT is aborted and an error exit is issued to return to the DESCRIPTION where the defined action on error exit is entered.

If it is necessary to specify several possible lists in sequence in the same SCRIPT, the command CONTINUE may be used as a separator between the lists.

If POS ANSWs of only the type ANY appear in the list, the POS ANSW with the longest parameter field must be the first one in the list. If types ANY and STRING are in the list, a type STRING must be the first one. This is to prevent answers from being truncated.

## 6.7 THE DESCRIPTION COMMANDS

As for the SCRIPT, each line in a DESCRIPTION is a command which includes these three fields:

- command code
- subcode
- parameter

Some commands do not require a subcode and/or a parameter; in such cases the subcode and/or parameter field(s) are left blank. A summary of all of the DESCRIPTION commands, together with their subcodes and parameters where applicable, is given in the following table. Detailed descriptions of the commands are given in the sections immediately following this table.

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| COMMENT | LOCAL | The comment string, without control code. |
| COMMENT | DISPLAY | The comment string, without control code. |
| CONSTANT | ref numb | Constant string. |
| CONSTANT | ref numb | ?=Operator prompt message. |
| TIME OUT | SECONDS | Decimal value. |
| TIME OUT | MINUTES | Decimal value. |
| TIME OUT | PERCENT% | Decimal value. |
| CH. WAIT | -------- | Decimal value, number of seconds. |
| OP. WAIT | SECONDS | Number of seconds. |
| OP. WAIT | MINUTES | Number of minutes. |
| OP. BELL | SILENT | Number of seconds. |
| OP. BELL | ACTIVE | Number of seconds. |
| ERR EXIT | DISPLAY | The comment string, without control code. |
| ERR EXIT | LINK TO | Description name. |
| ERR EXIT | SCRIPT | Script name. |
| ERR EXIT | ABORT | -------- |
| SCRIPT | -------- | Script name. |
| LINK TO | -------- | Description name. |
| ENABLE | ECHO ERR | -------- |
| ENABLE | ANSW ERR | -------- |
| ENABLE | ALT ANSW | -------- |
| DISABLE | ECHO ERR | -------- |
| DISABLE | ANSW ERR | -------- |
| DISABLE | ALT ANSW | -------- |
| ENTRY | Entry No | -------- |
| IF TRUE | Flag No | -------- |
| IF FALSE | Flag No | -------- |
| IF CONST | Const No | Any string. May include control codes. |
| ELSE | -------- | -------- |
| END IF | -------- | -------- |
| ABORT | -------- | -------- |
| OPEN LOG | -------- | Log file name. |

### 6.7.1 COMMENT

The parameter is the comment string terminated by CR. If the subcode is LOCAL, then no action is taken. This is only a local comment in the DESCRIPTION. If the subcode is DISPLAY, the comment is written on the operator terminal.

### 6.7.2 CONSTANT

The subcode is the constant reference number. The parameter is the constant string, which may include control codes and JOKERs, unless it starts with ?= . In this last case, the operator will be prompted to enter the constant from the terminal keyboard, after the message following ?= is displayed.

A constant may be defined or redefined at any place in the DESCRIPTION, and up to 255 constants may be defined within a DESCRIPTION. A defined constant may be transferred into any parameter field of a SCRIPT (see page 73).

For the operator to respond to a constant prompt, a useful way to proceed is to define first the concerned constant with a fixed string used as a default value. On the operator prompt command, the old contents are displayed and ready for editing, and Telefix will wait up to 15 seconds by default (or up to the time specified by the last OP. WAIT command) to get an operator input.

The operator has three alternatives:

- He/she gives an input or edits the old value; it becomes the new definition for the constant.
- He/she cancels the input with the home cursor key; the prompt is requested again.
- He/she does not give any input within the delay; the constant definition remains unchanged, either undefined or with the old contents.

### 6.7.3 TIME OUT

The parameter is a decimal value. It is used to increase the MAX WAIT time found in the SCRIPT. The subcode indicates either SECONDS or MINUTES which means that MAX WAIT is increased by a fixed value; or PERCENT% which means that MAX WAIT is increased by a percentage of its own value.

### 6.7.4 CH. WAIT

The parameter value is the number of seconds the program will wait to receive characters from the remote system. It may be used when receiving answers, when checking the echo, and before sending a command to see if there are any unexpected answers. It is possible to break the automatic mode with the ESCAPE key during this waiting time, but it is recommended that you not use large values. By default, the automatic mode starts with 3 seconds as the CH. WAIT value.

### 6.7.5 OP. WAIT

Note: The OP. WAIT command for DESCRIPTION programming is not the same as the OP. WAIT command for SCRIPT programming.

The SCRIPT command
1) sets up a wait-state with no maximum wait time,
2) can control bell activation, and
3) can send a message to the operator terminal.

The DESCRIPTION command
1) sets up a wait-state of fixed maximum duration,
2) cannot control bell activation, and
3) cannot send a message to the operator terminal

The OP. WAIT command subcode is either SECONDS or MINUTES, and the parameter is the number (of seconds or minutes), i.e. the maximum amount of time to wait for input from the operator during constant prompts. The default value when starting the execution of a DESCRIPTION is 15 seconds.

### 6.7.6 OP. BELL

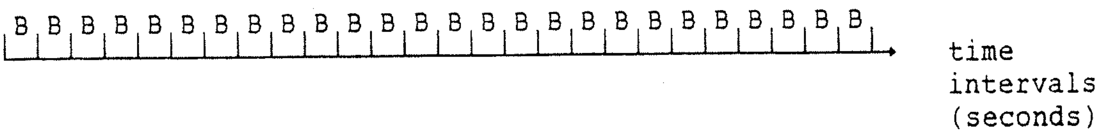
This DESCRIPTION command works in the same way as the SCRIPT command of the same name. The subcode is ACTIVE or SILENT. The parameter is the time in seconds. This command determines the activity of the operator terminal bell for subsequent constant prompts.

The bell has alternating ACTIVE and SILENT periods. The duration of each (in seconds) is specified by the parameter in the OP. BELL command with the respective subcode.

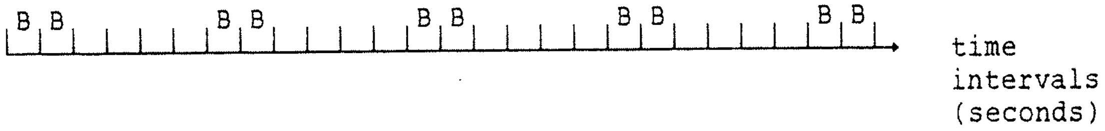
The default values when starting a DESCRIPTION execution are 1 second ACTIVE and 0 seconds SILENT, i.e. a continuous bell tone.

Diagrammatically, this may be pictured as follows: (B=bell tone)

```
Commands:     OP. BELL  ACTIVE  1    (the default situation)
              OP. BELL  SILENT  0
```



```
Commands:     OP. BELL  ACTIVE  2
              OP. BELL  SILENT  4
```



### 6.7.7 ERR EXIT

This tells what action to take if a SCRIPT execution is aborted by an error. The subcodes have the following meanings:

DISPLAY: The parameter is a comment string to be written to the operator terminal. It is not exclusive with the following subcodes, and the message remains valid until it is redefined or cancelled by giving another ERR EXIT DISPLAY command with an empty parameter field. If this subcode is the last ERR EXIT defined condition, no action other than displaying the message will be taken in case of error exit from the SCRIPT, whatever the previously defined condition was. An ERR EXIT DISPLAY with no parameter resets the previously defined conditions.

LINK TO: The present DESCRIPTION execution is aborted, and a new one is immediately entered. The parameter is the name for the new DESCRIPTION.

SCRIPT: A new SCRIPT execution is immediately started. The parameter is the name for this SCRIPT.

ABORT: The automatic mode is aborted.

### 6.7.8 SCRIPT

The parameter gives the SCRIPT name. There is no subcode. The specified SCRIPT execution is started. Control will return to the DESCRIPTION when the SCRIPT execution is finished, or if an error exit occurs.

### 6.7.9 LINK TO

The present DESCRIPTION execution is aborted, and the one corresponding to the name found in the parameter is entered.

### 6.7.10 ENABLE

Used to enable the function found in the subcode. The possible subcodes are:

ECHO ERR: If an echo error is found, the corresponding error message is displayed.

ANSW ERR: If an answer error or an unexpected answer is found, the corresponding error message is displayed.

ALT ANSW: If the answer does not match the expected one, alternative answers may be looked at, if any are specified.

By default, these conditions are all enabled for the AUTOMATIC-MODE command, and all are disabled for the RUN-SCRIPT command, unless the default values are inverse (see INVERSE-DEFAULT-PARAMETERS command).

### 6.7.11 DISABLE

Used to disable the function found in the subcode. The possible subcodes are:

ECHO ERR: If an echo error is found, the corresponding error message is not displayed, and an error exit cannot be issued.

ANSW ERR: If an answer error is found, the corresponding error message is not displayed, and an error exit cannot be issued. No further checking for the presence of unexpected answers is performed.

ALT ANSW: No further checking for the presence of possible alternative answers is performed.

### 6.7.12 ENTRY

Defines an entry point for a PROG. EX (programmable exit) from the script. The subcode is the entry point number (1 to 255). When the script execution is stopped by a programmable exit, control returns to the DESCRIPTION at the corresponding entry point, if defined. If not defined, controls return in the normal way at the next DESCRIPTION command line to be executed.

### 6.7.13 IF TRUE

This command is used to test the state of a programmable flag set or reset during the script execution (see SET FLAG and RESET FL). The subcode is the flag number (1 to 255). There is no parameter.

If the condition is satisfied, the DESCRIPTION commands following the IF line are executed up to the ENDIF, or the ELSE, if present. If the condition is not satisfied, a skip after the END IF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### 6.7.14 IF FALSE

This command is used to test the state of a programmable flag set or reset during the script execution (see SET FLAG and RESET FL). The subcode is the flag number (1 to 255). There is no parameter.

If the condition is satisfied, the DESCRIPTION commands following the IF line are executed up to the ENDIF, or the ELSE, if present. If the condition is not satisfied, a skip after the END IF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### 6.7.15 IF CONST

This command is used to test the contents of a constant (see "CONSTANTS as parameters", page 73). The subcode is the constant number and the parameter the string to compare with, which may include control codes and JOKERs.

If the condition is satisfied, the DESCRIPTION commands following the IF line are executed up to the ENDIF, or the ELSE, if present. If the condition is not satisfied, a skip after the END IF or the ELSE is executed. It is possible to nest IF sequences up to 100 levels.

### 6.7.16 ELSE

This command indicates that DESCRIPTION execution will continue with the statement immediately following the ELSE, in the event that the test condition for the associated IF TRUE, IF FALSE, or IF CONST is not satisfied. There is no subcode and no parameter.

### 6.7.17 END IF

This command indicates that unconditional execution of the DESCRIPTION will be resumed with the statement after the END IF, following the execution of the test condition, and possibly also of intervening code. There is no subcode and no parameter.

### 6.7.18 ABORT

There is neither subcode, nor parameter. The automatic mode is aborted and control returns to the Telefix command processor. This command is also executed when reaching the end of the DESCRIPTION.

### 6.7.19 OPEN LOG

This command has no subcode. The parameter is the name of a file to be used as a log file. The command opens the specified log file for use. If a log file was already in use, it is closed before the new one is opened.

# CHAPTER 7

# A DEBUG MODE FOR RUNNING SCRIPTS AND DESCRIPTIONS

# 7 A DEBUG MODE FOR RUNNING SCRIPTS AND DESCRIPTIONS

## 7.1 GENERAL INFORMATION

The automatic mode commands AUTOMATIC-MODE and RUN-SCRIPT can now be run in debug mode. Debug mode is turned ON and OFF with the command DEBUG-MODE-ON-OFF. When debug mode is ON, execution of either the AUTOMATIC-MODE or the RUN-SCRIPT command entails activation of the debugger program. This program will then accept the following commands:

```
HELP
ACTIVE-ROUTINES
DISPLAY <parameter> (<parameter>...)
PROGRAM-INFORMATION
TRACE
RESET-TRACE
STEP (<count>)
BREAK-DESCRIPTION <line> (<count>)
BREAK-SCRIPT <line> (<count>)
RESET-BREAKS
INITIALIZE-TERMINAL <terminal type>
CONTINUE
EXIT
```

Descriptions of these commands can be found in the following sections of this chapter.

### 7.1.1 HELP

Format: HELP

Lists the debugger commands, and their syntax.

### 7.1.2 ACTIVE-ROUTINES

Format: ACTIVE-ROUTINES

Displays the active SCRIPT ROUTINEs (see page 71), as well as the script line numbers from which they were called. The current DESCRIPTION and SCRIPT names and lines are also displayed.

### 7.1.3 DISPLAY

Format: DISPLAY <parameter> (<parameter>...)

The current value(s) of the specified parameter(s) is/are displayed. The following parameters may be specified:

| Parameter | Information Displayed |
|---|---|
| ANSWER-ERROR | Current answer error condition |
| CHARACTER-WAIT | Current character wait time |
| CONSTANT <constant number> | Current contents of the specified constant |
| COMMAND-LINE | Next DESCRIPTION or SCRIPT line to be executed |
| ECHO-ERROR | Current echo error condition |
| FLAG <flag number> | Status of the specified programmable flag |
| LAST-ANSWER | Last received answer from the remote system |
| MAX-WAIT | Current maximum wait time |
| REJECTED-CHARACTERS | Number of rejected characters since the SCRIPT start |

### 7.1.4 PROGRAM-INFORMATION

Format: PROGRAM-INFORMATION

The Telefix STATUS command is executed (see page 32), and information on the status of the debugger program is displayed.

### 7.1.5 TRACE

Format: TRACE

This command activates the TRACE function. Before the execution of each SCRIPT line, the SCRIPT line is written to the log file, the operator terminal, and the follow-up terminal.

Note: The parameter-oriented Telefix trace command of Release B is no longer supported.

### 7.1.6 RESET-TRACE

Format: RESET-TRACE

This command resets the TRACE function.

### 7.1.7 STEP

Format: STEP (<count>)

This command permits stepping through the execution of a DESCRIPTION or SCRIPT. If <count> is not specified, the step-size is one, i.e. the next line is executed, and control is returned to the debugger program which waits for another command. If <count> is specified, that number of DESCRIPTION or SCRIPT lines are executed before the debugger program expects another command.

### 7.1.8 BREAK-DESCRIPTION

Format: BREAK-DESCRIPTION <line> (<count>)

A break point is set at the specified line for the DESCRIPTION execution. If <count> is specified, the break point will be passed count-1 times before it is executed.

### 7.1.9 BREAK-SCRIPT

Format: BREAK-SCRIPT <line> (<count>)

A break point is set at the specified line for the SCRIPT execution. If <count> is specified, the break point will be passed count-1 times before it is executed.

### 7.1.10 RESET-BREAKS

Format: RESET-BREAKS

Resets the DESCRIPTION and SCRIPT breaks.

### 7.1.11 INITIALIZE-TERMINAL

Format: INITIALIZE-TERMINAL <terminal type>

This debug mode command is the same as the corresponding Telefix command (see page 33).

### 7.1.12 CONTINUE

Format: CONTINUE

Control is returned to continue the execution of the automatic mode (RUN-SCRIPT or AUTOMATIC-MODE commands). The debugger program will be re-entered at the next break point, if any exists.

### 7.1.13 EXIT

Format: EXIT

This command stops the debugger program and the automatic mode, and returns control to the Telefix command processor.

# CHAPTER 8

# USE AND COMMANDS OF THE DESCRIPTION AND SCRIPT EDITORS

## *8 USE AND COMMANDS OF THE DESCRIPTION AND SCRIPT EDITORS*

The DESCRIPTION editor and the SCRIPT editor are in fact the same program, and so work in the same way. The DESCRIPTION editor is identified by the prompt DED: at HOME position, while the SCRIPT editor prompts with SED:.

### *8.1 HOME COMMANDS*

| | |
|---|---|
| F | = Display the first window |
| L | = Display the last window |
| N or cursor DOWN | = Display the next window |
| P or cursor UP | = Display the previous window |
| + | = Advance the displayed window by 5 lines |
| - | = Rewind the displayed window by 5 lines |
| M | = Move to line <area/line number> |
| D | = Delete area <area/from> <to> |
| C | = Create new area <new area name> |
| H or ? | = Help |
| E | = Exit editor, return to Telefix commands |
| CR | = Go to first displayed line, first field |
| LF | = Go to displayed line 5, first field |
| HOME | = Go to last referenced field and line |
| R | = Retrieve <area name> |
| S | = Search for CODE or SUBCODE |
| U | = Substitute one character string for another in the parameter fields. |
| W | = Store <file name> |
| X | = Display file information |
| G | = Get parameter field for string occurrence |
| A or CTRL+B | = Set/reset append mode |
| SPACE | = Redisplay the present window |

HOME command: F

The first window is displayed, lines 1 to 20. A window is 20 lines long.

HOME command: L

The last window is displayed, lines $-19 to $. ($ refers to the last existing line.)

HOME command: N or cursor DOWN

The next window is displayed. If this next window has nonexistent lines, the last window is displayed.

HOME command: P or cursor UP

The previous window is displayed.

HOME command: +

The displayed window is advanced by five lines. If this advance would result in the end of an area being reached, then the displayed window is only advanced by the number of lines remaining in the area.

HOME command: -

The displayed window is rewound by five lines.

HOME command: M

Move the displayed window. If a line number is specified and it is an existing line in the selected area, the displayed window is adjusted to start with this line. Otherwise, an existing area name is expected, and its first window is displayed. The default area is selected by giving a Carriage Return as a response to the area name prompt. The parameter <line number> is a decimal value in the range 1 to last line number. The last line number may also be specified by $, with a possible negative displacement. Example: $-3 means the line which is three positions before the last one.

HOME command: D

Delete function. If a line number is specified and it is an existing line in the selected area, a second line number is requested; lines are deleted including these two specified limits. Otherwise, an existing area name is expected; the corresponding area is fully deleted and does not exist any more. It is not possible to delete the default area. $ may be used in line numbers in the same way as for the Move command (HOME command: M).

HOME command: C

Create a new area which is appended to the ones already existing. The cursor is positioned to enter the first line.

HOME command: H or ?

HELP information is displayed. The keys cursor UP and DOWN are used to move the HELP pages. The HOME commands are still available.

HOME command: E

Exit command. The editor is aborted and the Telefix commands are available.

HOME command: CR (Carriage Return)

The cursor goes to field 1 (code) in the first displayed line. If no line exists yet, the first line is created.

HOME command: LF (Line Feed)

The cursor goes to field 1 (code) in the displayed line 5 (or in the last line). If no line exists yet, the first line is created.

HOME command: HOME

Move the cursor back into the displayed window, at the place it was last positioned, or at first line if this is the first time it goes into the window.

HOME command: R

Retrieve command. This command is also executed when calling the editor together with an area name. If there is no line already entered, the whole file is retrieved and the specified area selected for display and editing. If lines are already entered, it asks for a clear or not before retrieving the new file. If the answer is Yes, the whole file is retrieved in the same way as the previous case. If the answer is No, you are requested to give either:

- a new area name: the specified area in the R command is appended to the existing ones with the new name.
- a line number in the range 1 to last line for the presently selected area: the specified area in the R command is inserted into the selected area, after the given line number. The default value is: append after last line in the selected area.

HOME command: S

Search for a code or a subcode in the selected area. All existing codes and subcodes are displayed on the last screen line. Cursor keys are used to select one, and the Carriage Return key to start the search function (the HOME key may be used to cancel the search command). The cursor is moved to the next corresponding code or subcode.

HOME command: U

This command searches the parameter fields for occurrences of a specified string, and replaces those occurrences with another string. It is possible to perform an automatic substitution of all occurrences, or to have a manual check.

Note: Substitutions can be performed only in parameter fields which do not contain purely numeric values. Furthermore, a CONSTANT reference (*=x) must be exactly replaced by another one.

HOME command: W

The store command. All areas are stored on the specified file. This is the only Telefix command which allows building up of multiarea DESCRIPTION and SCRIPT files.

HOME command: X

Display file information. The area names, with the corresponding number of lines, are displayed. An arrow points to the currently selected area. The default file name for R or W command is also shown. The keys cursor UP and DOWN are used to move the information pages. The HOME commands are still available.

HOME command: G

Get (search for) a string in the parameter fields in the selected area. The cursor is moved to the next parameter field where the specified string occurs.

HOME command: A or CTRL+B

The append mode, used to enter new lines, is set or reset.

HOME command: SPACE

The terminal is re-initialized, and the selected window redisplayed.

## 8.2 EDITING COMMANDS

After CR, LF or HOME has been given from the the HOME position, the cursor is positioned in the selected window. If no line already exists, the line number 1 is created. If the cursor is moved after the last line, an extra line is also created. The following keys are used when navigating in the window:

| | |
|---|---|
| HOME | = Go to HOME position (HOME commands). |
| Cursor UP | = Move cursor to previous line. |
| Cursor DOWN | = Move cursor to next line. |
| CR | = Move cursor to next line and first field, or append if the append mode is set. |
| LF | = Move cursor 5 lines down. |
| Cursor LEFT | = Move to previous field in same line. |
| Cursor RIGHT | = Move to next field in same line. |
| D or DELETE | = Delete the line where the cursor is located. |
| R or CTRL+W | = Restore last deleted line. |
| I or CTRL+L | = Insert a new line. |
| A or CTRL+B | = Set/reset append mode. |
| E | = Enter or edit the field where the cursor is located. |
| S | = Search for next CODE or SUBCODE. |
| G | = Get next parameter field for string occurrence. |

## 8.3 FIELD ENTRY AND EDITING

After the cursor has been positioned on the desired field and E (edit) pressed, or when creating a new line, the field entry operation starts. For fields 1 and 2 (code and subcode), the several possible alternatives are displayed on the last screen line, starting at the lower right corner. The cursor keys are used to move the cursor to the several possibilities, and then, when the choice is made, CR is used to validate the selection. The HOME key may be used to cancel the entry and to restore the line. The cursor may be automatically moved to a next field for editing if the previously entered one requests an extra entry.

The type of entry to be made for field 3 (parameter) is displayed on the last screen line. If a numeric value is requested, it is not possible to enter something else, except a constant reference. If a file name is requested, any displayable character may be entered but no check is performed for the file name syntax; be careful. If a string is requested, any displayable character may be entered, but no control code. If any characters are requested, any displayable code is normally entered while one control code may be entered only if CTRL+O is typed first; CTRL+O must be repeated for each control code to be entered. If JOKERs may be entered, CTRL+J is typed first and the JOKER code follows. It is not possible to enter the control-code CR, which is used to terminate the line entry. If field 3 was not empty, the old content is still available and the cursor is positioned after the last character, waiting for more characters or for the control codes used for editing any string (see section 8.4).

## 8.4 LINE EDITING

When entering field 3 or when giving parameters for a HOME command, it is possible to use the following control codes for editing:

| | |
|---|---|
| HOME | = For a home command parameter entry:<br>cancel command and return to HOME position.<br>= When entering field 3:<br>cancel entry, restore previous contents. |
| Cursor LEFT | = Move to previous character. |
| Cursor RIGHT | = Move to next character. |
| CTRL+F char. | = Move forward to specified character.<br>char.= CTRL+F move beyond last character. |
| CTRL+R char. | = Move backwards to specified character.<br>char.= CTRL+R move to beginning of line. |
| CTRL+A or DELETE | = Delete one character. |
| CTRL+D char. | = Delete up to and including cursor position.<br>char.= CTRL+D delete the entire line.<br>char.= CR delete the rest of the line.<br>char.= CTRL+R delete from beginning of the line. |
| CTRL+K or CTRL+Q | = Delete the whole line. |
| CTRL+E | = Set/reset expand mode. |
| CTRL+O | = Next is a control code (except CR, may be DEL):<br>only legal to enter a special type of parameter. |
| CTRL+J | = Next is a JOKER code: only legal to enter a special type of parameter. |
| CR | = Terminate the entry. |
| CTRL+V | = Verify character value. |

## 8.5 EDITING FUNCTIONS FOR WORKING WITH MARKED AREAS (MARKED BLOCKS)

| | |
|---|---|
| FUNC CTRL+Z (or MARK)<br>or<br>FUNC CTRL+V (or SHIFT + MARK) | = Mark the block limit |
| FUNC X (or CANCEL) | = Cancel the current block marking |
| FUNC Q (or SHIFT + CANCEL) | = Re-mark the last-marked block |
| FUNC D (or DELETE) | = Delete the marked block |
| FUNC C (or COPY) | = Copy the marked block<br>(insert it at the current position) |
| FUNC I (or SHIFT + COPY) | = Replace cursor position block with marked block |

FUNC M (or MOVE) = Move the marked block
(insert it at the current position)

The last three functions (Copy, Replace, Move) work between two different editor areas, i.e. the marked block may be in another area than the one which is selected and displayed.

# CHAPTER 9

# LOG FILE FORMATS

## 9 LOG FILE FORMATS

The log files written by Telefix follow some rules explained in this chapter.

The line traffic characters are recorded with bit 7 (usually used as the parity bit) forced to zero. The Telefix messages are also recorded with character bit 7 forced to zero, except for characters which will not have to be displayed by the program designed to analyze the log files. These characters have bit 7 forced to one, and are mainly the CR/LF used to terminate the message lines. If the codes CR and LF must be displayed only, without terminating the line, they have to be respectively recorded as 315B and 312B (Bits 6 and 7 forced to one).

A Telefix message is identified by starting a line with the following pattern:

```
==TFXxy=>
```

followed by a space and the message text. If the message continues on the next line, it starts with:

```
========>
```

followed by a space and the message text. The code x and the subcode y are used by the program for log file analysis to recognize the message type.

LOG FILE code and subcodes

| Code | Subcode |
|---|---|
| 0 = Message | 0 = Comment |
| | 1 = Operator prompt |
| | 2 = Follow-up terminal information |
| | 3 = Wait for operator |
| | 4 = Debug mode information |
| | 5 = System message |
| 1 = Start for | 0 = Manual mode |
| | 1 = Automatic mode |
| | 2 = DESCRIPTION file |
| | 3 = SCRIPT file |
| 2 = End for | Same as for code 1 |
| 3 = Reported error | 0 = Communication error |
| | 1 = Echo error |
| | 2 = No echo |
| | 3 = No answer |
| | 4 = Wrong answer |
| | 5 = Unexpected answer |
| 4 = System error | 0 = File system error |
| | 1 = Run-time error |
| | 2 = Hardware error |
| | 3 = Fatal error |
| | 4 = Incorrect command |
| 5 = Telefix action | 0 = Abort job |
| | 1 = Skip to next SCRIPT command |
| | 2 = Repeat previous SCRIPT command |
| | 3 = Operator break |

# CHAPTER 10

# THE LOG-ANALYZER PROGRAM

## 10 THE LOG-ANALYZER PROGRAM

This program is intended for working with log files produced by Telefix or any other subsystem using the same formats. It permits displaying the log data on a terminal screen, and going through it in page mode. Search functions are available, as well as functions to write statistics and reports.

The log-analyzer program may be started with a file name following the LOG-ANALYZER command. In this case, the program does an automatic recover with this specified file, and, if the recover command is successfully terminated, a short log status command (HOME command: T). The default file type for a log file is :LOGS.

The log data is read only, never written or modified by the log-analyzer program.

> *CAUTION*:
>
> If a log file is read and written back with an editor, data is destroyed !

Description of the commands:

Commands are given with the cursor in the home position (behind LOG:) or when navigating in the displayed page.

HOME commands:

H or ? = Display help menu.

! = Enter a new terminal type.

SPACE = Re-initialize the terminal and redisplay the current page.

@ = Execute a SINTRAN command.

C = Display codes definitions.

R = Retrieve a log file. The other commands cannot be executed as long as the retrieve command is not executed. The log data is transferred to the scratch file with lines tuned to 80 characters maximum length.

J = Activate TELEFIX program.

E = Exit LOG-ANALYZER program.

Commands to display the log (HOME commands):

F = Display the first page. A page is 20 lines of 80 characters.

L = Display last page.

N or cursor down = Display next page.

P or cursor up = Display previous page.

+ = Move the displayed page five lines forward.

- = Move the displayed page five lines backward.

M = Move to specified line number. $ (last line number) may be used for relative values.

Commands to get outputs from the log (HOME commands):

O = Output to a specified file. The log data is written in the same form as it appears on the terminal screen, including the log file format messages.

D = Output data to a specified file. The log data is written without the log file format messages.

T = Short log status. General information concerning the log file is displayed on the screen.

W = Write report. The short log status command (T) is executed, and all log messages are written to the specified file together with the line number where they come from.

Search commands (HOME or navigation commands):

G = Get a string.

S = Search for a log message. The message title is five characters long, corresponding to the identification and codes shown in the chapter "log file formats". Replace the last two stars (meaning any character) with the code searched for, and the first three stars with TFX if other identification is also present in the log.

HOME commands to go in navigation mode:

CR = Go to first page line, first character.

LF = Go to line five, first character.

HOME = Return into page.

Navigation commands:

The character coordinates and the value corresponding to the cursor position is continuously displayed at the screen top.

Cursor right/left/up/down = Used to move in the page.

CR = Move to next line, first character.

LF = Move 5 line down, first character.

CTRL+T = Move 10 characters right in the same line.

CTRL+Y = Move 10 characters left in the same line.

CTRL+F = Move forward up to specified character.

CTRL+R = Move reverse up to specified character.

HOME = Return to HOME commands.

G = Get next string, or get string if no string already specified.

S = Search next message, or search message if no message identification already specified.

# CHAPTER 11

## SOLVING PROBLEMS ENCOUNTERED WHILE USING TELEFIX

# 11 SOLVING PROBLEMS ENCOUNTERED WHILE USING TELEFIX

This chapter should help those who encounter problems when using Telefix.

## 11.1 NO ANSWER ON REMOTE CONNECTION

When the physical connection has been established, it is advisable to try the MANUAL-MODE first to see if the communication works correctly. Do not forget the EXIT character which will allow you to leave the manual mode. If, unfortunately, the remote system does not answer when you press ESCAPE while SINTRAN is running, or when you press any other key while in stop mode, the following points must be checked:

- The line is not lost: the carrier signal from the modem is still present. This may be checked by looking directly at the modem indicators or by using the Telefix command STATUS.

- The Telefix Adaptor is correctly connected and placed in REMOTE position at the remote site. The system is in the expected state, with SINTRAN running, or in stop mode with OPCOM active.

- The transfer speeds for the remote system console channel, the two modems, and the Telefix center communication line are the same. The last one may be changed by the SET-TRANSFER-SPEED command.

- The interface and driver on both sides are compatible concerning the number of data bits, stop bits, and parity.

- The modems on both sides are compatible, and the connection is established in both directions rather than in just one.

- The modem cables are correctly connected and the Telefix center communication interface is set up in a V24 mode, and not in a current loop mode.

- The telephone line quality may be poor. Try calling again in order to pick up another line from the public network.

If the communication runs, but with rubbish or lost characters:

- Check the modem compatibility, interface, and drivers compatibility for frame configuration.

- Try telephoning again to work with another public network line.

- Try a lower transmission speed, if possible.

Remember that it is also possible to perform a modem test when using the TEST-MODEM command (formerly part of the CONNECT-LINE command). The modem at the Telefix center side may be tested alone if it is possible to place it in a test mode configuration (control signals active, and data out and in looped together). If this is the modem at the remote side which is placed in test mode, both modems plus the communication line are tested.

## 11.2 COMMUNICATION DEAD

The communication may suddenly die due to hardware failures in the line itself: the public network line crashed or the modems have lost their connection. Another not so obvious reason may also kill the communication: if a power failure occurs at the Telefix system and the interface transfer speed has been changed from its default value with the SET-TRANSFER-SPEED command. The default value is restored again when the system restart occurs, and it no longer corresponds with the communication speed.

## 11.3 NO ACTION WHEN GIVING SINTRAN COMMANDS TO THE REMOTE SYSTEM

The communication is established, but any time a SINTRAN command is given, the remote system does not execute it or gives an error message. The echo returned from your input looks correct, but be aware that nonprintable codes may be inserted in your command string. To see if that is the case, open a log file and type a command to the remote system. Then, if the command does not operate, type CTRL+D to repeat the same command string and to have it recorded to the log file. Look at the log file to detect any extra rubbish characters and see what they are. It is very likely that you will find extra DEL codes (all one bit) coming from extra signals injected on the phone line, probably at the remote system side. This happens with special electronic switchboards which use a scanning system for lines, made of signals with about the same frequency as the modem carrier.

## 11.4 PROBLEMS WITH SEND-FILE AND GET-FILE INITIALIZATION

The execution of the file-transfer commands requires that Telefix log in to the remote system to start the receiver or transmitter program. The log-in procedure is displayed at the operator terminal. It may fail if the remote-system background processor is already working. In such a case, Telefix may not be able to abort it, and you will have to enter the manual mode to kill the running program before restarting the transfer command.

## 11.5 PROBLEMS WHEN USING THE DESCRIPTION AND SCRIPT EDITORS

The DESCRIPTION/SCRIPT editor works with terminals having a screen size at least equal to 24 lines of 80 columns.

When entering characters in a parameter field, it is allowed to enter characters up to the last line position at column 80. Then, if the cursor goes out to the next line's first position it means that the conditions "beginning of line wrap" and "end of line wrap" are not both turned ON. Check your terminal to correct the corresponding options.

The spaces are displayed as a half sparse box (≡), but this is a graphic character which may be not available on your terminal type. In such a case, it is usually replaced by the equals sign (=).

## 11.6 PROBLEMS WHILE CREATING OR REGENERATING SCRIPTS

During a SCRIPT creation, incorrect sequences may be generated if the operator types the command characters to the remote system too much in advance. It is necessary to wait for all echoes during a command entry before terminating the command line, usually with a CR code, to get the sequences of COMMANDs and ECHOs registered properly.

It may be difficult or impossible to regenerate a SCRIPT if the source SCRIPT is too dissimilar from what it should be, especially if using a full SCRIPT and not a master SCRIPT. If there are more answers arriving than are specified in the source SCRIPT, the effect on the destination SCRIPT is identical to what happens during a CREATE-SCRIPT when the operator types ahead too many command characters. You may try to increase the parameter <character wait time>, which is used as a time-out to finish the received answers not terminated with a CR. It is also possible that several regeneration passes will give a result which becomes more and more accurate.

# CHAPTER 12

## EXAMPLES OF SCRIPT AND DESCRIPTION PROGRAMMING

## 12 EXAMPLES OF SCRIPT AND DESCRIPTION PROGRAMMING

Example 1. Script file to log in as user FLOPPY-USER

The control codes are shown like this: <xxB>

Script file name: EXECUTE-LOGIN

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| TYPING | AHEAD | 1 |
| ECHO ERR | EXIT | 0 |
| ANSW ERR | EXIT | 0 |
| COMMAND | ANY | <33B> |
| ECHO | NONE | |
| MAX WAIT | SECONDS | 8 |
| ANSWER | NO CARE | |
| COMMAND | STRING | FLOPPY-USER |
| ECHO | FULL+ LF | |
| ANSWER | ANY | PASSWORD: |
| COMMAND | STRING | |
| ECHO | FULL+ LF | |
| ANSWER | STRING | |
| ANSWER | STRING | |
| ANSWER | STRING | OK |
| ANSWER | ANY | @ |

Example 2. Description file to run the previous script

Description file name: TRY-LOGIN

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| COMMENT | LOCAL | Log in as user FLOPPY-USER |
| TIME OUT | PERCENT% | 20 |
| CH. WAIT | | 3 |
| ERR EXIT | DISPLAY | System does not answer to log in. Try log out. |
| ERR EXIT | LINK TO | TRY-LOGOUT |
| SCRIPT | | EXECUTE-LOGIN |
| COMMENT | DISPLAY | User FLOPPY-USER is entered |

Example 3. Description file entered, if error exit from script EXECUTE-LOGIN

Description file name: TRY-LOGOUT

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| COMMENT | LOCAL | Log out command |
| TIME OUT | SECONDS | 10 |
| CH. WAIT | | 5 |
| ERR EXIT | DISPLAY | Definitely no answer from remote system. |
| ERR EXIT | ABORT | |
| SCRIPT | | EXECUTE-LOGOUT |
| COMMENT | DISPLAY | One user was already logged in. Retry ENTER |
| SCRIPT | | EXECUTE-LOGIN |
| COMMENT | DISPLAY | User FLOPPY-USER is entered |

Example 4. Script file to execute the log out command

Script file name: EXECUTE-LOGOUT

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| TYPING | FULL | |
| ECHO ERR | REPEAT | 1 |
| ANSW ERR | REPEAT | 2 |
| COMMAND | STRING | LOGOUT |
| ECHO | FULL+ LF | |
| MAX WAIT | SECONDS | 5 |
| DELAY | SECONDS | 5 |
| ANSWER | NO CHECK | |
| ANSWER | STRING | --EXIT-- |

Example 5. Single script to enter as user FLOPPY-USER, and also if a user is already logged in, by using the DO command on answer error

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| TYPING | AHEAD | 1 |
| ANSW ERR | DO | |
| COMMAND | STRING | LOGOUT |
| ECHO | NO CHECK | |
| ANSWER | NO CARE | |
| END DO | CONTINUE | |
| ECHO ERR | EXIT | 0 |
| COMMAND | ANY | <33B> |
| ECHO | NONE | |
| ANSWER | NO CARE | |
| COMMAND | STRING | FLOPPY-USER |
| ECHO | FULL+LF | |
| ANSWER | ANY | PASSWORD: |
| COMMAND | STRING | |
| ECHO | NONE | |
| ANSWER | STRING | |
| ANSWER | STRING | |
| ANSWER | STRING | OK |
| ANSWER | ANY | @ |

Example 6. A script part where the communication is controlled by sending CTRL+Q in case of echo error, to cancel the command input string and to re-send it

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| ECHO ERR | DO | |
| COMMAND | ANY | CTRL+Q |
| ECHO | NONE | |
| ANSWER | NO CHECK | |
| ECHO ERR | EXIT | 0 |
| END DO | CONTINUE | |
| COMMAND | ANY | LIST-USER,,, |
| ECHO | FULL | |
| ECHO ERR | EXIT | 0 |
| COMMAND | STRING | |
| ECHO | FULL+ LF | |

Example 7. The same example as the previous one, but the DO sequence is general for all commands and allows 2 retries

```
 CODE      SUBCODE    PARAMETER

ECHO ERR   DO + REP   2
COMMAND    ANY        CTRL+Q
ECHO       NONE
ANSWER     NO CHECK
END DO     RETURN
--------   --------   --------
```

Example 8. Example of JOKER use

On the SINTRAN command LOGOUT, you must check that the date is greater than 18 June 1983, and not after the last day of June. The standard answer formats on LOGOUT will be, for example:

```
 06.11.12      23 JUNE   1983
```

The corresponding script part must be:

```
 CODE      SUBCODE    PARAMETER

COMMAND    STRING     LOGOUT
ECHO       FULL+ LF
ANSWER     STRING      11.11.11     138 JUNE   1983
                       ## ## ##      #
ALT ANSW   STRING      11.11.11     311 JUNE   1983
                       ## ## ##     # #
ANSWER     STRING     --EXIT--
```

The time is replaced by JOKER 1, no check. The first answer is true if the day is greater than 18 and less than 20. The alternative answer is true if the day is greater than 19.

## Example 9. Example of running MOVER test program

| CODE | SUBCODE | PARAMETER |
|---|---|---|
| TYPING | AHEAD | 1 |
| ECHO ERR | CONTINUE | |
| ANSW ERR | CONTINUE | |
| COMMAND | ANY | <33B> |
| ECHO | NONE | |
| MAX WAIT | SECONDS | 5 |
| ANSWER | NO CARE | |
| COMMAND | STRING | FLOPPY-USER |
| ECHO | FULL+ LF | |
| ANSWER | ANY | PASSWORD: |
| COMMAND | STRING | |
| ECHO | FULL+ LF | |
| ANSWER | STRING | |
| ANSWER | STRING | |
| ANSWER | STRING | OK |
| ANSWER | ANY | @ |
| COMMAND | STRING | MOVER |
| ECHO | FULL+ LF | |
| ANSWER | STRING | |
| ANSWER | STRING | |
| ANSWER | STRING | MOVING MEMORY TEST PROGRAM |
| ANSWER | STRING | |
| ANSWER | STRING | HAR-1863C    APRIL 29, 1977 |
| ANSWER | STRING | |
| ANSWER | STRING | MAXIMAL MEMORY ADDRESS: 177777 |
| ANSWER | STRING | |
| ANSWER | ANY | DO YOU KNOW THIS PROGRAM (Y OR N) ? |
| COMMAND | ANY | Y |
| ECHO | FULL | |
| ANSWER | STRING | |
| ANSWER | ANY | MOVE: |
| COMMAND | STRING | 1 |
| ECHO | FULL | |
| ANSWER | STRING | |
| ANSWER | ANY | ADDRN: |
| COMMAND | STRING | 0 |
| ECHO | FULL | |
| ANSWER | STRING | |
| ANSWER | ANY | PROG (ADDRS. FROM 1 TO 002735 ARE ILLEGAL): |
| COMMAND | STRING | 0 |
| ECHO | FULL | |
| ANSWER | STRING | |
| ANSWER | ANY | FILE NAME FOR ERROR MESSAGES: |
| COMMAND | STRING | TERM |
| ECHO | FULL | |
| MAX WAIT | SECONDS | 12 |
| ANSWER | STRING | |
| ANSWER | STRING | |
| ANSWER | STRING | ERROR MESSAGES WILL APPEAR AS 4 NUMBERS: |
| ANSWER | STRING | |
| ANSWER | ANY | PROG   FAILING ADD   EXP CONT   WRONG CONT |
| MAX WAIT | SECONDS | 5 |

```
ANSWER     STRING
ANSWER     STRING
ANS ERR    ERR EXIT    19
DELAY      MINUTES     10
COMMAND    ANY         <33B>
ECHO       NONE
ANSWER     STRING
ANSWER     STRING      USER BREAK AT      22
                                           #
ANSWER     ANY         @
COMMAND    STRING      LOG
ECHO       FULL+ LF
ANSWER     NO CHECK
ANSWER     STRING      --EXIT--
```

When the MOVER program is started, you can see that each command sent is followed by:

```
ECHO       FULL
ANSWER     STRING
```

and not by the usual ECHO FULL+ LF

This is due to the fact that MOVER echoes the CR given as the last command string character, and afterwards sends CR and LF. Usually, command processors also send CR/LF but without echo on the received CR. After the program is fully started, it waits for 10 minutes before breaking it with the ESCAPE code. During this 10 minute delay, Telefix checks that there is no unexpected answer in the communication line input buffer. If they are unexpected answers, which means the test program reports memory faults, the first 20 lines only will be printed, because error exit on answer error with 19 permitted errors is specified.

## Example 10. Example of CONSTANT use

In the previous example, the MOVER test program runs for 10 minutes. With the same script it is possible to run MOVER for different times corresponding to different descriptions by using a CONSTANT as the DELAY command parameter:.

```
DELAY    MINUTES     *=1
```

A description to run MOVER for one hour will include the following constant definition placed before the SCRIPT command:

```
CONSTANT     1       60
```

It is also possible to use an OPERATOR PROMPT to request the running time with the following description:

```
 CODE       SUBCODE     PARAMETER

COMMENT     DISPLAY     Automatic job to run MOVER test program
TIME OUT    PERCENT%    50
CH WAIT     5
ERR EXIT    ABORT
COMMENT     LOCAL       The time MOVER will run is defined below:
CONSTANT       1        30
CONSTANT       1        ?=Give number of minutes MOVER must run:
SCRIPT                  MOVER
COMMENT     DISPLAY     --- END OF MOVER   No error detected ---
```

If the operator does not answer to the prompt, MOVER will run for 30 minutes.

Example 11. Example of using a list of possible answers

```
  CODE       SUBCODE    PARAMETER

--------   --------
COMMAND    STRING     RUN
ECHO       FULL+LF
MAX WAIT   MINUTES    1
POS ANSW   STRING     Section 1 running
POS ANSW   STRING     Section 2 running
POS ANSW   STRING     Section 3 running
POS ANSW   STRING     Section 4 running
POS ANSW   STRING
POS ANSW   ANY        All sections (A) or only one (O) :
POS SEQU
COMMAND    ANY        A
ECHO       FULL
END POS    RETURN
POS ANSW   STRING     End of test
POS SEQU
END POS    CONTINUE
MAX WAIT   SECONDS    5
ANSWER     ANY        >
COMMAND    STRING     EXIT
ECHO       FULL+LF
--------   --------
```

## Example 12. Complex example to run MOVER test program

The following are the description and the scripts which allow you to start the MOVER test program in any case: Whether the remote system is log out or log in and with or without a program running, the accounting is turned on or not, or the SINTRAN prompt string is redefined or not.

LISTING FOR DESCRIPTION :

(SYSTEM-PACK:SCRIPT)MOVER-RUN:TDES;1

```
 CODE      SUBCODE       PARAMETER
 ----      -------       ---------

COMMENT   DISPLAY    MOVER test program
TIME OUT  PERCENT%   50
CH. WAIT             5
ERR EXIT  DISPLAY    ---- MOVER ABORTED, too many errors ----
ERR EXIT  SCRIPT     MOVER-ABORT
CONSTANT     1       FLOPPY-USER
CONSTANT     1       ?=Give user name (Default= FLOPPY-USER)
CONSTANT     2
CONSTANT     2       ?=Give user password (Default= none)
CONSTANT     3
CONSTANT     3       ?=Give project password (Default= none)
CONSTANT     4       30
CONSTANT     4       ?=Give number of minutes to run MOVER (Default= 30)
SCRIPT               MOVER-RUN
IF FALSE     1
COMMENT   DISPLAY    ---- END OF MOVER   No error detected ----
ABORT
END IF
COMMENT   DISPLAY    ---- END OF MOVER  Errors have occurred ----
ABORT
ENTRY        2
COMMENT   DISPLAY    ---- Wrong user name or password     ABORTED ----
ABORT
ENTRY        3
COMMENT   DISPLAY    ---- Wrong project password     ABORTED ----
ABORT
ENTRY        4
COMMENT   DISPLAY    ---- MOVER:PROG not available at remote system ----
SCRIPT               MOVER-ABORT
```

## LISTING FOR SCRIPT :

(SYSTEM-PACK:SCRIPT)MOVER-RUN:TSCR;1

```
 CODE     SUBCODE       PARAMETER
 ----     -------       ---------

TYPING    FULL
ECHO ERR  DO
ECHO ERR  ERR EXIT  0
END DO    CONTINUE
ANSW ERR  DO
ANSWER    NO CARE
COMMAND   ANY       [
                    &
ECHO      NONE
ANSWER    NO CARE
COMMAND   STRING    LOGOUT
ECHO      NO CHECK
ANSWER    NO CARE
ANSW ERR  ERR EXIT  1
END DO    CONTINUE
COMMAND   ANY       [
                    &
ECHO      NONE
MAX WAIT  SECONDS   5
ANSWER    NO CARE
COMMAND   STRING    *=1
ECHO      FULL+ LF
ANSWER    ANY       PASSWORD:
COMMAND   STRING    *=2
ECHO      NONE
ANSWER    STRING
POS ANSW  STRING
POS ANSW  STRING    OK
POS SEQU
END POS   CONTINUE
POS ANSW  ANY       PROJECT PASSWORD:
POS SEQU
COMMAND   STRING    *=3
ECHO      NONE
END POS   RETURN
POS ANSW  ANY       ENTER
POS SEQU
PROG. EX     2
END POS   CONTINUE
POS ANSW  STRING    PROJECT NAME:2
                                 #
POS ANSW  STRING    UNKNOWN PROJECT PASSWORD
POS SEQU
PROG. EX     3
END POS   CONTINUE
ANSWER    NO CHECK
ANSW ERR  DO
PROG. EX     4
```

```
END DO     CONTINUE
COMMAND    STRING     MOVER
ECHO       FULL+ LF
ANSWER     STRING
ANSWER     STRING
ANSWER     STRING     MOVING MEMORY TEST PROGRAM
ANSWER     STRING
ANSWER     STRING     HAR-1863C    APRIL 29, 1977
ANSWER     STRING
ANSWER     STRING     MAXIMAL MEMORY ADDRESS: 177777
ANSWER     STRING
ANSWER     ANY        DO YOU KNOW THIS PROGRAM (Y OR N) ?
COMMAND    ANY        Y
ECHO       FULL
ANSWER     STRING
ANSWER     ANY        MOVE:
COMMAND    STRING     0
ECHO       FULL
ANSWER     STRING
ANSWER     ANY        ADDRN:
COMMAND    STRING     0
ECHO       FULL
ANSWER     STRING
ANSWER     ANY        ADDRX:
COMMAND    STRING     177777
ECHO       FULL
ANSWER     STRING
ANSWER     ANY        PROGN (ADDRS. FROM 1 TO 002735 ARE ILLEGAL):
COMMAND    STRING     0
ECHO       FULL
ANSWER     STRING
ANSWER     ANY        FILE NAME FOR ERROR MESSAGES:
COMMAND    STRING     TERM
ECHO       FULL
ANSWER     STRING
ANSWER     STRING
ANSWER     STRING     ERROR MESSAGES WILL APPEAR AS 4 NUMBERS:
ANSWER     STRING
MAX WAIT   SECONDS    8
ANSWER     ANY        PROGN  FAILING ADDRESS EXPECTED CONTENTS WRONG CONTENTS
MAX WAIT   SECONDS    5
ANSWER     STRING
ANSWER     STRING
RESET FL       1
ANSW ERR   DO
SET FLAG       1
ANSW ERR   ERR EXIT   18
END DO     RETURN
DELAY      MINUTES    *=4
COMMAND    ANY        [
                      &
ECHO       NONE
ANSWER     STRING
ANSWER     STRING     USER BREAK AT     22
                                         #
ANSWER     NO CHECK
COMMAND    STRING     LOGOUT
```

```
ECHO      FULL+ LF
ANSWER    NO CARE
```

LISTING FOR SCRIPT :

(SYSTEM-PACK:SCRIPT)MOVER-ABORT:TS,CR;1

```
 CODE      SUBCODE      PARAMETER
 ----      -------      ---------

CLEAR
COMMAND    ANY         [
                       &
ECHO       NONE
ANSWER     NO CARE
COMMAND    STRING      LOGOUT
ECHO       FULL+ LF
ANSWER     NO CARE
```

# INDEX

## ************** SEND US YOUR COMMENTS!!! **************

Are you frustrated because of unclear information in this manual? Do you have trouble finding things?

Please let us know if you
- find errors
- cannot understand information
- cannot find information
- find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!

## ************ HELP YOURSELF BY HELPING US!! ************

Manual name: Manual number:

What problems do you have? (use extra pages if needed) ____________________

Do you have suggestions for improving this manual ? ____________________

Your name: ____________________ Date: __________

Company: ____________________ Position: __________

Address: ____________________

What are you using this manual for ? ____________________

**NOTE!**

This form is primarily for documentation errors. Software and system errors should be reported on Customer System Reports.

**Send to:**

Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

Norsk Data's answer will be found on reverse side

Answer from Norsk Data ______________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

Answered by________________________________________Date ______________

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

□

**Norsk Data A.S**  
Documentation Department  
P.O. Box 25, Bogerud  
0621 Oslo6, Norway