L1 MOS

System Software Generation and Installation User Guide

olivetti

L1 MOS

System Software Generation and Installation User Guide

olivetti

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis - 10015 IVREA (Italy)

PREFACE

This manual is a guide to the configuration and generation of L1 MOS software systems and the delivery and installation of systems for customers. It is intended for software specialists in Olivetti subsidiary companies and software houses.

SUMMARY

The manual is divided into seven parts.

Part I describes the approach to configuration. Basic concepts are defined and the choices of operating system and application environments are explained.

The configuration, generation and installation phases are briefly outlined.

Part II is dedicated to the operating system configuration parameters.

The theoretical aspects of configuring the operating system are discussed and all the parameters which are the means of realizing the desired operating system configuration are listed, with a description and the default value for each.

Part III describes the interactive utility which configures the COBOL environment, the SYSCONF utility which configures the operating system, and the process of generating systems for customers.

Part IV is a guide to the preparation of distribution kits for the delivery of customer systems and the subsequent installation (if necessary) of customer systems.

The application-level configuration process needed to prepare systems for the end user is also illustrated.

Part V is a reference section providing details of the configuration and installation commands used during the various phases.

Part VI describes the SYSCONFR procedure, which permits to configure, generate and save on the installation support operating systems with characteristics belonging to a given range.

Part VII gives information related to the memory allocation (Appendix A) and the correspondence between the user files and their internal mapping (Appendix B). Some examples of system configuration are listed in Appendix C.

An analytical index is present at the end of the manual.

REFERENCES

Read first ...

L1 MOS Release Guide 5.2

Introduction to MOS - Code 4002130 G (vol. 2)

CSS Generation and Configuration - User Guide - Code 4000750 H (vol. 7A)

For further information, read ...

MOS - SHELL Commands - Reference Manual - Code 4002770 Q (vol. 3)

System Software Maintenance - User Guide - Code 4002300 M (vol. 7B)

RS232/CL Interface - Programmer Guide - Code 4004450 H (vol. 6G)

MOS Structure and Functioning - Code 4002420 M (vol. 8)

MTS - Configuration Guide - Code 4003450 L (vol. 7A)

Terminal Emulators - Configuration Guide - Code 4000930 U (vol. 7A)

BEAM - User Guide - Code 4003180 R (vol. 6H)

MOS ESE Environment - Software Installation - User Guide
Code 3985500 P (vol. 7A)

ONE - Network Management Guide - Code 4008550 C (vol. 7A)

NMS - CMS Component - Operating Guide - Code 4000980 H (vol. 7C)

NMS - LMS Agent Component - Operating Guide
Code 4000960 X (vol. 7C)

FOURTH EDITION: September, 1984 - Release 4.0

FIFTH EDITION: February, 1985 - Releases 4.1 / 4.2

SIXTH EDITION: June, 1985 - Release 5.0

UPDATE: October, 1985 - Release 5.0

SEVENTH EDITION: March, 1986 - Release 5.1

EIGHTH EDITION: April, 1987 - Release 5.2

SUMMARY OF AMENDMENTS

This manual is a re-edition, incorporating the Release 5.1 Errata Corrige, the new items for Release 5.2 and editorial changes.

CHANGES

The following list gives the principal software changes in MOS release 5.2 that have been incorporated into this manual. The Chapters in which an item is described are indicated when appropriate.

1. Some new hardware items are now supported:

   a) ESDI 140 Mb (Chapters 3, 4 and 9)

   b) ST506 20 and 40 Mb (Chapter 9)

   c) SMD 275 Mb hard disks (Chapters 9)

   d) SCT5 Streaming Cartridge Tape (Chapter 6)

   e) M70 machine (Chapters 3 and 6)

   f) REDAC board (Chapter 6)

2. Some new commands are available (Chapter 11):

   a) CONF (to configure on a PC the L1WSE emulator program)

   b) CONFMAN (to display or modify existing configurations)

   c) MODCON (to display, print or modify existing $CON files)

   d) OLICONF (to configure on a PC the OLIEMU emulator program)

   e) REBUILD (to regenerate in batch the $CHM or the $IPL files, or the SYS volume, without needing to reconfigure when all the parameters are unchanged).

   f) WSECONF (to configure on a PC the WSELAN emulator program)

3. A new tool is available (SYSCONFR) which allow the configuration and generation of BC stand alone systems with predefined characteristics (Chapters 2 and 12). The SCRIPLBUILD utility is available which produces the $IPL modules required by SYSCONFR and places them in the directory where SYSCONFR expects them to be.

4. SYSCONF (Chapter 6):

   - can now produce the SYS volume (and no longer produces the $BST 1-module)

   - uses a different file system

   - places its output in an archive directory subsequently used as input for SYSGEN.

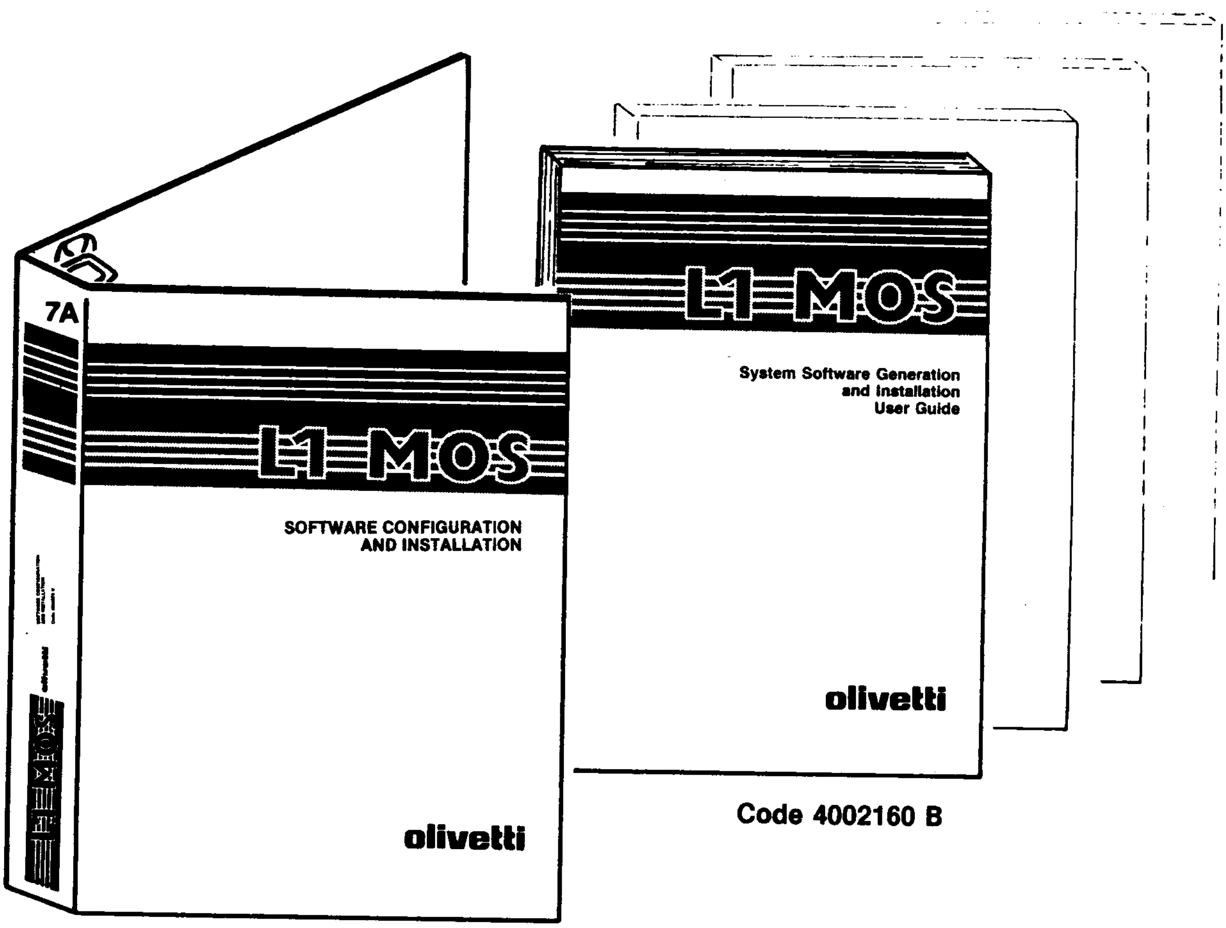
5. The SYSGEN procedure has been modified (Chapters 2 and 7)

6. The run-time COBOL structure has been modified (Chapter 5)

7. A new NOMLOG function can be used in the Grandpa configuration file, which allows system start up without operator login on the master work station.

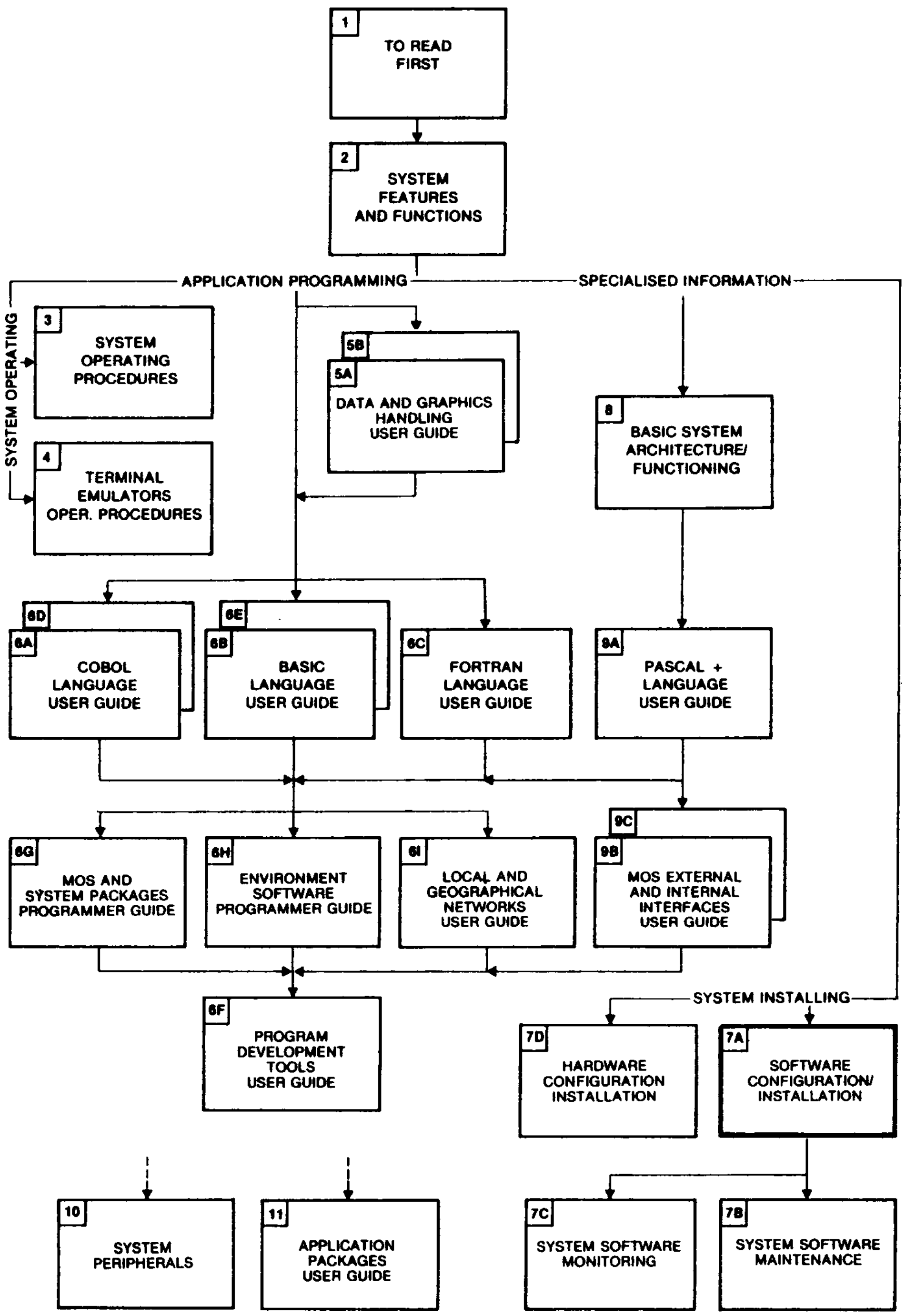
8. A new RTTY keyword is available for the Grandpa configuration file (Chapter 10).

9. Higher limits have been fixed for the Grandpa configuration file entries (Chapter 10).



Code 4002160 B

Code 4004870 U


1 TO READ FIRST
2 SYSTEM FEATURES AND FUNCTIONS
APPLICATION PROGRAMMING
SPECIALISED INFORMATION
SYSTEM OPERATING
3 SYSTEM OPERATING PROCEDURES
4 TERMINAL EMULATORS OPER. PROCEDURES
5B
5A DATA AND GRAPHICS HANDLING USER GUIDE
8 BASIC SYSTEM ARCHITECTURE/ FUNCTIONING
6D
6A COBOL LANGUAGE USER GUIDE
6E
6B BASIC LANGUAGE USER GUIDE
6C FORTRAN LANGUAGE USER GUIDE
9A PASCAL + LANGUAGE USER GUIDE
6G MOS AND SYSTEM PACKAGES PROGRAMMER GUIDE
6H ENVIRONMENT SOFTWARE PROGRAMMER GUIDE
6I LOCAL AND GEOGRAPHICAL NETWORKS USER GUIDE
9C
9B MOS EXTERNAL AND INTERNAL INTERFACES USER GUIDE
SYSTEM INSTALLING
6F PROGRAM DEVELOPMENT TOOLS USER GUIDE
7D HARDWARE CONFIGURATION INSTALLATION
7A SOFTWARE CONFIGURATION/ INSTALLATION
10 SYSTEM PERIPHERALS
11 APPLICATION PACKAGES USER GUIDE
7C SYSTEM SOFTWARE MONITORING
7B SYSTEM SOFTWARE MAINTENANCE

CONTENTS

PAGE

PAGE

CONFIGURATION STRATEGY

THE CONFIGURATION PARAMETERS

GUIDE TO THE GENERATION PROCESS

SYSTEM DELIVERY, INSTALLATION AND USE

CONFIGURATION AND INSTALLATION COMMANDS

SYSTEM GENERATION AND CONFIGURATION USING SYSCONFR

APPENDICES

1

PART I : CONFIGURATION STRATEGY

INTRODUCTION TO PART I

In Chapter 1 the general concepts fundamental to the process of configuring and generating L1 MOS software are presented.

In Chapter 2 a general description of the generation, configuration and installation process is given.

1. CONCEPTS

This Chapter presents the main concepts underlying the configuration of L1 software. The application areas supported are business computers (BC), scientific/technical (ST) and data processing (DP). The system software is configurable (by Olivetti subsidiary companies), in order to obtain:

- one or more application environments appropriate to the needs of business, scientific/technical or data processing applications

- an operating system suited to the chosen configuration.

The configuration of L1 MOS software assumes that all hardware configuration decisions have been made. The configuration process thus involves two choices:

1. First, one or more application environments appropriate to the application area and the use that the particular customer intends to make of the system, must be chosen.

2. Second, the parameters needed to configure the operating system so that it is able to manage the selected hardware and software must be specified.

L-MODULES AND UNITS

The basic unit of software is the l-module. All L1 MOS software is released as l-modules, which are executable code modules in a format that is directly loadable by the operating system (OS).

The OS l-modules may contain software units. For a complete list of all the OS l-modules and the units within them, see the Release Guide.

CONFIGURATION PARAMETERS

Any OS software unit may be configurable, that is, may contain parameters having a default value that can be modified when the system is configured. If a particular unit is configurable, all of its parameters must be assigned values that accurately reflect the required characteristics of the operating system.

The values assigned to the parameters of all the various units combine to configure the operating system and thus to determine the characteristics of the product for the customer.

A complete list of configuration parameters is given in Chapter 4. The generation of MOS systems on subsidiary company machines is described in Chapter 7.

FUNCTIONAL SETS

For the purposes of configuring and generating systems, L1 software has been grouped into functional sets (FS). These may be the system functional set, which contains the OS l-modules, or application functional sets, each of which contains a viable version of an application environment or facility.

The functional set is not a totally fixed entity as it may contain optional modules and subsets of alternative modules requiring choices to be made by the person generating the system.

The organisation of the software in functional sets is intended to fulfill various aims:

- To guarantee that the systems generated are both internally consistent and viable. It is not possible to generate an incomplete and therefore infeasible system, as the choice of a particular functional set automatically involves the inclusion in the system of all the non-optional (and thus necessary) l-modules for that facility.

- To avoid redundancy in the product. After the user has made his selections, the functional sets will contain only those modules which are needed for the operating system and the selected application environments respectively.

One system functional set is used to generate the operating system for a product. To generate application environments, one or more application FS must also be chosen.

THE GENERATION PROCESS

System generation is carried out on Olivetti subsidiary companies' premises, using a suitable machine known as the "mostro" and a "starter" system specific to the machine type. Details about the hardware configuration required by the "mostro" machine are given in the Release Guide.

The input needed for the generation phase is a distribution kit of floppy disks, magnetic tapes or Streaming Cartridge Tapes, containing all the available software in the release including the starter system.

A particular procedure (KIT_INSTALL) is provided in the starter system which installs the whole Release from the distribution kit to the hard disk of the mostro system.

Once the software has been installed, the following steps must be carried out:

1. The choice of functional sets for the customer.

2. The tailoring of the operating system to the customer's hardware installation.

3. The preparation of a kit to deliver to the customer.

The available application environments, operating system facilities, and functional sets contained in this current Release are described in the Release Guide. The user can see from these descriptions which functional sets to select for any particular system.

2. INTRODUCTION TO THE CONFIGURATION, GENERATION AND INSTALLATION PROCESS

The configuration and generation of L1 MOS software systems is carried out by subsidiary companies on a machine, known as the "mostro", configured appropriately as described in the Release Guide. A specialized "starter" system is used when installing software releases on the mostro.

The input for the configuration and generation process is a distribution kit of floppy disks or tapes containing all the software files available in a particular release, including the starter operating system.

The output of the configuration and generation process is an individually configured software system for a particular customer, ready to be installed. The distribution medium at this point can be floppy or mini-floppy disk, streaming cartridge tape, or magnetic tape, depending on the customer's hardware installation.

Various activities take place between the software release date and the first startup of a customer's system. A summary of these is given below and a flowchart is given in Fig. 2-2.

The release is effected when the subsidiaries receive software on magnetic storage (the release distribution kit) comprising the "Release Library" and the software needed to install it on their own machine, that is, a starter system for installation purposes.

The **Release Library** contains all the software modules in the release in executable form, subdivided into "system components" and "dependent components".

The **starter** system makes possible:

- the installation of the starter operating system on the subsidiary's machine

- the installation of the Release Library on the subsidiary's machine.

The starter system contains the starter operating system l-modules, some dependent components and the KIT_INSTALL procedure which makes possible the automatic installation of the whole Release library on the subsidiary's machine.

A characteristic of the subsidiary's mostro machine is the simultaneous availability of a variety of magnetic support types. Details are given in the Release Guide.

In order to generate, on the mostro, a configured product for a customer:

1. activate the mostro machine with the starter system

2. install the Release library using the KIT_INSTALL procedure (as described in the Release Guide)

3. reset the mostro machine.

The configured product consists of the "FREE" volume, which is an image of the customer system. This volume comprises the system volume "SYS", containing the l-modules of the operating system, and the dependent components volume "DPC", containing the components which are "dependent" on the operating system , that is, the environments and programming languages selected for inclusion in the customer's system.

To generate the system volume SYS (using the SYSGEN procedure), the user is asked to select alternative or optional l-modules appropriate to the hardware and software customer machine configuration.

To generate the dependent components volume DPC, the user is asked to select one or more of the functional sets stored in the Release Library.

The SYS and DPC volumes are dumped onto an appropriate magnetic support and delivered to the customer to be installed.

GENERATION TOOLS

The following utilities may be used to generate every product for customers on the subsidiary's mostro machine:

MAKECRTS : for the generation of the required COBOL run time support.

SYSCONF : for the compilation of the system configuration file $CON and, optionally, for the generation of the l-modules $IPL and the SYS volume.

LINECONF : for the compilation of the CSS units' parameters and configuration tables, of the routing table and, optionally, for the generation of the l-module $CHM (see <u>CSS Generation and Configuration, User Guide).</u>

SYSGEN : for the generation of the system volume SYS.

DPCGEN : for the generation of the dependent components volume DPC.

SYSCONFR : for the configuration, the generation and the saving of only a limited range of systems.

CONFMAN : for the modification of existing system configurations.

SYSCONFR is an <u>alternative</u> to the sequence of utilities SYSCONF, SYSGEN, DPCGEN, and steps described in Chapters 8, 9, and 10, performing all these activities with little user-interaction. Ascertain that SYSCONFR is appropriate by reference to "menu three" in Part VI and, if it is, omit intervening Chapters.

The Shell commands needed during the preparation of customer systems and their installation are listed in Tab. 2-1, which also shows where each command is documented.

COMMAND	FUNCTION	MANUAL/CROSS REFERENCE
CLEARDIR	Clears directories	SHELL Commands Reference Manual
CONF	Configures the L1WSE emulator program	Part V, current manual
CONFMAN	Displays or modifies existing configurations	Part V, current manual
COPY	Copies files, volumes or directories	SHELL Commands Reference Manual
FILETAR	Saves and restores volumes on magnetic tape	SHELL Commands Reference Manual
FLD	Saves and restores volumes on FD	SHELL Commands Reference Manual
HEXED	Displays or modifies files containing hexadecimal characters	System Software Maintenance User Guide
INSTALLQM	Installs the queue manager	Part V, current manual
KEYCONF	Creates or modifies a keyboard table	Part V, current manual
LINKDEV	Links devices to form a single logical device	Part V, current manual
MKBOOT	Copies a bootstrapper	Part V, current manual
MKDIR	Creates new directories	SHELL Commands Reference Manual
MKENV	Prepares track 0 of a system floppy disk	SHELL Commands Reference Manual
MKLOGIN	Activates login mechanism for a new system	Part V, current manual
MKSYS	Prepares an SCT or an MTU and copies onto it the SYS directory of a configured system	Part V, current manual

Tab. 2-1 Shell Commands for System Preparation (cont.)

COMMAND	FUNCTION	MANUAL/CROSS REFERENCE
MKVOL	Creates a new volume on disk	SHELL Commands Reference Manual
MNT	Mounts a removable volume	SHELL Commands Reference Manual
MODCON	Modifies, displays or prints the $CON file	Part V, current manual
OLICONF	Configures the OLIEMU emulator program	Part V, current manual
PARSER	Converts source to object file for user environment configuration	Part V, current manual
REBUILD	Regenerates the $CHM or the $IPL file, or the SYS volume	Part V, current manual
REMOVE	Removes files, volumes and directories	SHELL Commands Reference Manual
RENAME	Renames files, volumes and directories	SHELL Commands Reference Manual
SCRIPLBUILD	Creates the $IPL files required by SYSCONFR	Part VI, current manual
SETWDIR	Resets the current working directory	SHELL Commands Reference Manual
SHDIR	Displays or writes to file the contents of directories	SHELL Commands Reference Manual
SPCONF	Defines and installs the spooling system environment	Part V, current manual
UNMNT	Unmounts a removable volume	SHELL Commands Reference Manual
VOLSR	Saves and restores volumes on SCT	SHELL Commands Reference Manual
WSECONF	Configures the WSELAN emulator program	Part V, current manual

Tab. 2-1 Shell Commands for System Preparation

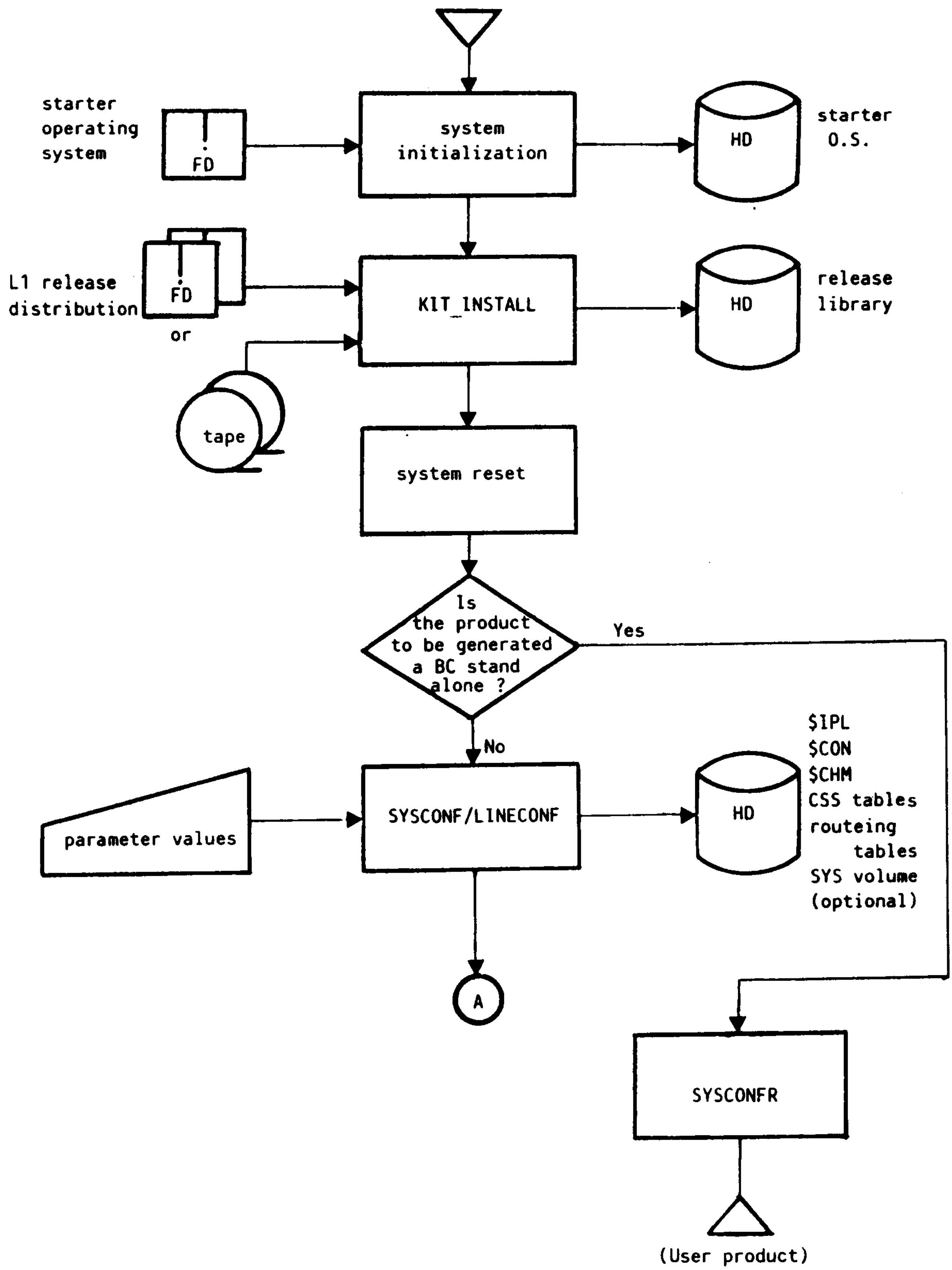


Fig. 2-2 Flowchart of System Preparation on the Mostro Machine (cont.)

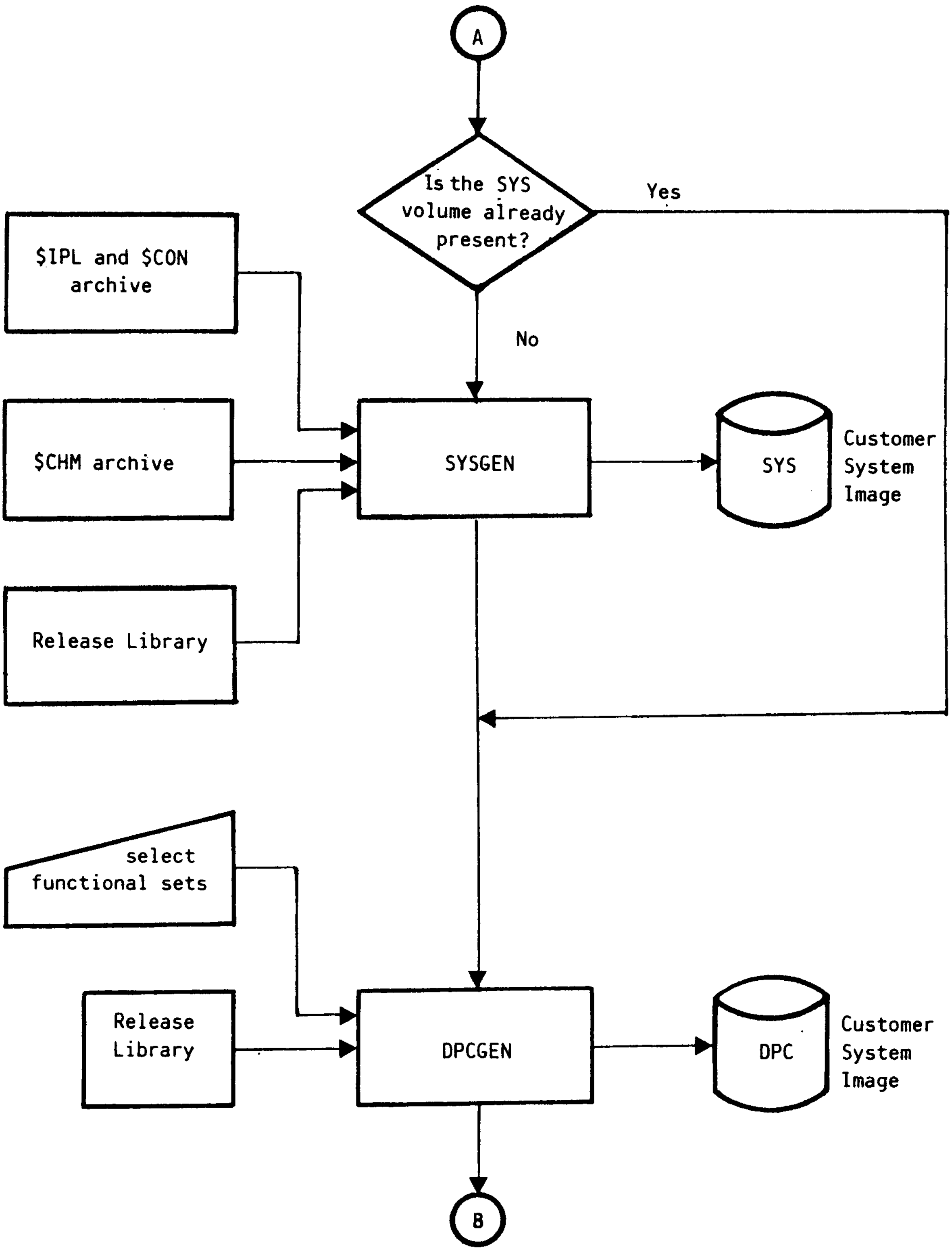


Fig. 2-2 Flowchart of System Preparation on the Mostro Machine (cont.)

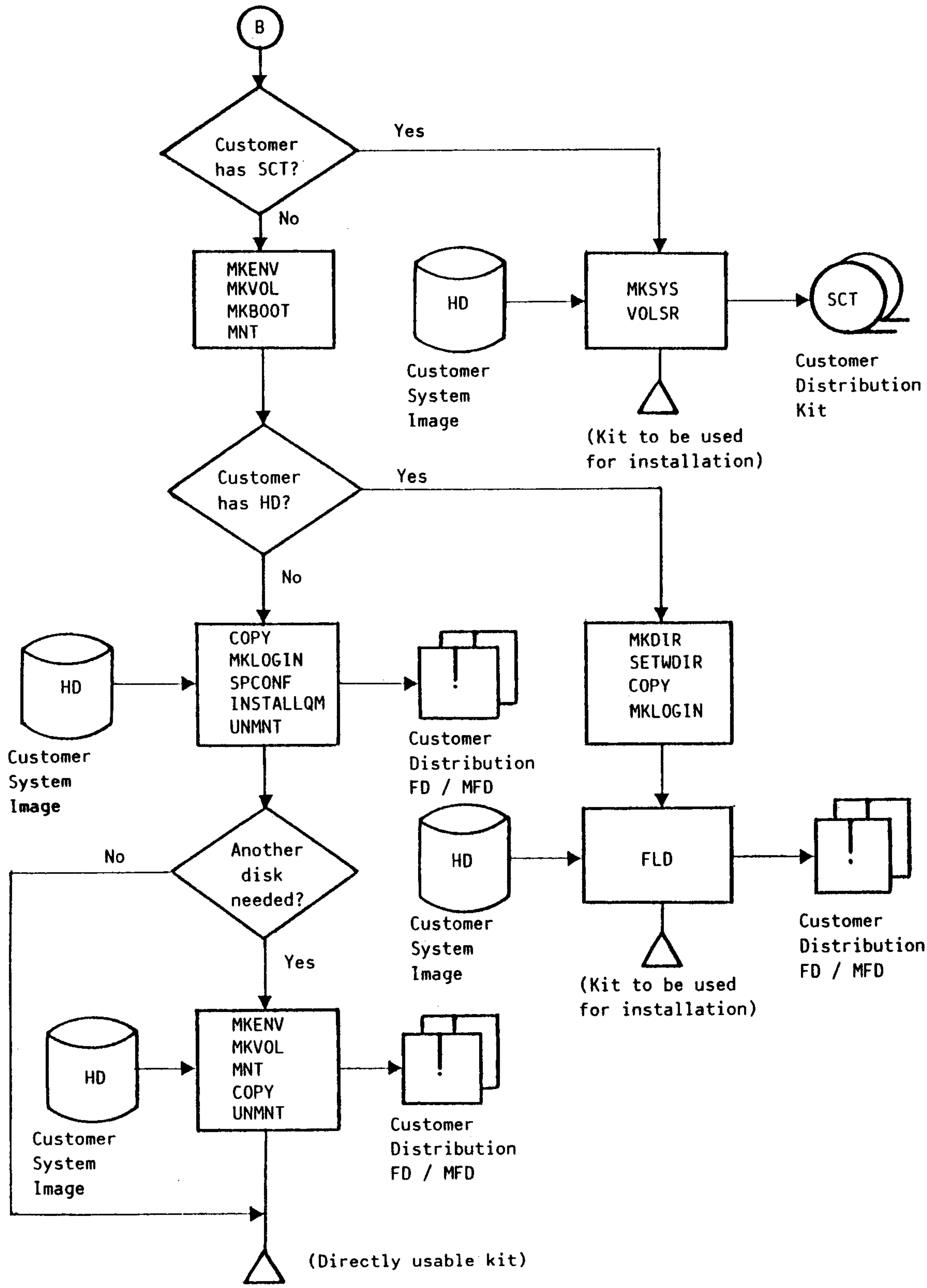


Fig. 2-2 Flowchart of System Preparation on the Mostro Machine

2

PART II : THE CONFIGURATION PARAMETERS

INTRODUCTION TO PART II

Chapter 3 is a guide to configure the required operating system.

In Chapter 4 the configuration parameters for the operating system are listed, together with their default value and a description for each.

3. CONFIGURATION DECISIONS

This chapter provides details on the configuration program requests (SYSCONF, see Chapter 6). The topics discussed cover:

- static data relating to hardware considerations (magnetic storage units, printers, communication lines, ...)

- dynamic data relating to software environments or user programs (families, processes, files, buffers) which influence directly the system tuning and performances.

In all these cases, the help given by SYSCONF (or the explanation of the single parameter in a Unit) may be insufficient because:

- the relation between the single parameter with other parameters, belonging to different screens, cannot be clearly expressed

- a parameter can be linked to several different parameters and its general description may not clearly emphasize the answers to be given in individual cases

- a parameter can be conditioned by the application's characteristics, the hardware constraints or the presence of software environment, from which information must be acquired beforehand.

The topics are handled so as to provide guidelines so that:

- the case to be handled can be defined

- the relevant data can be collected

- the data to be given to SYSCONF can be found using simple relations or reference tables provided directly in the Chapter.

HARDWARE CONSIDERATIONS

The operating system must be compatible with the hardware configuration of the L1 system.

The hardware installation and the software configuration are normally carried out by different people at different times in different places. For this reason it is necessary to provide tools which allow the various people involved in the installation/configuration process to work with the correct information.

SYSCONF is aimed at providing such a tool. SYSCONF requests specific and exact information on the hardware configuration related to:

- the type of system and its name
- the magnetic storage units
- the objects processed by the WSM such as:
  - peripherals
  - lines
  - channels.

In order to correctly reply to these questions it is necessary to have a knowledge of:

- the various peripheral controllers which exist and their position within the system board cage
- the types of peripherals connected
- the characteristics of the physical connections with the peripherals
- whether EAROM is present or not.

Most of the data requested are directly determined by the elements present in the system (for example, the type of peripherals present). The remaining data are determined by the installation standards (for example, the speed with which data is transferred to a printer). A summary table similar to that given in the section "Summary of Information Requested" is an aid to defining the various hardware elements installed in the respective L1 machine. A document of this type may be useful when attempting to solve anomalies caused by Hardware or Software configuration errors. Such errors may be caused by elements badly configured or by incongruous parameters.

MAGNETIC STORAGE UNITS

The type of disk units and the number of each type connected to the controller is requested. A disk unit may be:

- a 1M floppy disk (FDU) (8")
- a 320K floppy disk (MFDU) (5 1/4")
- a 1M floppy disk (MFDU) (5 1/4")
- a hard disk unit (HDU).

An HDU can either be connected directly to the controller (specifically defined as an integrated controller) or by means of one of the following interfaces:

SASI (14M HDU, only on M3X)

SMD (60M HDU or 120M HDU only on M4X, M6X and M70)

ST 506 (10M HDU, only on M3X; 20M HDU, only on M54; 27M HDU on M3X and M4X; 40M HDU, only on M54; 65M HDU on M3X, M4X, M6X and M70)

ESDI (140M HDU only on M54, M64 and M70).

To an integrated controller only an 18M HDU can be connected.

The SASI, SMD, ST 506 and ESDI interfaces correspond to Units 6, 20, 15 and 7. The integrated controller corresponds to Unit 21.

The user must specify how many units are connected to each controller in the system.

Tape units (STC and MTU) can be connected in addition to disk units; these peripherals have no configuration parameters.

CONFIGURABLE OBJECTS HANDLED BY THE WSM

The objects that are handled by the WSM and which are configurable using the Unit 5 are:

- work stations and their peripherals
- system printers
- RS232 software driver instances
- channels
- lines
- keyboard tables

In this section the above objects are examined in detail and the list of parameters required for them is described briefly.

There is a table at the end of the section giving the various peripheral connections.

The figure that follows gives a general picture of the peripherals, connections and controllers used by the L1 MOS systems.

As can be seen in the figure, the following kinds of controllers and ELB_like devices can be used (for peripherals other than disk units):

- KDC: to which a video-keyboard, an ELB or a CAT 3300 can be connected
- MUX 3388: supplied with four RS232 lines or current loop
- SIC 3332/3367 (Twin controller): supplied with maximum two RS232 channels or current loop
- CPU board: supplied with one RS232 line
- ELB 1381/1382: the latter is supplied with maximum two RS232 channels, in addition to those for the screen/keyboard, Badge and PIN Pad. The former can only be connected to a screen-keyboard.
- ELB 3683/3684 (also known as WSL1): supplied with maximum two RS232 channels in addition to those for screen/keyboard, Badge and PIN Pad; it can only be connected to a MUX.

The figure also shows the difference between channels and lines.

A channel connection terminates at a peripheral (for example, channels are supplied with ELB 1381/1382 or Twin boards).

A line connection terminates either at a peripheral or ELB-like device.

At present the only lines available in the system are one output line from the KDC and four output lines from the MUX. The WSL1 work station can be connected to the latter, in addition to the various peripherals.

This distinction is essential during configuration because the line and channel identifiers occur in each object to be configured.

Note: From now on, the term **PC** will refer to the Olivetti Personal Computers M24, M24SP, M19 and M28.

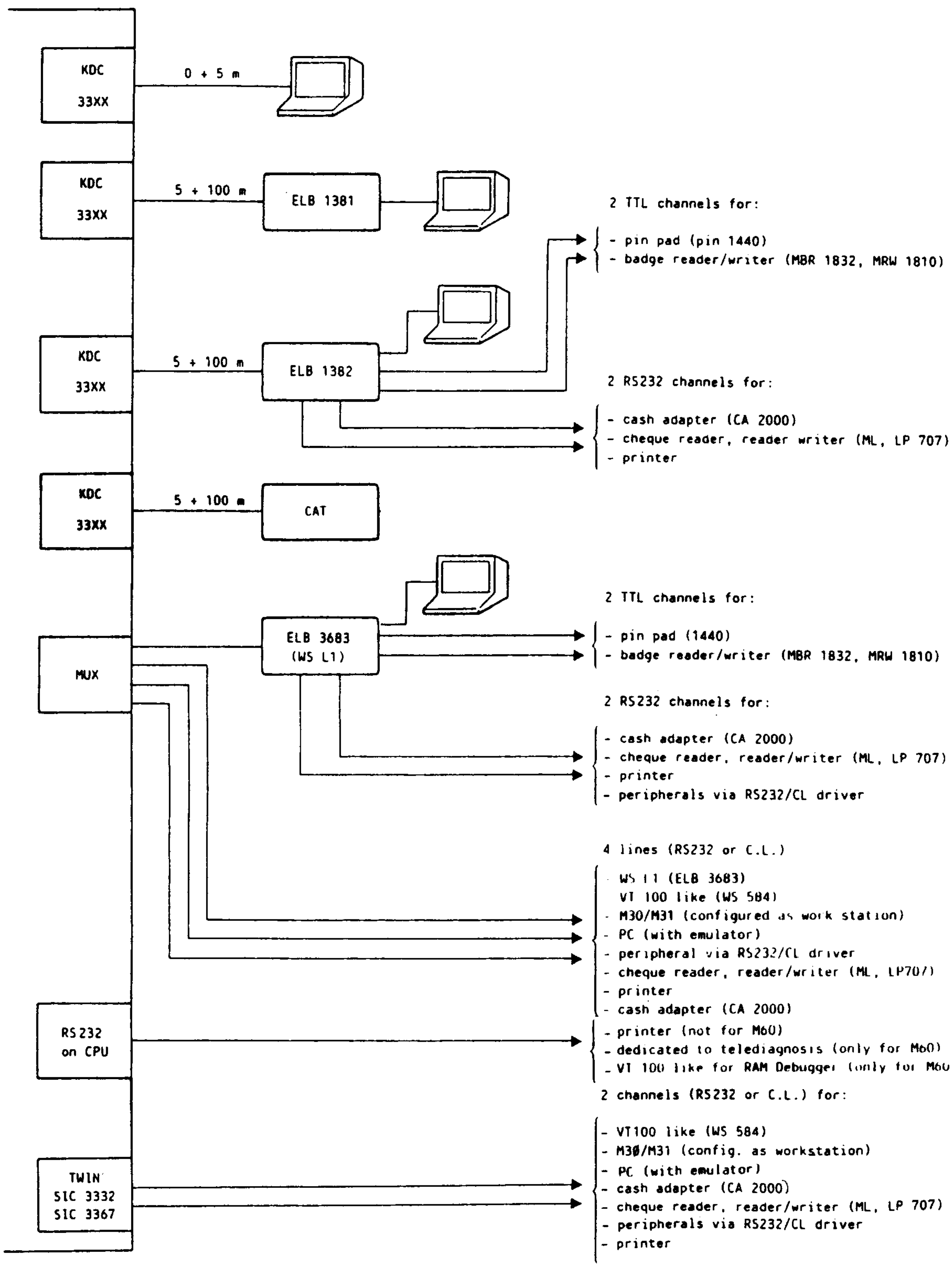


Fig. 3-1 Diagram of the Peripheral-Controller Connection

Work Station and their Peripherals

This is a logical set of several peripherals connected to the system in various ways. A work station for example can consist of a screen-keyboard, a Badge Reader and a printer connected via an ELB 1382, or a screen-keyboard connected via an ELB 1381 and a printer connected via MUX.

During configuration the user must declare how many work stations are connected and give some information as to the peripherals comprising each one.

These peripherals are as follows:

- Screen-keyboards
- Printers
- PIN Pads
- Badge Readers, Readers/Writers
- Cheque Readers, Readers/Writers
- Cash adapters
- CATs (Customer Activated Terminals)

The connections for each peripheral follow.

Screen-keyboard

Possible types are:

- integrated, that is, connected directly from the KDC or via ELB 1381/1382
- WSL1, that is, connected from MUX
- M30, M31, PC, VT100-like, that is, connected via RS232 from MUX or Twin
- work station connected via Port protocol (that is, PC in OLILAN).

Printer

These can be connected via RS232 from ELB 1382, MUX, WSL1, Twin or CPU (the latter is not possible on the M60).

Note: A modem may not be used to connect printers via MUX.

PIN Pad and Badge Reader, Reader/Writer

These can be connected via RS232 from ELB 1382, WSL1.

Cheque Reader, Reader/Writer and Cash Adapter

These can be connected via RS232 from ELB 1382, WSL1, MUX or Twin.

In configuration the user must declare the total number of work stations to be used (maximum 32), and then configure the peripherals for each one of them.

The user must provide the following information with which to configure each peripheral:

- If it is present or not: all the peripherals listed above are proposed to the user. This information ensures that only those that are present will be configured. If a peripheral is absent, the corresponding parameters are non-significant.

- A key that univocally selects the specific peripheral by describing the connection mode. This key comprises the following four characters:

  . One character that identifies the controller to which it is connected (for example, MUX, KDC, ...)

  . One character that identifies the channel (for controllers with several output channels, such as ELB, WSL1 or Twin)

  . One character that identifies the relative position of the controller in the slots within the controllers of the same type

  . One character that identifies the output line (only variable for MUX, which is the only one with several output lines)

- Some information which depends on the peripheral

System Printers

There is no physical difference between system and work station printers.

A printer connected via the RS232 Twin, for example, can either be configured as a system printer or as any work station printer. The difference is in their software handling. The former is managed by means of the File System Management and the latter by means of the logical driver of Work station Management.

The maximum number of system printers that can be configured is eight. In configuration the user must specify the total number of system printers present and the following information for each one:

- A four character key that univocally selects the printer (its composition is described under the work station peripherals)

- The connection mode (if it is current loop or not) and the communication protocol

CAT

A Customer Activated Terminal consists of a screen-keyboard, a work station printer and an RS232/CL driver instance connected via an ELB 1382 to a KDC controller.

RS232 Driver

There can be several RS232 channels in the system. These can be connected to the following controllers (see the figure at the beginning of the section):

- CPU board

- ELB 1382

- MUX

- WSL1

- SIC 3332 (only for M30/M40) and SIC 3367 (for M30/M40/M60 monoprocessor)

Either of the following can be connected to an RS232 channel:

- A peripheral controlled by the specific drivers via the WSM interface (Cash Adapter, Cheque Reader, ...)

- A peripheral controlled via the RS232 driver

In configuration, the user must decide the number of RS232 software driver instances present in the system. Each instance is dedicated to handling a channel (and thus a peripheral). When the number of RS232 driver instances is calculated, the RS232 peripherals handled by specific

WSM driver must not be included. When an instance of the RS232 software driver is to be configured, the user gives the key for the corresponding channel, which is made up as follows:

- One character identifying the channel controller
- One character identifying the channel (variable for controller with several output channels, such as MUX or Twin or ELB)
- One character that identifies the relative position of the controller (in the slots) within the controllers of the same type
- One character identifying the line (variable for channels in output from WSL1, connected to MUX via the line)

The key the user defines when an instance of the RS232 driver is configured is exactly the same as one of the keys defined when configuring the channels (see further on).

Channels

These are terminal objects to which peripherals are connected.

A configured peripheral corresponds to each channel and, vice versa, a channel corresponds to each peripheral. The number of channels is exactly the same as the number of peripherals in the system including the system printers and the peripherals connected via the RS232 Driver (excluding magnetic storages). A channel can sometimes operate as a line; for details of this see the section on lines.

At configuration time the user must specify the total number of channels in the system (theoretical maximum 256) and, for each channel:

- the same key defined when the peripherals, work stations, system printers or RS232 peripherals are configured and which is made up as follows:
  - . One character that identifies the controller that supplies the channel
  - . One character that identifies the channel (variable for controllers supplied with several output channels, such as Twin or ELB)
  - . One character that identifies the relative position of the controller (in the slots) within the controllers of the same type
  - . One character that identifies the line (variable for output channels from WSL1, connected to the MUX via a line)
- the physical/logical characteristics of the channel (reception/transmission speed, parity type, stop bits, buffer dimensions, ...).

Lines

These are objects that are not necessarily terminal, and to which both peripherals and other ELB-like devices can be connected.

The only lines available at present are output lines from MUX or KDC. MUX has four output lines to which peripherals or WSL1s can be connected. KDC has only one output line for integrated screen-keyboards or ELB.

When a terminal peripheral is connected to a line (for example, VT100 connected to MUX or a screen-keyboard on KDC) the latter also works as a channel and consequently must be included when both the channels and the lines are calculated. When a non-terminal ELB-like device is connected to a line (for example, WSL1 to MUX or ELB to KDC), the line does not work as a channel, in this specific case the channels are those in output from the device.

The user must define the total number of lines present in the system at configuration time (maximum 32) and the following for each line:

- A key that univocally identifies it (as for the other configurable objects), which is made up of four characters as follows:

  . The identifier of the controller supplying the line (MUX or KDC)

  . Identifier of one of the channels supplied by the device connected to the line

  . One character that identifies the relative position of the controller (in the slots) within the controllers of the same type

  . The line identifier (only variable for MUX)

- A value that specifies the speed of data transfer on line

Note

The same key can usually be given when different objects are defined.

If the following configuration is taken into consideration:

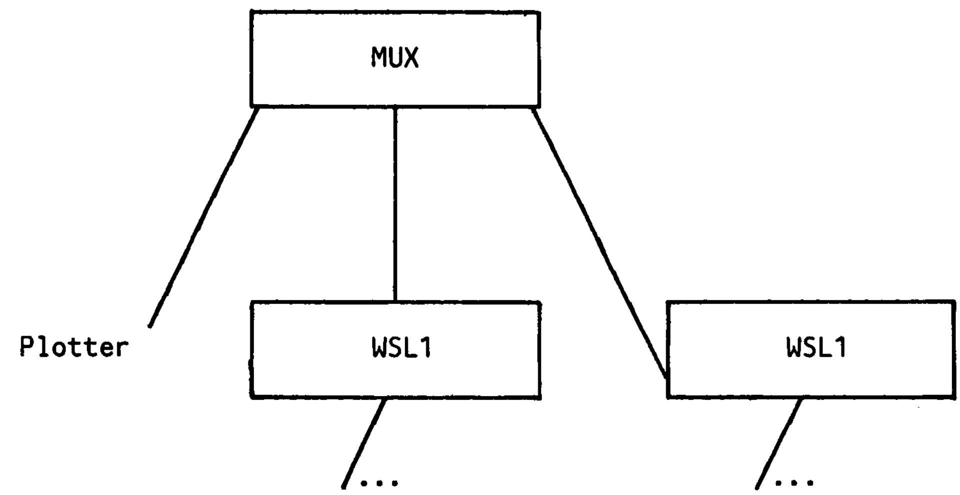


Fig. 3-2 Example of Configuration with Key used Repeatedly

the same key that is defined for the RS232 channel that connects the plotter to MUX must be specified in configuration of:

- lines
- channels
- RS232 drivers.

The following figure shows the various peripheral/controller connections.

Controller / Peripheral	SIC 3332/3367			MUX 3388			ELB 1382		WSL 1 (ELB 3683)		RS CPU
	RS232	MODEM	20 mA C. L.	RS232	MODEM	20 mA C. L.	1° RS232	2° RS232	1° RS232	2° RS232	
WSL1 (ELB 3683)	NO	NO	NO	YES	YES	YES ± 20mA	/	/	/	/	/
VT100 like (WS 584)	YES	YES	NO	YES	YES	YES	/	/	/	/	/
M30, M31 (WSG 3622 WSG 3623)	YES	NO	YES	YES	NO	YES	/	/	/	/	/
M 24	YES	YES	NO	YES	YES	YES	/	/	/	/	/
Cash adapter CA 2000	YES	/	/	YES	/	/	YES	YES	YES	YES	NO
Cheque reader reader/printer (ML, LP 707)	YES	/	/	NO	/	/	YES	YES	YES	YES	NO
PERIPHERALS VIA DRIVER "RS 232/CL"	YES	YES	YES	YES	YES	YES	NO	NO	YES	YES	NO
PRINTERS	*	NO	*	*	NO	*	*	*	*	*	*
PIN Pad (pin 1440) Badge reader/writer (MBR 1932 MRW 1810)	/	/	/	/	/	/	YES INTERFACE TTL (NO RS)		YES INTERFACE TTL (NO RS)		/
TELEDIAGNOSIS	/	/	/	/	/	/	/	/	/	/	(**)

Fig. 3-3 Table of Peripherals/Controller Connections

(*) See the next table
(**) Only available on the M60

The symbol "/" shows that the two hardware devices are not consistent with one another or their configuration is not advisable.

The figure below shows the various printer/RS232 controller connections for each type of printer.

PRINTER	RS FROM CPU (1)	RS FROM SIC	RS FROM ELB 1382	RS FROM ELB 3683/84	RS FROM MUX 3388
General Purpose					
PR 1450	YES	YES	YES	NO	NO
PR 1470	YES	YES(2)	YES	YES	YES(2)
PR 1480	YES	YES	YES	YES	YES
PR 1490	YES	YES(2)	YES	YES	YES
PR 15	YES	YES	YES	YES	NO
PR 17	YES	YES	YES	YES	NO
PR 19	YES	YES(2)	YES	YES	YES(2)
DM 280	YES	YES	YES	YES	YES
DM 290	YES	YES	YES	YES	YES
PR 3300	NO	YES	NO	NO	YES
PR 3600	NO	NO	NO	NO	YES
Specialized					
PR 2835	YES	YES	YES	YES	NO
PR 2880	YES	YES	YES	YES	NO
PR 2890	YES	YES	YES	YES	NO
PR 2840	YES	YES	YES	YES	NO
PR 2845	YES	YES	YES	YES	NO
PR 2850	YES	YES	YES	YES	NO
PR 40	YES	YES	YES	YES	NO
Letter Quality					
PR 340	YES	YES(2)	YES	YES	YES(2)
DY 450	YES	YES	YES	YES	YES
Non Impact					
PR 2400	YES	YES	YES	NO	NO
Multifunctional					
PR 38	YES	YES	YES	YES	NO
PR 1580	NO	YES(2)	YES	YES	YES(2)

Tab. 3-4 Table of Printer/Channel Connections

Note (1): On the M60, the RS232 of the CPU is dedicated to telediagnosis, consequently no printer can be connected to it. This column is only valid for other L1 systems.
Note (2): The mode 20 mA Current Loop is also available.

SUMMARY OF INFORMATION REQUESTED

The position of the various types of controller boards within the slots of the board cage is not of particular significance. It is only important to know the position of boards of one specific type with respect to the position of the other boards of the same type.

The first board of each type is the one closest to the first slot in the board cage.

The first slot, which in L1 systems is normally occupied by the CPU, is the first to the left when facing the back of the machine.

The following table is a summary of information related to the objects handled by the WSM.

KEYS SUMMARY

SLOT		Controller Type M/E/T/R	Line	Channel	Device	KEY: controller type + channel + slot (relat.) + line	Notes
Absolute Position	Relative Position						
			0	A			
				B			
				C			
				D			
				E			
				T			
			1	A			
				B			
				C			
				D			
				E			
				T			
			2	A			
				B			
				C			
				D			
				E			
				T			
			3	A			
				B			
				C			
				D			
				E			
				T			
			0	A			
				B			
				C			
				D			
				E			
				T			
			1	A			
				B			
				C			
				D			
				E			
				T			
			2	A			
				B			
				C			
				D			
				E			
				T			
			3	A			
				B			
				C			
				D			
				E			
				T			

WORK STATION DEVICE INFORMATIONS															
Screen/keyboard				Printer						MICR		CADP			
Key	Type	Conn. feat.	Hard. C. Key	Key	Curr. Loop Y/N	Free runn. Y/N	Hard Copy device	Printer shared: WSID	Ligth	Key	Type	Key	Type	Cadp shared: WSID

RS232 DRIVER INFORMATIONS			
KEY	DSR	CTS	DCD

LINE INFORMATIONS	
KEY	BAUD RATE

SYSTEM PRINTER INFORMATIONS		
KEY	Curr. loop Y/N	Free Runn. Y/N

CHANNEL INFORMATIONS											
KEY	STOP BITS	PARITY	BITS CHAR	BAUD RATE	XON/ XOFF	MARK/ SPACE	OTHER	DTR	RTS	KB TABLE ID	WS TYPE

The following example shows how to use the table for this hardware configuration.

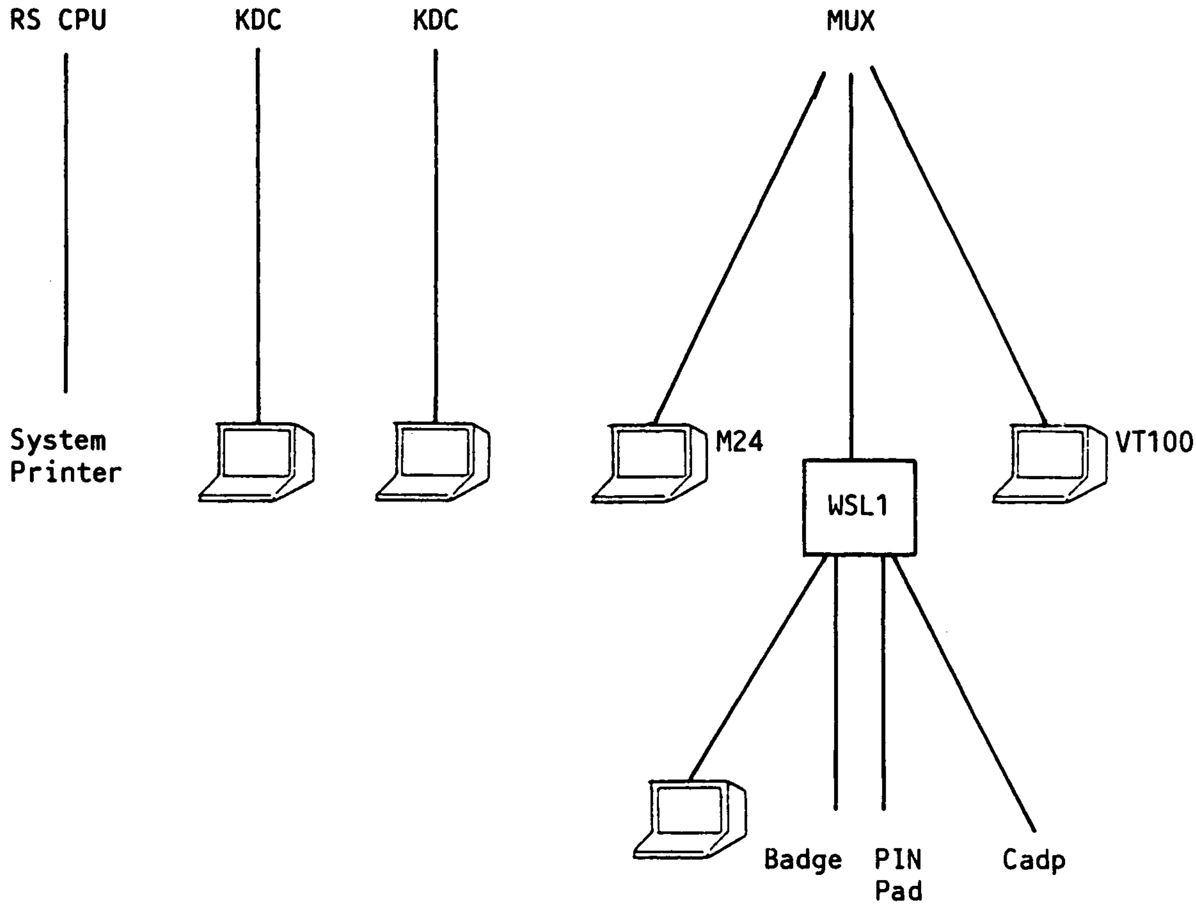


Fig. 3-5 Example of Hardware Configuration (cont.)

KEYS SUMMARY

SLOT		Controller Type M/E/T/R	Line	Channel	Device	KEY: controller type + channel + slot (relat.) + line	Notes
Absolute Position	Relative Position						
1	A	R	0	A	System Printer	RAAØ	
				B			
				C			
				D			
				E			
				T			
			1	A			
				B			
				C			
				D			
				E			
				T			
			2	A			
				B			
				C			
				D			
				E			
				T			
			3	A			
				B			
				C			
				D			
				E			
				T			
2	A	E	0	A			
				B			
				C			
				D			
				E	Screen/Keyboard	EEAØ	
				T			
			1	A			
				B			
				C			
				D			
				E			
				T			
			2	A			
				B			
				C			
				D			
				E			
				T			
			3	A			
				B			
				C			
				D			
				E			
				T			

Fig. 3-5 Example of Hardware Configuration (cont.)

KEYS SUMMARY

SLOT Absolute Position	SLOT Relative Position	Controller Type M/E/T/R	Line	Channel	Device	KEY: controller type + channel + slot (relat.) + line	Notes
3	B	E	0	A			
				B			
				C			
				D			
				E	Screen/Keyboard	EEBΦ	
				T			
			1	A			
				B			
				C			
				D			
				E			
				T			
			2	A			
				B			
				C			
				D			
				E			
				T			
			3	A			
				B			
				C			
				D			
				E			
				T			
4	A	M	0	A			
				B			
				C			
				D			
				E			
				T	M24	MTAΦ	
		M	1	A	Cash Adapter	KAA1	
				B			
				C	Badge	MCA1	
				D	PIN Pad	MDA1	
				E	Screen/Keyboard	MEA1	
				T			
		M	2	A			
				B			
				C			
				D			
				E			
				T	VT100	MTA2	
		M	3	A			
				B			
				C			
				D			
				E			
				T			

Fig. 3-5 Example of Hardware Configuration (cont.)

WORK STATION DEVICE INFORMATIONS														
Screen/keyboard				Printer						MICR		CADP		
Key	Type	Conn. feat.	Hard. C. Key	Key	Curr. Loop Y/N	Free runn. Y/N	Hard Copy device	Printer shared: WSID	Ligth	Key	Type	Key	Type	Cadp shared: WSID
EEAØ	Olivetti	No New Line	NO									MAA1	Normal	NO
EEBØ	"	"	"											
MEA1	"	"	"											
MTAØ	"	"	"											
MTA2	No Olivetti	Auto New Line	"											

RS232 DRIVER INFORMATIONS			
KEY	DSR	CTS	DCD
MAA1	ØØ	Ø1	ØØØ1

LINE INFORMATIONS	
KEY	BAUD RATE
EEAØ	48ØØ
EEBØ	48ØØ
MEA1	96ØØ
MTAØ	48ØØ
MTA2	48ØØ

SYSTEM PRINTER INFORMATIONS		
KEY	Curr. loop Y/N	Free Runn. Y/N
RAAØ	N	Y

CHANNEL INFORMATIONS											
KEY	STOP BITS	PARITY	BITS CHAR	BAUD RATE	XON/ XOFF	MARK/ SPACE	OTHER	DTR	RTS	KB TABLE ID	WS TYPE
EEAØ	1	even	7	48ØØ	XON/XOFF Receive	MARK	Ø	1	1	ITALY_DP	alpha B/W
EEBØ	"	"	"	"	"	"	"	"	"	"	alpha colour
MTAØ	"	no parity	8	"	"	"	"	"	"	—	alpha B/W
MEA1	"	even	7	96ØØ	"	"	"	"	"	—	"
MCA1	"	"	"	"	"	"	"	"	"	—	—
MDA1	"	"	"	"	"	"	"	"	"	—	—
MAA1	"	"	"	"	"	"	"	"	"	—	—
MTA2	"	"	8	"	"	"	"	"	"	—	alpha B/W
RAAØ	"	"	7	"	"	"	"	"	"	—	—

Fig. 3-5 Example of Hardware Configuration

SOFTWARE CONSIDERATIONS

In this section the parameters that are strictly software dependent are discussed.

SIZE OF THE SYSTEM SEGMENTS

Some generation parameters relate to the size of the segments occupied by the operating system.

These parameters are influenced by the size of the selected 1-modules and so depend on:

- the system configuration to be generated
- the optional modules requested.

At configuration time the user must specify the amount of memory to be allocated for the Dynamically Allocated Memory (D.A.M.) and for the hardware pool. On the basis of what specified SYSCONF calculates the values of the following parameters:

- Number, size and allocation of the segments of the D.A.M., handled dynamically by the system as work areas at IPL time (Unit 1: NUMDAMSEG, SEG_NUMBER and SEG_SIZE parameters)
- Number and allocation of the segments of the hardware pool, handled by the hardware controllers (Unit 1: NUMHWDSEG and SEGNUMBER parameters)

In L1 systems with one MMU, up to seven segments can be dedicated to the D.A.M. and the hardware pool. These are allocated in the segments %08, %0C and %15 - %19.

In L1 systems with two MMUs, up to 20 segments can be dedicated to the D.A.M. and the hardware pool. These are allocated in the segments %40 - %4B and %50 - %57.

Some approximate values for D.A.M. and hardware pool area occupation by the components that require the most space in these, are as follows:

COMPONENT	D.A.M	HARDWARE POOL
O.S.	30 Kbytes	-
Work station	9.5 Kbytes per work station	if graphics work station, 0.5 segment per work station
printer	1.5 Kbytes per printer	-
BSC/3270 and SNA line	approx. 1 segment per line (64 Kbytes)	0.5 segment per board
PRI/3270 line	approx. 0.5 segment per line (32 Kbytes)	0.5 segment per board
MUX controller	-	0.5 segment per board, see Note 2
encryption controller	-	see Note 1
OMNINET controller (LCU 3323)	-	see Notes 1 and 2
ETHERNET controller (LCU 3345)	-	0.5 segment per board, see Note 2
LPU 3390 controller	-	see Notes 1 and 2
LPU 3396 controller	-	0.5 segment per board, see Note 2
LPU 3348 controller	-	see Notes 1 and 2
Lion 100+100 controller	-	1 segment

Note 1: There are two versions of this controller, the old version and the updated one. The former occupies one segment of the hardware pool and the latter half a segment.

Note 2: With n segments of the hardware pool, a theoretical maximum of:

- n controllers of the old type

- 2 * n controllers of the new or updated type

can be handled. Actually this can only be obtained if the controllers are physically adjacent two-by-two in the board cage and only in this way they can share a pool segment.

L1 Systems with one MMU Configurations

The user is advised to occupy all seven segments that are available when configuring the systems M30/M40 (that is, SEG_SIZE = %FFFF, and NUMDAMSEG = 7).

The reason for this is that the area of memory dedicated to the D.A.M and the hardware pool and not actually allocated at IPL time is released to the user when the IPL terminates.

Bearing the above in mind, the values given in the preceding table are required by the user to evaluate the following:

- Whether the seven segments made available by the system are sufficient or not to support the configuration.

- How to divide the seven segments between the D.A.M. and the hardware pool.

In M30/M40 systems the segments are usually assigned as follows:

%08, %0C, %15, %16	to the D.A.M.
%17, %18, %19	to the hardware pool

In M60 with one MMU system the segments are usually assigned as follows:

%08, %0C, %15, %16, %17	to the D.A.M.
%18, %19	to the hardware pool

For M30/M40 systems there are critical hardware configurations; when these are exceeded there are not enough D.A.M. and hardware pool segments. Two examples follow:

Configuration 1: Line concentrator

This configuration supports 12 lines (on LPU), 4 WSs (on MUX) and Encryption board.

12 lines	:	3 segments in pool
1 encryption and 1 MUX	:	1 segment in pool
System support	:	1 D.A.M. segment
Communication	:	1 D.A.M. segment
4 Work stations	:	1 D.A.M. segment

The total is 4 hardware pool segments plus 3 D.A.M. segments (the values are rounded up to the nearest segment).

Configuration 2: Terminal concentrator

This configuration supports 8 WSs (on MUX), 2 external lines and 1 internal line and Encryption board.

2 MUX	:	1 segment in pool
1 encryption and 1 LPU	:	1 segment in pool
1 internal line	:	1 segment in pool
System support	:	1 D.A.M. segment
Communication	:	1 D.A.M. segment
8 Work stations	:	2 D.A.M. segments

The total is 3 segments in the hardware pool plus 4 D.A.M. segments (the values rounded up to the nearest segment).

L1 Systems with two MMUs Configurations

In M60 system configurations there are no limits to the dimensions of the physical memory, but no more than a total of 20 segments can be allocated to both D.A.M. and hardware pool. Of these, a maximum of 12 segments can be dedicated to D.A.M. The default situation is as follows:

- Twelve segments are dedicated to the D.A.M. and are allocated in %40 - %4B.

- There are eight segments dedicated to the hardware pool and are allocated in %50 - %57.

It must be remembered that if a D.A.M. segment is dimensioned to 0 (SEG_SIZE = 0) it is the same as not declaring it at all in the D.A.M. segments.

FAMILIES, PROCESSES, PROGRAMS, CHANNELS AND FILES

SYSCONF requires a series of parameters relating to the software elements (number of families, processes, programs, channels and files connected and open) which are influenced by the user activity, by the system and by the user subsystems present and simultaneously active on the L1 MOS system.

The following table shows how much influence the system and the subsystems have on this data. The data relating to the user programs must be added to the data in the table.

For example, for the number of families and processes, executing a program under Shell is exactly the same as executing a Shell command (a family is created and a process activated for executing the program).

Similarly, a user program executed under MTS uses a family and a process which have already been defined for the software environment.

Once the user program has been activated, it has a direct influence on the pre-defined part of the programming interface (file connection, family/process creation, ...). It is the system analyst's responsibility to collect the data of the application.

The number of 'Connected Files' is that to be taken into consideration when evaluating the value for the FILENUMBER parameter (UNIT 50).

The MTS, Features and Functions manual must be read before evaluating this value if the Chained Data Base of the MTS software environment is used.

No instances are used for those components which are marked with the character (&): the values given in the following table for them are absolute.

For the other components, the values given in the following table refers to each activation (or to each work station, if they are activated by Grandpa). For example, the Shell interpreter consists of one family, two processes, and so on (as indicated in the table) for each work station with which it is associated via the Grandpa configuration file; analogously a compiler consists of the given families, processes, and so on for each time it is activated.

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
System:					
OS Resident (&)	1	3+No. CPU	-	-	-
Line Manager (&)	-	3 + 2 for each Line Manager server	-	-	-
Grandpa (&)	1	3 or 4 if Master W.S.	1	1	3 + No.printers + 5 * (No.programs and/or user packages)
Shell:					
Shell Interpr.	1	2	1	1	11
Shell Command	1	1	1	-	4 to 6
EDITOR	1	2	1	-	6 (structure) + (5*instance)
Sort/Merge:					
LANGUAGE	1	1	1	-	1
FULLSORT	1	2	2	-	1 + (2 * No. Input file)
MERGE	1	1	1	-	2 * (No.Input file) +(No.Output file)
DMS:					
DML Interpr.	1+No.CALL	1+No.CALL	1+No.CALL	No.CALL	11
DDL Interpr.	1	1	1	-	6
INITDD	1	1	1	-	4

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
Languages:					
BASIC Interpr. (BC and ST)	1	1	1+No.CALL	-	1 * each data file + (1 for each intermediate code file)
Graphic BASIC Interpreter	1+1	1+1	2	-	8 + (8*FONTFILE) + (2*File IMAGE)
COBOL Compiler	1	1	1	-	31
COBOL Run Time	1	1	1	-	10
ICE Compiler	1	1	1	-	15
ICRTS	1	1	1	-	10
BASIC Compiler	1	1	1	-	25
BASIC Run Time	1	1	1	-	2
FORTRAN	1	1	4	-	1 (structure) + (4*instance)
PASCAL+ Compiler	2	2	2	1	3 (structure) + (7*instance)
PASCAL+ Debugger	2	2	2	1	3 (structure) + (2 * each module in deb)
ZLOC	1	1	1	-	1 (structure) + (2*instance)
OLINK	1	1	1	-	4

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
VISA	1	1	2+No.forms	1	No.forms + (No.Valid.Prog.)
TFORM	1	1	1	-	3
PGU	-	-	1	-	(No.FONTFILE) + (2 FILE IMAGE)
RTGSP	-	-	1	-	(No.FONTFILE) + (2 FILE IMAGE)
Services:					
QUEMAN (&)	-	-	-	-	15
SPOOLER (&)	1 + No.printers	1 + No.printers	1 + No.printers	1	16 structure + (8*No.printers)
BATCH (&)	2	2	2	2	28 structure + (14*instance)
SERVER (&)	1	1	1	1	5
FSERVER	1	1	1	-	-
GSERVER	1	1	1	-	-

Note: For FSERVER and GSERVER the value given to the FSERVERINST and GSERVERINST parameters (UNIT 32) are also to be considered.

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
COMMIT_CMANAGER (&)	1	1	1	-	6
COMMIT	1 + No.WS	1 + No.WS	1 + No.WS	-	-

Note: No. WS = Number of the WS where the Commit program is running.

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
NEMOS (&) ALARM AGENT	1	-	1	-	1
NEMOS (&) END TO END AGENT	1	3	1	1	1
NMS LMS Agent (&)	1	2	1	-	2
LMS (&)	-	-	-	-	12
SYSLOG (&)	1	1	1	1	-
Userdump	-	-	-	-	(2 * No.WS) *(No.FA + 1)

Notes: No. WS = Number of WS on which an application program which uses this function runs.
No. FA = Number of files processed by each application program.

Software Environments:					
ESE	1	2	1	-	12 to 15

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
MTS:					
MSWMAN (&)	-	1	1	-	1
MTSCTLG (&)	-	1	1	-	-
GMAN (&)	1	2	3	1	3
MSWDIS (&)	1	2	1	-	-
STDSR (&) (for COMMIT)	No. COMMIT Instances (No.<= 3)	No. COMMIT Instances (No.<= 3)	No. COMMIT Instances (No.<= 3)	-	2*No.COMMIT Instances (No.<= 3)
STDSR (for STD. SERVER DIFAS_like)	No. Server Instances (No.>= 1)	No. Server Instances (No.>= 1)	No. Server Instances (No.>= 1)	-	-
TUMAN/GTSMAN	No. Server Instances (No.>= 1)	No. Server Instances (No.>= 1)	No. Server Instances (No.>= 1) + 1 + 1 (if COBOL) + 1 (if DB) + (No. TU of pre-load)	-	2
DUALLOG (&)	1	1	1	-	1
ITSC	1 + No.ITS Instances (No.>= 1)	1 + No.ITS Instances (No.>= 1)	1 + No.ITS Instances (No.>= 1)	-	-
SMAN/LMAN	1 * No.WS	1 * No.WS	1 + (1*No.WS) + 1 (if COBOL) + 1 (if DB) + 1 (Appl. Pr., see Note)	-	2

Notes:
- Instead of one application program, a file can be created (via EDIT) which contains from 1 to 16 path names of application programs, some (or all) of pre-load.
- No. WS = number of work stations, including background node(s), if any.

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
BEAM:					
BEAM_MAIN	1	2	1	1	15 + No. user file
BEAMMON	-	-	-	-	2
BEAM_CONF	1	1	1	1	20
ONE:					
COMMUNIC. MANAGER	1	3	1	1	-
SESSION LAYER	1	4	1	1	3
TRANSPORT LAYER	1	3	1	-	3
DISPATCHER	1	2	1	1	1
FTR_PRIMARY SLAM/SNAM	1	2	1	-	2
RPX_PRIMARY SLAM/SNAM	1	1	1	-	1
FTR_SECONDARY SLAM/SNAM	1	1	1	-	2
RPX_SECONDARY SLAM/SNAM	1	1	1	1	2
FILE TRANSFER UTILITY	1	1	1	-	-
FILE TRANSFER SERVER	1	2	1	1	1
MONITORING UTILITY	1	1	1	-	-
SLAM (ONE) CONFIGURATOR	6	6	6	-	10
SNAM (ONE/SNA) CONFIGURATOR	3	3	3	-	6

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component (cont.)

FUNCTIONAL AREA	FAMILIES * INSTANCE	PROCESSES * INSTANCE	PROGRAMS * INSTANCE	CHANNELS * INSTANCE	CONNECTED FILES
OLILAN:					
OLILAN SERVER	1 + (1 * DOS_SERVER instances)	1 + (1 * DOS_SERVER instances)	1	1	-
DOS_SERVER	-	-	-	-	30
NSERVER (&)	1	1	1	-	-
Note: The value given to the 'instances' parameter when activating the OLILAN SERVER module in the Grandpa configuration file must be taken into account for OLILAN SERVER and DOS_SERVER (see also the "Specific Server" section at the end of Chapter 3 and the "OLILAN" section in Chapter 10).					
Terminal Emulators:					
2780 EMULATOR	4	8	5	2	3
3780 EMULATOR	4	8	5	2	3
3270 EMULATOR SNA/BSC	2*LU1 2*LU2 + 2*LU3 + 1	1*family + 2	2*LU1 2*LU2 + 2*LU3 + 1	2*LU1 2*LU2 + 2*LU3 + 1	2
USER (#)	(#)	(#)	(#)	(#)	(#)
Note: (#) = Configuration elements and values that depend on the kind of application program. Interpreted application programs are not to be considered, as they are taken into account when considering the relevant interpreter.					

Tab. 3-6 Number of Families, Processes, Programs and Files for each MOS Component

I/O BUFFERS

All I/O operations are executed with the support of memory areas handled as buffers. At system generation time, information is requested on the size and number of buffers which the system analyst wants to reserve for the various components executing I/O operations. If a lot of buffers and/or large buffers are defined, I/O operations are improved. On the other hand, each buffer which is defined occupies an area of memory which must be taken into account.

Some supplementary information is given in this section on the buffers used for I/O towards disk.

The user has several options available for I/O operations towards disk:

- Private buffer pools, reserved for a process for a single file

- A system buffer pool, used by all the files which have not been reserved a private buffer pool by a process

System Buffer Pools

SYSCONF requests the number and size to be defined of the buffers making up the system buffer pool, which is used by the File System.

The space for the system buffer pool is taken from the space allocated to the Virtual Disk (which is normally a 64 Kbyte segment). The Virtual Disk space which is not used by the system buffer pool is used for the Memory Volume.

This aspect must be remembered to avoid the space allocated to the system buffer pool from over-reducing the space reserved for the Memory Volume. The number of files which can be defined in the Memory Volume may be insufficient, or there may not be enough space for the data belonging to the Memory Volume's files.

The buffer and buffer pool descriptors do not occupy space on the Virtual Disk but are allocated in other areas of memory which are used by the system (Dynamically Allocated Memory).

Private Buffer Pools

SYSCONF requests the number of private buffer pools and the global number of buffers belonging to them to be defined.

The space for the private buffer pools is dynamically taken from the space reserved for the Memory Volume. The Memory Volume therefore contains:

- descriptors of the files present in the volume (this space is allocated statically, according to the total number of forecast files: FILFORMEMVOL parameter of Unit 50; 156 bytes x file)
- data of the Memory Volume's files (this space is allocated dynamically when operations on files are requested)
- private buffers (this space is allocated dynamically when operations on files are requested)

The descriptors for the buffer pools and the private buffers do not occupy space in the Virtual Disk but are allocated in other areas of memory used by the system (Dynamically Allocated Memory).

DISTRIBUTED SYSTEM CONFIGURATION

This section gives some general information about the distributed system.

The aim of this information is to help the user during configuration and in distributing the environments on each machine in the system. It also supplies an explanation of some of the UNIT 32 parameters.

HARDWARE TOPOLOGY

The distributed system configurations use the following local networks:

- Ethernet, for connecting one or more local networks
- Omninet, for connecting only one local network
- Lion, for Cluster configurations

Ethernet Network

The following can be connected to this network:

- One or more L1 MOS systems with functions of server for one or more services (see the next section)
- One or more L1 MOS systems with Application/Terminal Server functions (see the next section)
- One or more L1 MOS systems which are master in a MOS cluster or more generally, with functions of server of one or more services of the subnetwork interconnected

This topology is defined as "interconnected local networks"; the following is an example of this:

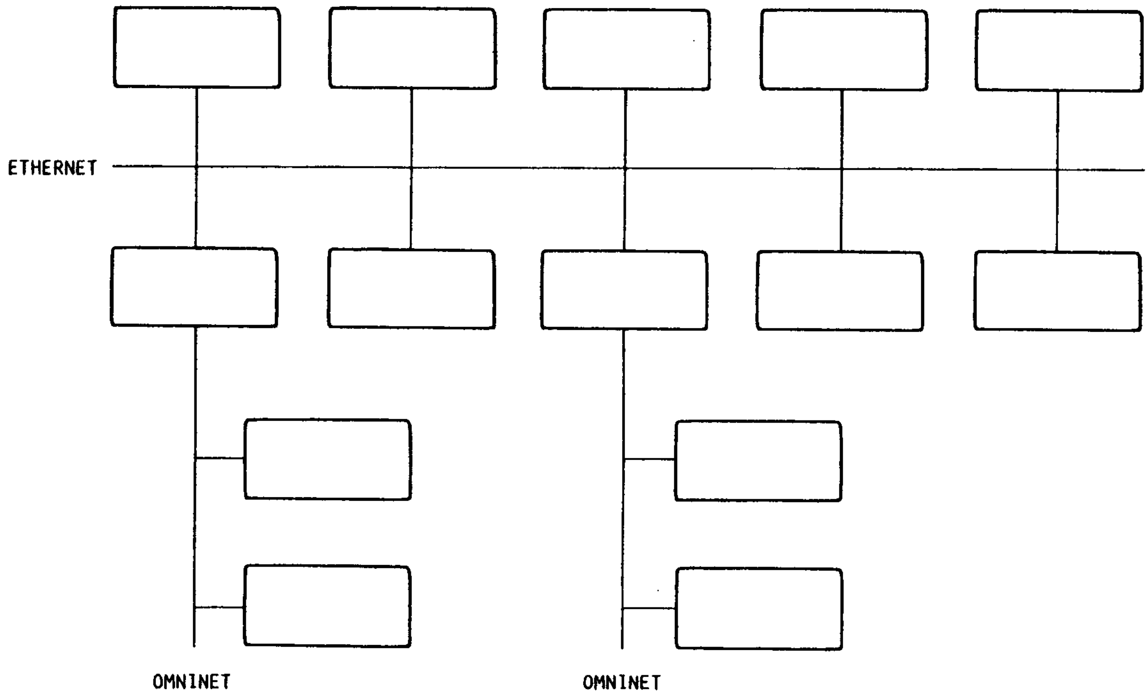


Fig. 3-7 Example of Interconnected Local Networks

Omninet Network

The following can be connected to this network:

- One or more L1 MOS systems with server functions for one or more services

- One or more L1 MOS systems with Application/Terminal server functions

Connections cannot be made to other networks.

SERVERS IN A DISTRIBUTED SYSTEM

A system which is part of a distributed system can be a:

- Server System, enabled to handle requests for services (access to local resources) coming from other machines in the distributed system

- Client System, enabled to request services from one or more servers. This system is known as the Application/Terminal Server.

The servers available at the moment are:

- **Route Server:** handles the distributed configuration. It controls correspondency between the logical name of all the machines in the network and their physical address, using the Routing Tables. For interconnected local networks, the Route Server only handles one network.

- **Name Server:** handles the catalogue of the distributed system global resources (files, devices, spooling and batch services, Communication Units, ...) by means of the Object Table. This table gives the correspondency between the logical name and the address of the resource. It is handled dynamically, as a resource can be inserted and retrieved at Operator request (Shell commands), or configuration directive request (GRANDPA). For interconnected local networks, the service only handles the resources in one network.

- **File Server:** handles access to local files upon request from users of different systems (with the File System Management features).

- **Spooler Server:** handles spooling services on the local printers (with the Spooling Management features).

- **Batch Server:** handles batch execution of programs (with the Batch Monitor features).

- **Communication Servers:** handle communications towards external environments via the line (these are the Line Manager Server, ONE Server and management of non-MOS terminals).

- **Transactional Server:** provides the Distributed Modular Transactional System (DMTS), with MTS features. Includes systems with MTS Server and Client functions (the latter is defined an "Application Server").

- **Commit Server:** handles the Commit remote services.

- **BEAM Server:** handles the global information (for example, users logged and activities running tables) and the configured Data Base stored on the local machine.

- **Local Management Server:** handles the system and the user logging.

- **Application/Terminal Server:** is a configured system comprising:

    - local File System services

    - application programs that access remote services.

Several servers can be resident on the same network system simultaneously and this can be a client system for the other remote servers.

Each server mentioned above has one or more agents in the network, that can reside in different systems from those on which there is the server. These agents allow access to remote servers, in other words this means that local applications can interface the corresponding server wherever it is.

The following figure illustrates the above situation as far as a general server is concerned.

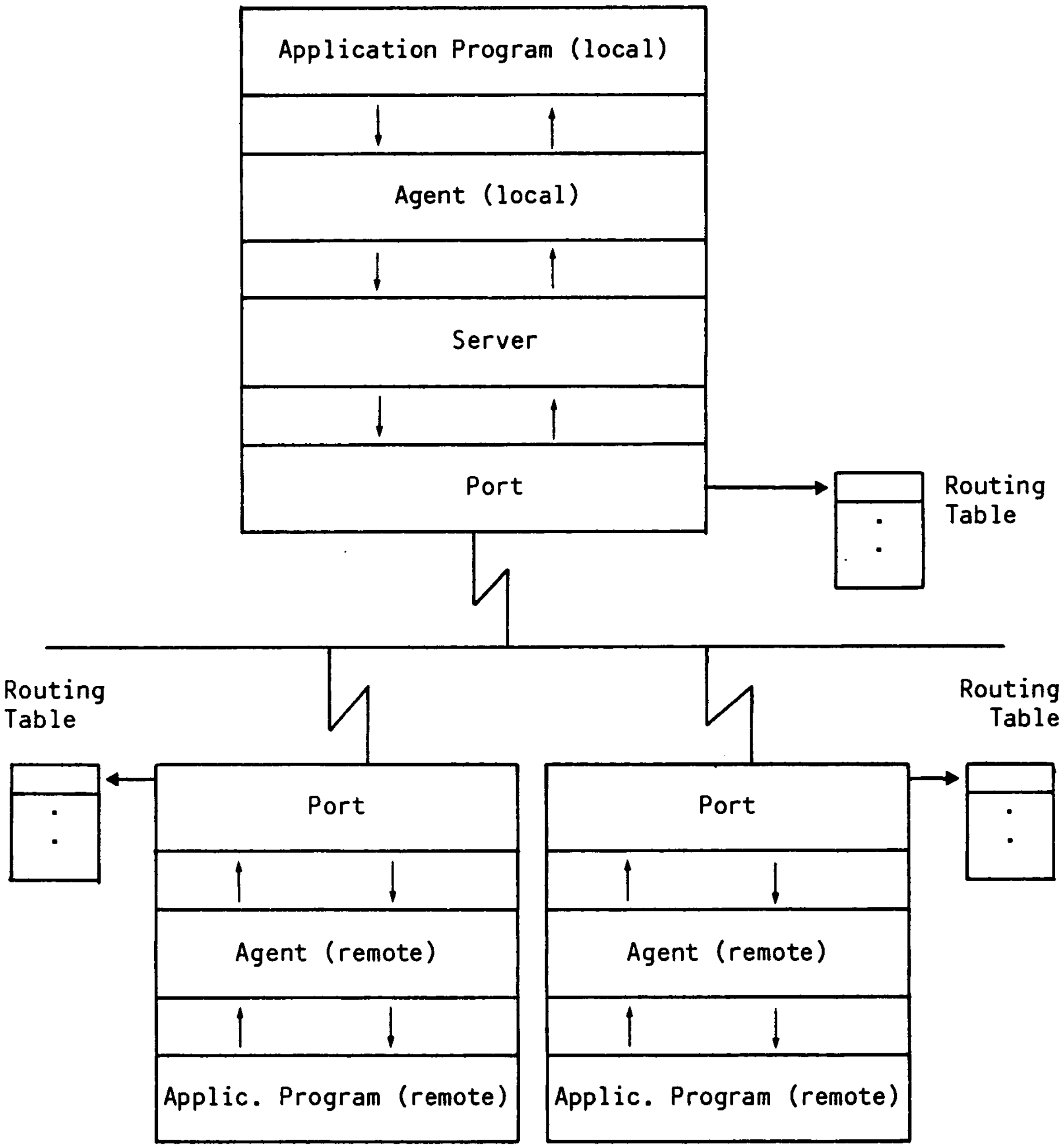


Fig. 3-8 The Server-Agent Functions in a Local Network

Agents interact with their servers via the Port Manager. The Port Manager routes the message over the network towards the server system, using the Routing Tables, from which it gets the system's physical identifier. These tables are continuously updated by the Route Server that interfaces its Route Agents, which are resident on each system in the network (not shown in the figure).

In a similar way to the other servers, the Name Server has its agent. This agent is present in each system in the network and interfaces the Name Server in order to:

- make a local resource global (by inserting it in the remote Object Table)

- obtain the identifiers of global resources (by accessing the remote Object Table).

To sum up what has already been said about agents:

- Both the Route Agent and the Name Agent are present in each system in the network.

- The other agents are only present in systems that request services from their servers.

Some limits exist on the number of servers in the network. See the Section "Software Configuration Limits".

The **Route Agent** and **Route Server** functions are implemented in the same l_module $CHM (by the Port Manager). The difference between the functions is specified in the configuration in each machine (LINECONF program).

The Route Agent only has the identifiers of the systems with which it has exchanged messages, in its Host Table at run time.

The Route Server has the identifiers of all the systems that are up in the network, in its Host Table at run time, and a significant Network Table (that contains as many entries as there are networks in the set of interconnected local networks).

The Host Table configuration is dynamic. The entries are created/deleted either when a machine in the network is turned on/off, or when the client machine accesses the functions of the server machine.

The **Name Server** and **Name Agent** functions are implemented in the l_module $SERVER which uses the $BST_2 functions. The difference between the Name Server and Name Agent functions is specified in the configuration of each machine (UNIT 32).

The Name Server has the identifiers of all the global objects in the network, in its Object Table at run time.

The Name Agent only has the identifiers of the objects that it has accessed, in its Object Table at run time.

The Object Table configuration is dynamic. The entries to the Object Table of the Name Server are created when a server of the network makes its objects global. The entries to the Object Table of the Name Agent are created when a local agent wants to access a global object (when an application program requests this).

The functions of the other servers/agents mentioned above (except for the File Server) are implemented in the various User Environments, which are configured using the parameters of the file $CONFGP (see Chapter 10).

LIMITS TO SOFTWARE CONFIGURATION

There are a few limits to the distributed system configurations, listed in the following:

- There is only one Name Server and one Route Server in each network.

- The Name Server and the Route Server must reside in the same system.

- In interconnected local networks, each network has its own Name Server and Route Server. The Route Server in the lower level subnetwork must reside in the system that joins the subnetwork to the principal network (case of the cluster).

- In interconnected local networks each system in a network only recognises global objects in its own network. For example, satellite systems in a Cluster connected to a network have no visibility of the servers and of the objects in the network.

- In interconnected local networks a generic Server can reside on the system which connects two subnetworks. The Server is then shared by both these subnetworks. This Server cannot be the Route Server or the Name Server.

The table that follows shows the maximum number of servers allowed for each network service. If the network comprises interconnected subnetworks, these limits refer to each subnetwork.

Two values are given for each service:

1. The maximum number of servers supported in the distributed system (N is the number of systems connected in the network, S is the number of physical subnetwork).

2. The maximum number of servers allowed in the current release.

SERVICE	MAXIMUM NUMBER OF SERVERS IN THE NETWORK:	
	SUPPORTED	ALLOWED
Batch	1	1
Spooler	8	8
BEAM	S	2
Commit	N	4
Communication (Line Man.)	1	1
File	N	8
MTS	N	4
Message Switching	N	128
LMS	N	128
ONE	N	N
OLILAN	N	N
File Transfer in ONE	1	1
File Transfer in SNA	1	1
Emulator 2780	1	1
Emulator 3780	1	1
Emulator 3270 SNA	8	8

Tab. 3-9 Table of the Maximum Number of Servers in the Network

Notes:

Batch: This restriction depends on the fact that the name of the batch queue is fixed. The batch server and the spooler queue (if this is present) must be resident on the same system.

Spooler: 8 spooling queues at most can be defined in the network; thus the systems on which the servers are allocated are minimum one (with eight queues) and maximum eight (with one queue each).

BEAM: There can only be one Beam Server in each network/subnetwork, because it is inserted in the network using an identifier that explicitly depends on the name of that network/subnetwork.

Commit: If there are several Commit Servers, static definition is made as to which server supports the requests from each of the machines in the network.

Communication: The communication server to Host can reside on only one machine and must have a fixed machine name.

File: Each of the servers allocated on different machines serves requests from any machine in the network. This is a result of the fact that the file servers of the network which are recognised by each machine are not predefined but the naming feature is used.

MTS: The maximum number of servers established is a result of an evaluation on the system performance rather than of software restrictions.

ONE: There can be more than one server (N = 1, 2, 3, ...) but the systems that can access each server are defined statically at ONE configuration time.

File Transfer: The ONE File Transfer server (FT monitor + primary) can reside on only one machine. Connection is made to the port server by means of one of the interface objects. The same thing applies to the SNA file transfer.

Emulator 2780: Only one server system can be defined that is accessed by the other systems by means of the secondary commands.

Emulator 3780: Only one server system can be defined that is accessed by the other systems by means of secondary commands.

Emulator 3270: The maximum number of servers is established by the software.

The example in Fig. 3-7 is completed by inserting the mandatory servers (Route and Name) and the optional servers (the others), and keeping in mind the limits mentioned before.

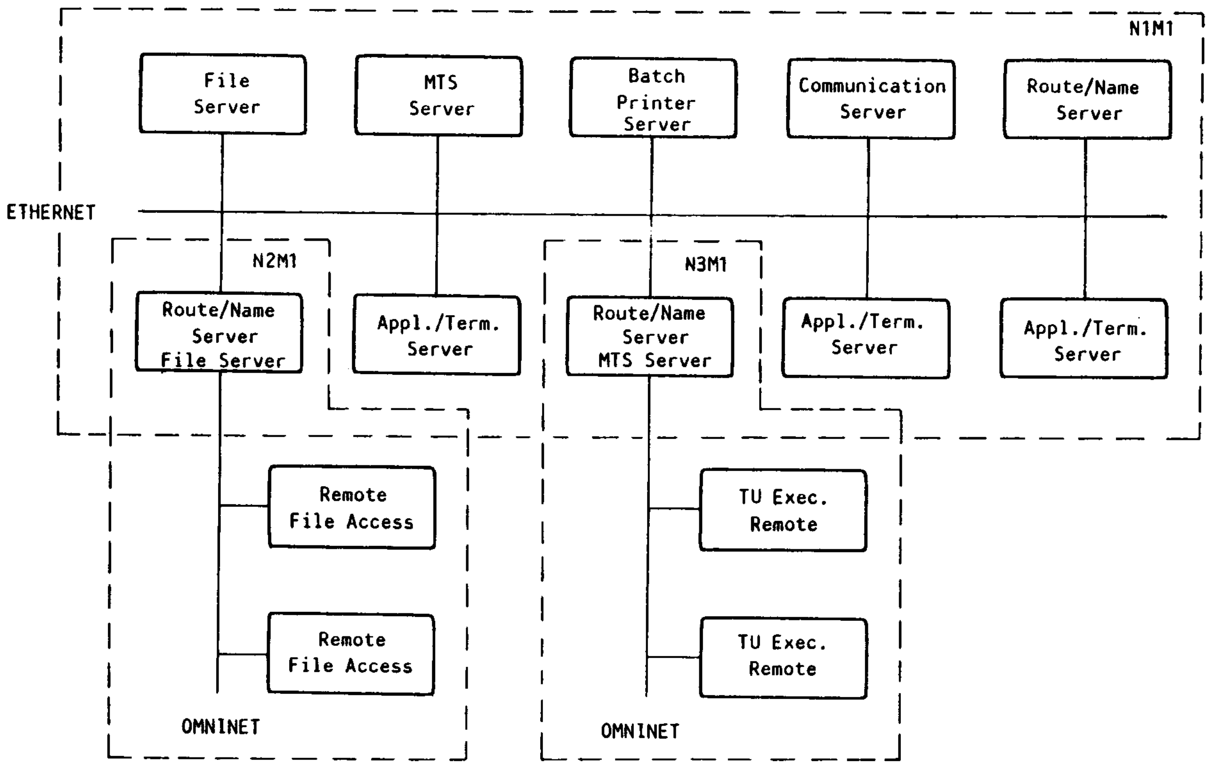


Fig. 3-10 Example of a Server Distribution in a Interconnected Local Network

The Route Server and Name Server of the Omninet networks must reside on the systems called N2M1 and N3M1; the Route Server and Name Server of the Ethernet network can reside on any of the systems connected. The dashed line indicates the three "domains" associated with the three networks interconnected. Each system in a network only accesses objects (such as files, servers, ...) which are in its domain.

In further developments (an object will be able to be accessed by all the machines in the set of interconnected local networks) the hierarchy of the N1M1 system will be higher that that of the N2M1 and N3M1. The Global Name Server will reside on this system that is defined as the Network Administrator (NSERVERTYPE = 2 in Unit 32).

As a result the N2M1 and N3M1 systems become the Domain Administrators (NSERVERTYPE = 1), that is the Domain Name Server resides on them. These systems are Name Agents for the Global Name Server and Name Servers for the other systems in their network.

The results of the restrictions in the current release are as follows:

- Only N2M1 and N3M1 systems (Masters, if the Omninet networks are Clusters) can access the servers and objects of the Ethernet.

- The other systems (Slave, if the Omninet networks are Clusters) can only access the servers and objects in the same network.

- The systems of the Ethernet network cannot access servers and objects allocated on systems of the Omninet networks, other than the N2M1 and the N3M1.

- The N2M1 and N3M1 systems operate as Application/Terminal Servers or servers to the Ethernet network.

CLUSTER CONFIGURATION

A Cluster is a particular network configuration of a local network in that it is a configuration where the servers are only allocated on the Master system.

If the Cluster is interconnected with a local network, the functions of the Master, as far as the other components in the network are concerned, are as follows:

- Application/Terminal Server: can contain application programs that access all the objects in the principal network.

- Server: can contain a principal network server. This server can be shared by the principal network and the Cluster.

The Cluster satellites access neither other network servers nor network objects. There are a few limits to this configuration:

- The MTS or the BEAM environment can be used alternatively, that is, only one of the servers can be present on the Master.

- For the MTS, the physical Cluster corresponds to the logical one.

REMOTE ACCESS COMPONENTS

A general server uses some basic operating system components for allowing remote access to its functions.

The three following modules provide the basic remote access mechanism:

- $SERVER, Server activator

- $FSERVER, Remote File Access Server (File System Server)

- $GSERVER, Remote Procedure Call Server

These operating system modules, together with the Remote Object Access (ROA) of the $BST_2 module and the Port mechanism of the $CHM module, are necessary for the following components of the release:

- MTS

- Message Switching

- Line Manager

- Spooling and Batch facilities

- 2780/3780 Emulator

- 3270 Emulator

- Master Work Station Handler

- Remote File Access from application environment

- BEAM

- Commit

The following table shows the requirements of these components for the various elements of the remote-use mechanism on different types of system. (I.L. = Internal Line, RFA option needed = the Remote File Access feature is selected in the last menu displayed by SYSCONF - see Chapter 6).

	SERVER SYSTEM	CLIENT SYSTEM	STAND-ALONE
MTS	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $CHM (Port) (RFA option needed)
MTS-DIFAS_like (see Note 1)			
	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $CHM (Port) (RFA option needed)
MTS-GTS_like (see Note 2)			
First : Server only with SERVER ENVIRONMENT			
	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $CHM (Port) (RFA option needed)
Second : Server with SERVER and CLIENT ENVIRONMENT			
	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $CHM (Port) (RFA option needed)
Message Switching	$SERVER $CHM (Port+I.L.)	$SERVER $CHM (Port+I.L.)	-

Tab. 3-11 Relationship of Remote Use Mechanism to Application Environment (cont.)

Note 1: MTS configuration to supply the same features as Rel. 4.0 DIFAS: Chained Files management without user Server Programs.

Note 2: MTS configuration to supply the same features as Rel. 4.0 GTS: user Server Programs management without Chained Files visibility and SECURE service utilization.

	SERVER SYSTEM	CLIENT SYSTEM	STAND-ALONE
Line Manager (CSS)	$BST_2 $CHM_(Port+I.L.)	$BST_2 $CHM_(Port+I.L.)	$BST_2
Batch and/or Spooling	$BST_2 $SERVER $FSERVER $GSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_1
Emulator 2780/3780	$BST_2 $SERVER $FSERVER $GSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_1
Emulator 3270 SNA	$BST_2 $CHM_(Port)		$BST_2 $CHM_(Port)
Master Work Station	$CHM (Port+I.L.)	$CHM (Port+I.L.)	$CHM (Port)
BEAM	$BST_2 $SERVER $FSERVER $GSERVER $CHM (Port+I.L.)	$BST_2 $CHM_(Port+I.L.)	$BST_1 $CHM_(Port)
COMMIT	$BST_2 $SERVER $FSERVER $GSERVER $CHM (Port+I.L.)	$BST_2 $SERVER $FSERVER $CHM (Port+I.L.)	$BST_2 $CHM_(Port) (RFA option needed)
OLILAN	$SERVER $GSERVER		$SERVER $GSERVER

Tab. 3-11 Relationship of Remote Use Mechanism to Application Environment

MOS CONFIGURATION FOR OLILAN

This section gives general information on OLILAN. It describes the software components which are required on the L1 MOS system to support OLILAN.

HARDWARE TOPOLOGY AND FUNCTIONS

An Olivetti Personal Computer can be connected to an OLILAN network in two ways:

- Via RS232 cable coming from the L1 machine (MUX or Twin); this is a cluster topology.

- Via Ethernet or Omninet cable; this is a bus topology.

The PC under OLILAN offers the following functions in both network topologies:

- MOS Terminal Emulator: MOS utilities and applications can be executed in both interactive and batch mode.

- MS-DOS: global files and network services can be accessed from the MS-DOS environment.

These two environments (emulated and MS-DOS) can coexist on the same machine.

In the **cluster** topology, the PC (running under MS-DOS) can only access the MS-DOS objects on the L1 MOS system to which it is connected. If, however, it is working as an L1 MOS terminal emulator, it can access the MOS services and global files of the whole network to which the L1 MOS system is connected.

In the **bus** topology, the PC can access all the network's services and global files.

The LAN limitations are still true for OLILAN: each system can only access the objects of its own subnetwork (see above the section "Distributed System Configuration").

Note: At least one integrated work station is necessary in any L1 system for "master" use at initialisation time.

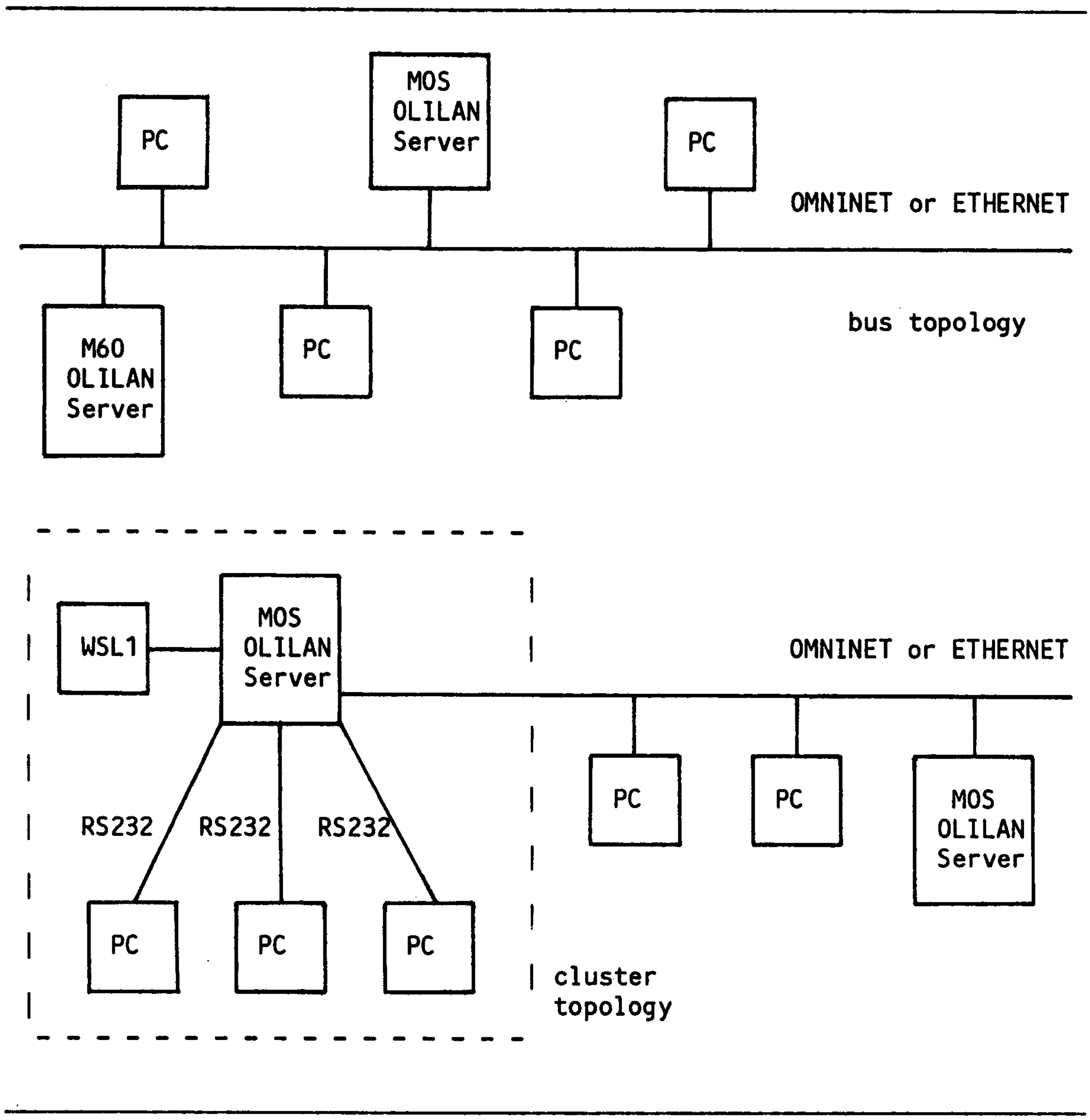


Fig. 3-12 OLILAN Topology Examples

In the first example given in the figure above, the PCs can access all the objects in the network, regardless of whether they operate as terminal emulators or in the MS-DOS environment. As terminal emulator, each PC may log in to any of the L1 MOS systems connected.

In the second example, the PCs can access all the network's global objects if they are operating as terminal emulators, but only the global objects resident on the L1 MOS system to which they are connected if they are working in the MS-DOS environment.

In both cases, if the networks shown in the figure were subnetworks of a larger network, then the PCs (like the other machines) would not be able to access the objects on the other subnetwork.

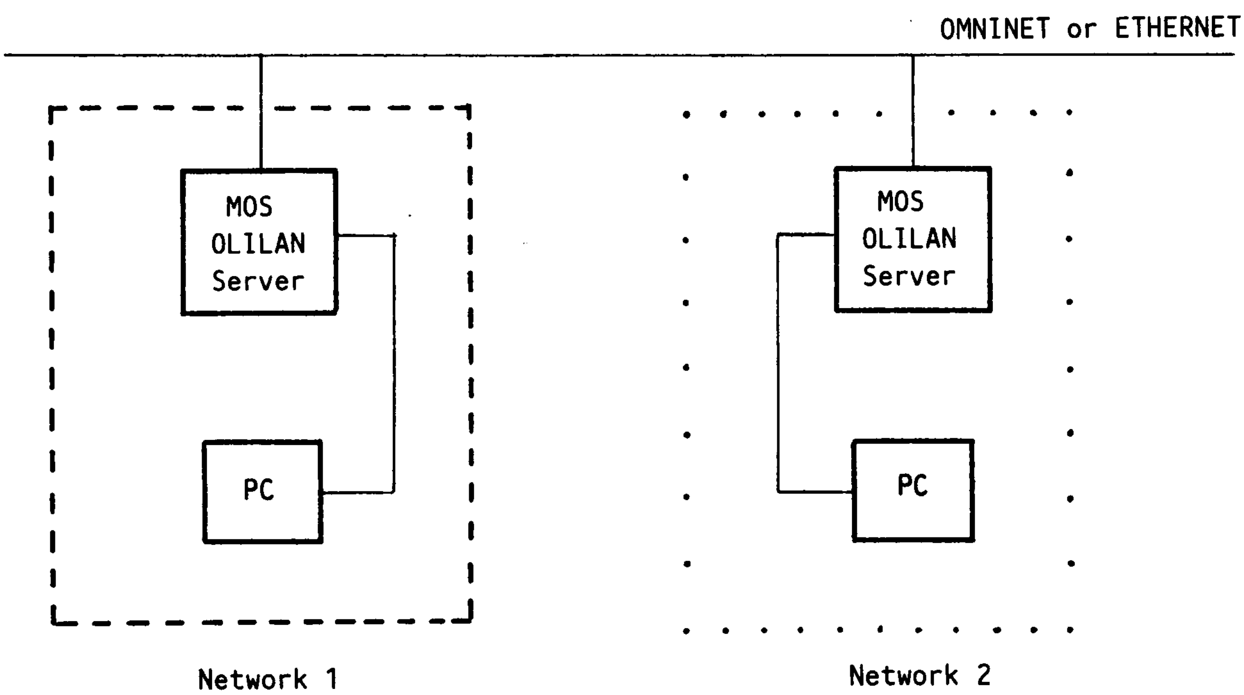
Two Levels Network

With Omninet or Ethernet network interconnections, it is possible to use a "two levels" network topology, that is, a network that can be seen as a set of subnetworks logically interconnected. The only existing constraints are that:

1. at least one Name Server must be present in each subnetwork

2. all the CONFNS files must be the same.

An example is given in the figure below.



Tab. 3-13 OLILAN "Two Levels" Network Topology Example

As mentioned above, the CONFNS files present in the two OLILAN Server system of this example must be identical.

CONFIGURATION STEPS

The following table summarizes which UNITs are involved when configuring the MOS operating system, according to the type of connection used for the PC in OLILAN.

RS232 With Ports	ETHERNET	OMNINET
UNIT 2 UNIT 4 UNIT 5	UNIT 4 UNIT 27	UNIT 4 UNIT 23

Tab. 3-14 UNITS to be Configured According to the Type of Connection in OLILAN Environment

In particular, the involved parameters are listed below together with their default or suggested values.

As far as UNIT 2, UNIT 4, UNIT 23 and UNIT 27 are concerned, the user is referred to the manual CSS Generation and Configuration, User Guide, where the relevant parameters and their interconnection are fully described. Those which follows are working examples, assuming that the rules described in that manual have been followed.

UNIT	PARAMETER NAME	VALUE	
UNIT 2	CONTR_NUM	1	
	NAMEPT	STRI	
	FLOWCTL	0	
	ACK_TIMEOUT	100L	
	PACK_PORT	0	
	RETRY_NUM	3	
	HALF_DUPLEX	0	
	BUF_NUM	4	
	BUF_SIZE	600	
	LINE_SET	1	
	SETRES_STOP	%0100	
	PAR_RXCHR	%0003	
	TXCHR_RXBAUD	%030C	
	TXBAUD_DUMMY	%0C00	
	CTLFLG_CTLM	%0100	
	BREAKFL_PERF	%0303	
	DTR_RTS	%0303	
	LINE_NAME	TAA0	[See the Note]
UNIT 4	GLO_PORT_NUM	20	
	PRI_PORT_NUM	20	
	TX_PORT_NUM	20	
	USR_MSG_SIZ	540	
	LOC_BUF_NUM	20	
	DELAYECHO	50	
	NUM_MACH	4	
	NUM_DRIV	2	
	NUM_NACK	7	
	NUM_FSR	2	
	TIME_OUT	20000	
	ROUT_FILE	RSETRTB	
UNIT 5	See Chapter 4 below for the description of the parameters belonging to UNIT 5 and their values.		

Note: The value given to the LINE_NAME parameter must be the same as that given in UNIT 5 to identify the channel (Twin or MUX) used for the connection.

UNIT	PARAMETER NAME	VALUE
UNIT 4	GLO_PORT_NUM	20
	PRI_PORT_NUM	20
	TX_PORT_NUM	20
	USR_MSG_SIZ	540
	LOC_BUF_NUM	20
	DELAYECHO	50
	NUM_MACH	4
	NUM_DRIV	2
	NUM_NACK	7
	NUM_FSR	2
	TIME_OUT	20000
	ROUT_FILE	RSETRTB
UNIT 27	BOARDS_NUM	1
	BOARD_ID	%006F
	DRIVER_ID	ETHR
	PHYS_ADD1	%0000
	PHYS_ADD2	%0000
	PHYS_ADD3	%0001
	TYPE_FLAG	1
	PROM_FLAG	1
	RX_SIZE	622
	RX_SKIP	28
	TX_SIZE	622

UNIT	PARAMETER NAME	VALUE
UNIT 4	GLO_PORT_NUM	20
	PRI_PORT_NUM	20
	TX_PORT_NUM	20
	USR_MSG_SIZ	540
	LOC_BUF_NUM	20
	DELAYECHO	50
	NUM_MACH	4
	NUM_DRIV	2
	NUM_NACK	7
	NUM_FSR	2
	TIME_OUT	20000
	ROUT_FILE	RSETRTB
UNIT 23	CONTR_NUM	1
	BOARD_ID	%007b
	PHY_ADD	%0001
	EAROM	0
	RXBUF1_SIZE	622
	RXBUF1_NUM	5
	RXBUF2_SIZE	0
	RXBUF2_NUM	0
	INT_LEV	1
	RETRY_NUM	60
	DRIVER_ID	OMN1

L1 MOS SERVERS

The mechanism used on LAN to access remote objects is adopted on OLILAN. It is based on a dialog between server and agent. A Name Server resides on the L1 MOS system (which, however, is different from the LAN Name Server), together with several other Servers for the various services.

Name Server

The Name Server uses a list of global resources written in the CONFNS file. Information on each resource is given in this file (position, files, volumes, ...). The file is created on the L1 MOS system using the NSPARSER utility. The complete phase of preparing this file is fully described in the OLILAN, Local Area Network, User Guide.

The Name Server reads the CONFNS file when the L1 MOS system is being initialized. When a PC client system wants to access a resource, it interacts with its own Name Server. This gives the position of the resource (using the information from the CONFNS file) or, to be more precise, the position of the Server which handles the resource on the L1 MOS system.

The network can have more than one Name Server. Each one has its own CONFNS file and the information contained in these files must be coherent (in particular regarding replicated structures). The PCs request the position of their Name Servers by sending a broadcast message at initialization time.

Specific Server

A PC can remotely access:

- MOS files: this feature allows PC applications to access L1 MOS files and is implemented by the DOS_SERVER File Server on MOS.

- The Spooler service: this feature allows PC applications to use printers connected to L1 MOS system and is implemented by the MOS Spooler Server.

- The Batch service: this feature allows PC applications to activate programs on the L1 MOS system and is implemented by the MOS Batch Server.

The OLILAN configuration on the L1 MOS system, exporting services to the PC, can therefore be summarized as including the following software components:

- Name Server, which corresponds to the NSERVER 1-module.

- File Server, which corresponds to the DOS_SERVER and SERVER 1-modules. The first is the actual File Server and the second handles client system initialization and failure.

The NSERVER and SERVER 1-modules are activated by Grandpa and must therefore be declared in the Grandpa configuration file (see the section "OLILAN" in Chapter 10).

The DOS_SERVER 1-module is activated by SERVER.

For 3278/79 Terminal emulation on a PC, note that an appropriate $CHM configuration is necessary, as described in the CSS Generation and Configuration, User Guide.

MOS CONFIGURATION FOR PCs USED AS L1 WORK STATION

A PC can be connected to an L1 MOS system thus permitting to use it as:

- L1 work station (running the L1WSE program or the OLIEMU program, or the WSELAN program in the OLILAN environment)

- stand alone Personal Computer.

The physical connection is made using:

1. an RS232 cable (and, optionally, a modem) between the RS232 interface present on the main board of the PC and an RS232 channel of the L1 MOS system (that is, Twin board, MUX)

2. a Current Loop cable between a serial interface present of the PC and a serial channel of a MUX connected to the L1 MOS system.

The following figure shows the various L1-PC connections.

L1 MOS / PC	RS CPU	SIC 3332/3367			MUX 3388			ETHERNET	OMNINET
		RS 232	MODEM	20ma CL	RS 232	MODEM	20ma CL		
L1 Work Station:									
L1WSE (1)	no	yes	yes	yes	yes	yes	yes	-	-
OLIEMU (2)	no	no	no	no	yes	yes	yes	-	-
OLILAN WSELAN (3)	no	yes	no	no	yes	yes	no	yes	yes

Tab. 3-15 L1-PC Connections

(1) No peripherals can be connected to the PC.

(2) Peripherals connectable to the PC are: printer, PIN Pad, badge reader/reader-writer, cash adapter and cheque reader/reader-writer.

(3) The only peripheral connectable to a PC in the OLILAN environment is a parallel printer (that is, a printer of /B type - IBM compatible, connected to the PC via the parallel interface).

The symbol "-" shows that the corresponding connection is not meaningful.

The PC is configured using the configurator program relevant to the emulator program to be run, as described in Part V.

The MOS operating system is configured declaring the PC as a work station connected via RS232. The involved parameters in this operation are listed below together with their suggested values.

RS232 Without Ports

UNIT	PARAMETER NAME	VALUE
UNIT 5	WSTYPEOLTER	%0101
	STOPBPAR	%0000
	RXTXCHLK	%0303
	PGMRXTXBL	%0C0C
	TANDEMCTLMO	%0101
	OUTSTATBRPE	%0000
	OUTSTDTRRTS	%0101
	INITRXBUF	2000
	INITTXBUF	2000
	INITXOFFLEV	1600
	INITXONLEV	600
	KBTABLEID	%0000
	WSTYPEID	%0000

RS232 With Ports

The user is referred to the homonymous section above, in the "MOS Configuration For OLILAN".

4. OPERATING SYSTEM CONFIGURATION PARAMETERS

This Chapter contains the description of all the parameters of the MOS operating system 1_modules which are necessary for generating the file $CON. They are listed in the same order in which they appear within the configuration file ($CON). Their values can be modified using the MODCON Shell command (see Part V).

Some Units are present in the $CON file that do not have configurable parameters. They are:

UNIT 16 (Encryption)
UNIT 18 (SCT Driver)
UNIT 24 (MTU Driver)
UNIT 29 (Performance Evaluation Program)
UNIT 30 (User PASCAL+ Run Time Support)
UNIT 31 (Local Management System)
UNIT 51 (Auxiliary functions)

They are not configurable by the user.

PARAMETER DESCRIPTION

The following information is given for each parameter:

- A definition
- The range of the values it can assume (if any).

Notes:

1) Default values for each parameter are not given as they are dynamically set by the system according to the answers given in the current configuration session. They can be seen by running the "SYSCONF" program (see Chapter 6). These default values permit the start up of a system in a minimum configuration.

2) Values for those parameters which are marked with an asterisk (*) are dynamically set by the system according to the answers given in the current configuration session. They can only be seen and modified (taking into consideration their possible relationship with other parameters) using MODCON.

CONTENTS

KERNEL PARAMETERS: UNIT 1

These are the kernel parameters.

USRFAMILY: Specifies the number of user families. It depends entirely on the application environment. If only the Shell application environment is present, this number may be obtained from the following formula:

2 + 2*ws_number

If other application environments are also present, the value of this parameter must be increased according to the number of Spawn primitives executed. See Chapter 3 "Processes, Families, Channels, Programs and Files".

-> A number of families too small relative to the requirements (number of Spawns) of the application environments intended to be used prohibits their activation.

-> Too large a number has the effects of occupying more D.A.M. area than is necessary for the allocation of the relative family descriptors.

Example: The value 10 activates the System Family Grandpa and Shell on four WS, irrespective of the type of machine used.

See Appendix A for memory allocation.

NUMPROCESS: Specifies the number of user processes. It depends entirely on the application environments used. If only the Shell application environment is present, this number may be obtained from the formula:

6 + (No. CPU-1) + 2*ws_number + application_prg_number*(1+fork_number)
+ 1[Master WS, if present]

See Chapter 3 "Processes, Families, Channels, Programs and Files".

-> A number too small for the requirements (number of Starts) of the application environments intended to be used prohibits their activization. Remember, however, that a user's application can use the sequence "Spawn, Load and Call" to carry out a program.

-> Too large a number has the effect of occupying more D.A.M. area than is necessary for the allocation of the relative process descriptors.

Example: The value 20 permits activation of the System, Grandpa and Shell processes on four WS, irrespective of the type of machine used.

See Appendix A for memory allocation.

MACHNAMEFLAG: Must be set to 1 when the name of the machine is stored in the $CON file (see the next two parameters). It is set to 0 when the name of the machine is stored in the EAROM. See Chapter 3 "Hardware Considerations".

NETWORKNAME: This parameter together with the next one identifies the machine within the global network system. This parameter identifies the network to which the system belongs. Its value must be in the range 1..255.

Warning: This parameter is meaningful only if the parameter MACHNAMEFLAG is set to 1.

HOSTNAME: Identifies the machine within the network. Its value must belong to the range 0..255.

For machines with an internal line on an intelligent controller (HDLC on LPU) it must be the same as the physical address of the station (see the manual CSS Generation and Configuration, User Guide, Chapter 4, Unit 38).

Warning: It is meaningful only if the parameter MACHNAMEFLAG is set to 1.

The previous two parameters must be consistent with the values given in the routing table (see the manual CSS Generation and Configuration, User Guide).

The next parameter and the ones that follow are not prompted to the user. Their values are calculated depending on the reply given to the SYSCONF questions: 'Number of hardware seg.' and 'Size of dynamic memory'.

NUMDAMSEG: Specifies the number of segments dedicated to the D.A.M. (Dynamic Allocated Memory). Seven segments at most can be dedicated to the D.A.M. and the Hardware Pool for L1 MOS systems with one MMU, and 20 for L1 MOS systems with two MMUs (of which no more than 12 can be dedicated to the D.A.M.).

For more details on the parameter and the ones that follow, see Chapter 3 "Size of the System Segment".

The two parameters that follow are repeated as many times as is specified in NUMDAMSEG.

SEG_NUMBER: Specifies the number of a segment dedicated to the D.A.M. in hexadecimal.

SEG_SIZE: Specifies the dimensions of the segment dedicated to the D.A.M. given in the previous parameter. It is a hexadecimal value. The value %0000 means that no D.A.M. area is allocated in the segment, and is the same as not declaring the segment at all.

NUMHWDSEG: Specifies the number of segments dedicated to the Hardware Segment Pool.

The following parameter is requested as many times as specified in NUMHWDSEG.

SEGNUMBER: Specifies the number of a segment dedicated to the Hardware Segment Pool in hexadecimal.

VIRTUAL DISK DRIVER PARAMETERS: UNIT 33

VDKMEMPAGES: Specifies the number of memory pages, each having 256 bytes, to be allocated to the Virtual Disk. The size of the memory area reserved for the operating system and the user files depends on the data structures required by the system. The space allocated to these data structures may be obtained from:

sys_data_size (in bytes) = 92n + 64n + 1024   (n = number of files)

Warning: For memory dimensioning see also the lower level File System parameter FILFORMEMVOL (Unit 50). See Chapter 3 "I/O Buffers".

The Virtual Disk structure is summarised in the following figure.

|←———— Virtual Disk ————→|

sys_data_size	Memory Volume
Memory area reserved for the operating system (for example, system buffer pools)	Memory area reserved for user files (for example, /DEV, /CONTEXT$ and /TMP directories)

Fig. 4-1 Virtual Disk

A free extent of at least 64 Kbytes must be available in the memory volume if the "speedy garbage collection" feature of the Interpreted BASIC is used.

2nd Char: identifies the controller output channel to which the object is connected. The values for this depend on the type of controller:

ELB/WSL1
A : object connected to the 1st RS232 channel
B : object connected to the 2nd RS232 channel
C : Badge Reader channel
D : PIN Pad channel
E : screen/keyboard channel

TWIN
A : object connected to the 1st RS232 channel
B : object connected to the 2nd RS232 channel

MUX
T : for peripherals connected directly to one of the four MUX lines. This means that the channel is not multiplexed (transparent). Otherwise the peripherals are connected to a WSL1.

KDC (integrated screen/keyboard)
E : if it is connected to a screen/keyboard, otherwise the connections are the same as for ELB.

RS of the CPU
A : always the same because there is only one RS232 output channel

3rd Char: relative position of the controller within the controllers of the same type (A, B, C, ...).

Example: 2 controller MUX and 4 KDC.

```
KDC ...      MUX ...      KDC
| |          | |          | |
A B          A B          C D
```

4th Char: identifies to which controller output line the object is connected. As mentioned before, the only controllers with output lines are MUX (4 lines) and KDC (1 line). For MUX its value can be in the range from 0 to 3 (the same as the value on the DBOX). For KDC it is always 0. For the other controllers the value is always 0 (ELB, TWIN, RSCPU).

Example: The WS peripheral keys (OSKARCHKEY) given in the following table are on the basis of this hardware configuration:

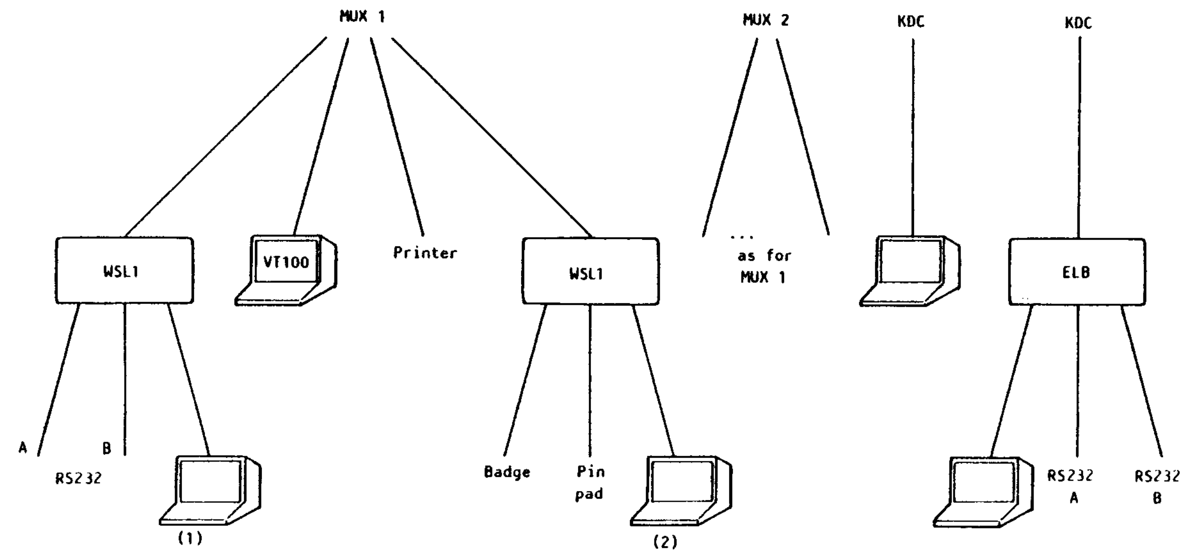


Fig. 4-2 Example of Hardware Configuration

	PERIPHERAL	OSKARCHKEY
MUX 1	screen/keyboard (1) VT 100 printer Badge PIN pad screen/keyboard (2)	MEA0 MTA1 MTA2 MCA3 MDA3 MEA3
MUX 2	screen/keyboard (1) VT 100 printer Badge PIN pad screen/keyboard (2)	MEB0 MTB1 MTB2 MCB3 MDB3 MEB3
KDC	screen/keyboard	EEA0
KDC-ELB	screen/keyboard	EEB0

The keys of the other configurable objects (lines, channels, RS232 drivers) are given in the sections specific to the objects.

Warning: For peripherals connected via the RS of the CPU, the key value is fixed: 'RAA0'.

Keeping to the rules described, for a peripheral connected via Twin the value of the key can be either of the following:

TAA0, TBA0 (channels A and B of the 1st Twin)

TAB0, TBB0 (channels A and B of the 2nd Twin)

..

..

and so on

WSTYPEOLTER: specifies the terminal characteristics. It is a two-byte hexadecimal value.

The value for the most significant byte can be:

%00: if it is not an Olivetti terminal (VT100, ...)

%01: if an Olivetti terminal is being used (integrated, WSL1, M30/M31, PC)

The value for the least significant byte can be:

%00: RS232 interface and no "new line" feature

%01: RS232 interface and automatic "new line" feature

%02: Current Loop interface and no "new line" feature

%03: Current Loop interface and automatic "new line" feature

The value of the least significant byte, for an Olivetti terminal, must be:

%01	integrated
%01	WSL1
%01 or %03	M30/M31
%01 or %03	PC

OSKARHCPRID: Hard copy printer key. If this printer is not requested the parameter must be filled with "*".

This parameter is made up of 4 characters whose meaning is the same as OSKARCHKEY. Only the WS printers (and not the system printers) can be used for the hard copy.

Printer Definition

PRREQUEST: Specifies whether the WS printer is present or not. Its value can be:

0 : no printer
1 : printer present

PRT_CHKEY: Printer key. For its structure, see OSKARCHKEY.

Warning: The channel indicated for the printer connection must be coherent with that defined in the work station SET-UP (the SET-UP process is fully described in the L1, ELB 3683 General Service Manual, Code 4105690E, and in the L1, ELB 3684 Service Manual, Code 4114250B).

PRT_PRMODEF: Specifies the connection mode and the communication protocol. It is a two-byte hexadecimal value.

The value of the most significant byte can be:

%00 : no Current Loop
%01 : Current Loop

The value of the least significant byte can be:

%00 : Olivetti controlled mode
%01 : Free Running mode (Typically for PR/B printers)

PRTHARDCP: Specifies the hard copy subdevice and is an integer in the range from 0 to 4:

0 = no hard copy
1 = hard copy on default subdevice
2 = hard copy on AFF (Automatic Front Feed)
3 = hard copy on SFF (Semi-automatic Front Feed)
4 = hard copy on ASF (Auto Sheet Feeder)

If the value 0 is specified no hard copy can occur even if a WS printer has been defined. Any work station printer can be an hard copy printer. If hard copy is requested on a WSL1, then the printer must be physically connected to the WSL1 itself.

SHARING: A two-byte hexadecimal value.

The most significant byte (target) specifies the WSID of the WS that is the owner of the printer that the current WS wants to share. It is only significant if PRREQUEST = 1. If this byte is set to 0 a new printer is defined. The WS identifier contained in target must be that of a previously configured work station.

The least significant byte (light) identifies which LED on the printer is associated with the WS. It is also used to handle the WS printer lock (if the printer is shared). This LED lights up when the WS is using the printer. For the owner WS this value is %01.

No more than four WS can share the same printer.

It is advisable to set the least significant byte in sequential order from %01 to %04, making sure that different WS, sharing the same printer, have a different value for this parameter.

Example: WS 4 is the printer owner; WS 5, 6 and 7 want to share the printer.

The value for the parameter SHARING is:

WS	SHARING (target + light)
4	%0001
5	%0402
6	%0403
7	%0404

Warning: On WS connected via MUX, hard copies can only be produced on the work station printer actually connected to the current WS (whether it is shared or not).

PIN Pad Definition

PINPADREQUE: Specifies whether there is a PIN Pad or not. Its values can be:

0 : no PIN Pad
1 : PIN pad present

PINPADCHKEY: PIN Pad key. For its structure see OSKARCHKEY.

Badge Reader Definition

BADGEREQUE: Specifies whether there is a Badge Reader or not. Its values can be:

0 : no Badge Reader
1 : Badge Reader present

BADGECHKEY: Badge Reader key. For its structure see OSKARCHKEY.

Cheque Reader/Writer Definition

MICRREQUE: Specifies whether there is a cheque reader/writer (MICR) or not. Its values can be:

0 : no Cheque Reader/Writer
1 : Cheque Reader/Writer present

MICRCHKEY: Cheque Reader/Writer key. For its structure see OSKARCHKEY.

MICRMLTYPE: Specifies the MICR device type and is an integer value. It can be:

0 : Cheque Reader
1 : Cheque Reader/Writer

Cash Adapter Definition

CADPREQUEST: Specifies if there is a Cash Adapter or not. Its value can be:

0 : no Cash Adapter
1 : Cash Adapter present

CADPCHKEY: Cash Adapter key. For its structure see OSKARCHKEY.

CADPCATYPE: Specifies the CADP configuration and is an integer value:

0 Indicates NORMAL (that is a sharable moving delivery throat CADP)
1 Indicates TELLER (that is a non sharable fixed delivery throat CADP)

CADPSHARING: Identifier (WSID) of the work station in which there is the CADP that the current work station wants to share. If the value is 0 a new CADP that is not shared is defined.

SYSTEM PRINTER DEFINITION

SYSPRTNUMBE: This parameter specifies the total number of system printers in the system configuration. It is an integer value in the range 0 to 8.

The following parameters of system printer definition are repeated for each system printer in the configuration.

SYSPRTCONF: Specifies whether the printer being configured is present or not. Its value can be:

0 : no printer
1 : printer present

SYSPRTCHKEY: System printer key. For its structure see OSKARCHKEY.

SYSPRTMODE: Specifies the connection mode and the communication protocol. It is a two-byte hexadecimal value.

The value for the most significant byte can be:

%00 : no Current Loop
%01 : Current Loop

The value of the least significant byte can be:

%00 : Olivetti controlled mode
%01 : Free running mode

It must be set the free running mode for system printers.

RSDRINSTNUM: Specifies the number of RS232 driver instances to be created in the system. It is the total number of devices connected to the RS232 driver in the system configuration. It is an integer value in the range 0 to 32.

The following parameters that define each RS232 driver are requested as many times as specified in RSDRINSTNUM.

RSDRCHKEY: RS232 channel key. For its structure see OSKARCHKEY.

Example: Referring to the example given for OSKARCHKEY, the keys for the six RS232 drivers are given below.

DRIVER	RSDRCHKEY
RS232 A in output from WSL1 connected to MUX1	MAA0
RS232 B in output from WSL1 connected to MUX1	MBA0
RS232 A in output from WSL1 connected to MUX2	MAB0
RS232 B in output from WSL1 connected to MUX2	MBB0
RS232 A in output from ELB	EAB0
RS232 B in output from ELB	EBB0

RSDROFFCTL1: Specifies the handling mode for the DSR (Data Set Ready) and CTS (Clear to Send) control signals. This is a two-byte hexadecimal value.

The values for the most significant byte can be:

%00 : DSR significant
%01 : DSR non-significant

The values for the least significant byte can be:

%00 : CTS significant
%01 : CTS non-significant

RSDROFFCTL2: Specifies how the DCD (Data Carrier Detector) message is to be handled. It is a two-byte hexadecimal value. The value for the least significant byte is %00; the value for the most significant byte can be:

%00 : DCD significant
%01 : DCD non-significant

CHANNEL DEFINITION

*** CHNUMBER:** Is the total number of channels configured. Each peripheral has one channel and vice versa. This value must consequently be the same as the number of peripherals configured (including the peripherals connected via the RS232 driver, the system printers and the debugger (see UNIT 54)). The maximum theoretical value is 256.

The following parameters used to define each channel are requested as many times as specified in CHNUMBER.

CHITEMCHKEY: Channel key. For its structure see OSKARCHKEY.

Example: Referring to the example given for OSKARCHKEY, the keys for the 20 channels are given below:

	CHANNEL	CHITEMCHKEY
MUX 1	RS232 A	MAA0
	RS232 B	MBA0
	screen/Keyboard (1)	MEA0
	VT100	MTA1
	printer	MTA2
	Badge	MCA3
	PIN pad	MDA3
	screen/Keyboard (2)	MEA3
MUX 2	RS232 A	MAB0
	RS232 B	MBB0
	screen/Keyboard (1)	MEB0
	VT100	MTB1
	printer	MTB2
	Badge	MCB3
	PIN pad	MDB3
	screen/Keyboard (2)	MEB3
KDC	screen/Keyboard	EEA0
KDC-ELB	screen/Keyboard	EEB0
	RS232 A	EAB0
	RS232 B	EBB0

The user will have noted that the channel keys are exactly the same as the keys for the peripherals defined for the work stations, plus the keys of the RS232 drivers.

STOPBPAR: Two-byte hexadecimal value. The most significant byte contains the number of stop bits and its value can be:

VALUE	NUMBER OF STOP BITS
%00	1 (mandatory for PC connection)
%01	1.5
%02	2

The least significant byte gives the type of parity check, and its value can be:

%00 : parity check not requested
%01 : odd parity
%02 : even parity

RXTXCHLK: Gives the length of the character being received/transmitted. It is a two-byte hexadecimal value. The two bytes must have the same value, that is the transmission character length must be the same as the reception length. The possible values are:

VALUE	LENGTH
%00	5 bits
%01	6 bits
%02	7 bits
%03	8 bits (mandatory for PC, M30 and M31 connection)

PGMRXTXBL: Gives the reception/transmission speed. It is a two-byte hexadecimal value. The two bytes must have the same value, that is the reception speed must be the same as the transmission speed. The possible values are:

VALUE	MEANING
%0E	38400 bps
%0D	19200 bps
%0C	9600 bps
%0B	4800 bps
%0A	2400 bps
%09	1800 bps
%08	1200 bps
%07	600 bps
%06	300 bps
%05	200 bps
%04	150 bps
%03	134.5 bps
%02	110 bps
%01	75 bps
%00	50 bps

The speed range permitted for a PC connection is from 110 to 19200 bps.

TANDEMCTLMO: Gives information about the XON/XOFF protocol (tandem mode). It is a two-byte hexadecimal value.

The most significant byte specifies the tandem mode setting. This is a peripheral-dependent characteristic. The values can be:

VALUE	MEANING
%00	reset tandem mode
%01	set tandem mode to XON/XOFF
%02	set tandem mode to BREAK
%03	set tandem mode to CTS

The least significant byte specifies if the tandem mode is supported in reception, transmission, or both. If tandem mode is supported in reception, when the XOFF signal is received from a peripheral, transmission is suspended until an XON is received. If it is supported in transmission, when the reception buffer is full, an XOFF suspends the transmission of characters by the peripheral. The values of this are:

VALUE	MEANING
%00	reception and transmission
%01	only reception
%02	only transmission

<u>OUTSTATBRPE:</u> Two-byte hexadecimal value. The most significant byte sets the state of the break signal. The values can be:

VALUE	MEANING
%00	mark
%01	space (breaker)
other values	no operation

The least significant byte sets the state of a general signal, depending on the peripheral. The value for this can be:

VALUE	MEANING
%00	logical 0
%01	logical 1
other values	no operation

<u>OUTSTDTRRTS:</u> Two-byte hexadecimal value. The most significant byte sets the state of the DTR signal (Data Terminal Ready). The values can be:

VALUE	MEANING
%00	logical 0
%01	logical 1
other values	no operation

The least significant byte sets the state of the RTS (Request to Send) signal. The values for this can be:

VALUE	MEANING
%00	logical 0
%01	logical 1
other values	no operation

INITRXBUFF and INITTXBUFF: Two integers that specify the dimensions of the input and output buffers respectively. These values must be selected on the basis of the data traffic with the peripheral.

In case of PIN Pads, dimensions are fixed to 16 - 16. In case of Badge readers, dimensions are fixed to 110 - 110. For the asynchronous windows (implicitly defined when the second character of OSKARCHKEY is 'E'), dimensions are fixed to 80 - 80 for WSL1 and to 80 (INITTXBUFF) and 120 (INITRXBUFF) for integrated work stations.

For the MUX, the sum of the dimensions of the buffers defined, plus the areas that are preallocated statically, must not exceed 16 Kbytes, because this is the limit of the Dual Port Memory (memory area reserved for data exchange via MUX). The dimensions of the statically preallocated areas are 256 bytes plus 128 bytes for each channel (for the output and input buffer respectively). If a screen/keyboard channel is being used and hand copy is to be made, the minimum value of the INITRXBUFF is 128. If a colour graphic work station is connected to the channel the minimum value of the INITRXBUFF is 256.

INITXOFFLEV and INITXONLEV: These parameters specify an integer number of bytes in the receive (RX) buffer at which points transmission is turned on or off by the Xon and Xoff control characters.

A difference must exist so as to avoid a deadlock situation. This difference must be carefully calculated so as to achieve a good synchronisation on these events. And moreover, to avoid degradation of the asynchronous operation mode of the driver.

KBTABLEID: Keyboard table identification code. This is only significant for integrated terminals (KDC). It must be remembered that WSL1 translates the key code if the terminals are connected to it.

The following table gives the hexadecimal values that correspond to the keyboard tables.

The string "TRANSTAB" is a parameter that the user must give later on in configuration (see SCREEN/KEYBOARD TABLES DEFINITION).

NATIONAL VERSION AND TYPE		VALUE	"TRANSTAB"
Denmark	business	%FBE6	DENMARK_DP
"	scientific	%FBA6	DENMARK_BAS
England	b.	%FBF0	ENGLAND_DP
"	s.	%FBB0	ENGLAND_BAS
France	b.	%FBE7	FRANCE_DP
"	s.	%FBA7	FRANCE_BAS
Germany	b.	%FBE3	GERMANY_DP
"	s.	%FBA3	GERMANY_BAS
Germany DIN2137	b.	%FAE3	GERMDIN_DP
" "	s.	%FAA3	GERMDIN_BAS
Greece	b.	%FBE8	GREACE_DP
Kana	b.	%FAEB	KANA_DP
"	s.	%FBBF	KANA_BAS
International	b.	%FBE0	INTER_DP
Italy	b.	%FBEA	ITALY_DP
"	s.	%FBAA	ITALY_BAS
Israel	b.	%FBE9	ISRAEL_DP
Leumi	b.	%FAE9	LEUMI_DP
Norway	b.	%FBEC	NORWAY_DP
"	s.	%FBAC	NORWAY_BAS
Portugal	b.	%FBE4	PORTUGAL_DP
Spain	b.	%FBE5	SPAIN_DP
"	s.	%FBA5	SPAIN_BAS
Sweden/Finland	b.	%FBED	SWFL_DP
" "	s.	%FBAD	SWFL_BAS
Swiss (German)	b.	%FAEE	SWISG_DP
" "	s.	%FAAE	SWISG_BAS
Swiss (French)	b.	%FBEE	SWISR_DP
" "	s.	%FBAE	SWISR_BAS
USA ASCII	b.	%FBF1	USAASCI_DP
" "	s.	%FBB1	USAASCI_BAS
USA OCA	b.	%FBF2	USAOCA_DP
Yugoslavia	b.	%FBF3	YUGOS_DP
USSR	b.	%FBEF	USSR_DP
CCCP	s.	%FBFF	CCCP_BAS
Japan	b.	%FBEB	NOKYO_DP
"	s.	%FBAF	NOKYO_BAS

Tab. 4-3 Keyboard Tables

NATIONAL VERSION	VALUE	"TRANSTAB"
Denmark	%FB86	DENMARK_M
England	%FB90	ENGLAND_M
France	%FB87	FRANCE_M
Germany DIN2B7	%FB96	GERMDIN_M
Greece	%FB88	GREACE_M
Israel	%FB89	ISRAEL_M
Italy	%F88A	ITALY_M
Norway	%FB8C	NORWAY_M
Portugal	%FB84	PORTUGAL_M
Spain	%FB85	SPAIN_M
Sweden/Finland	%FB8D	SWFL_M
Swiss (German)	%FB94	SWISG_M
Swiss (French)	%FB95	SWISR_M
USA ASCII	%FB91	USAASCI_M
Yugoslavia	%FB93	YUGOS_M
Spain extended	%FB98	SPAIN2_M

Tab. 4-4 Multi-functional Keyboard Tables

INPATTLEDS: Reserved for future use.

WSTYPEID: Identification code for the screen type. This is only significant for integrated terminals. The only possible value for WSL1 terminals is %00F1.

VALUE	SCREEN TYPE
%00F1	alphanumeric monochromatic or colour
%00F2	Kanji
%00F3	graphic monochromatic
%00F4	graphic colour

Warning: The channel parameter default values have been chosen, so that they are valid for most of the peripherals and terminals connected to different controllers. These parameters solve connection problems in 90% of the configurations.

LINE DEFINITION

LNNUMBER: Is the total number of lines in the system. Only the KDC and the MUX have output lines. KDC has one and MUX has four. The maximum value is 32.

The following parameters used to define each line are requested as many times as specified in LNNUMBER.

LNITEMCHKEY: Line identification key. As for the other keys, it comprises four characters whose meaning is given below:

1st Char: identifies the controller, and may be:

- E if KDC line
- M if MUX line

2nd Char: not strictly significant. It must be set to the identifier of a channel (usually the first one) present on the device connected to the line. If a terminal device is connected the values are T or E.

3rd Char: is the relative position of the slot in the controller

4th Char: identifies the line, and may be:

- 0 if KDC line
- 0 to 3 if MUX line (same as the value on the DBOX).

Example: Referring to the example given for OSKARCHKEY, the keys for the 10 lines are given below:

	LINE	LNITEMCHKEY
MUX 1	1st line	MAA0
	2nd line	MTA1
	3rd line	MTA2
	4th line	MCA3
MUX 2	1st line	MAB0
	2nd line	MTB1
	3rd line	MTB2
	4th line	MCB3
KDC		EEA0
KDC-ELB		EEB0

LNITEMRXTX: Gives the line reception/transmission speed. It is a two-byte hexadecimal value. The two bytes must be set to the same value, that is the reception speed must be the same as the transmission speed, and may be:

VALUE	MEANING
%0D	19200 bps
%0C	9600 bps
%0B	4800 bps
%0A	2400 bps
%09	1800 bps
%08	1200 bps
%07	600 bps
%06	300 bps
%05	200 bps
%04	150 bps
%03	134.5 bps
%02	110 bps
%01	75 bps
%00	50 bps

The only possible KDC line values are %08 and %0B.

WORK STATION FOR DYNAMIC LOGIN SUPPORT

NIMWSNUMB: Specifies if support to dynamic login is required, and its value may be:

- 0 if no dynamic login are requested
- 1 if dynamic login are requested.

M31/M30 WORK STATION DEFINITION

M31M40LINEN: specifies the number of the RS232 line that connects the M30/M31 system to the L1 MOS system. This is only significant in the M30/M31 system configurations. In other L1 system configurations its value must be 0. The possible values are:

VALUE	MEANING
0	L1 system (not a work station)
2	Twin RS232 A Channel
3	Twin RS232 B Channel

SCREEN KEYBOARD TABLE DEFINITION

* **TRANSCODTBN:** Is the total number of keyboard tables defined. These tables are only required for integrated terminals. If several terminals use the same keyboard this must only be counted once.

The following parameter is requested as many times as specified in TRANSCODTBN.

TRANSTAB: A character string identifying the keyboard table to be loaded. This keyboard table is contained in the file whose name is specified in this parameter. The possible strings are those given for the parameter KBTABLEID in section "CHANNEL DEFINITION".

FDU DRIVER PARAMETERS: UNIT 14

* **FDUNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

* **FDUUNITSNUM:** Specifies the number of units connected to the controller. It is an integer in the range 1 to 4.

MFU 320K PARAMETERS: UNIT 11

*** NUMOPERAT320:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

*** UNITSNUM320:** Specifies the number of units connected to the controller. It is an integer in the range 1 to 4.

MFU 1M PARAMETERS: UNIT 13

* **MFDNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

* **MFDUNITSNUM:** Specifies the number of units connected to the controller. It is an integer in the range 1 to 4.

* **HDUNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

HDUUNITSNUM: Specifies the number of units connected to the integrated controller.

This parameter is a sequence of positional descriptors which have the value 1 when the corresponding peripheral is connected (0 otherwise).

Warning: The unit number (from 1 to 8) must be used for defining the logical disk in Unit 50 (Byte Stream).

SMD DRIVER PARAMETERS: UNIT 20

*** SMDNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

SMDUNITSNUM: Specifies the number of units connected to the SMD controller.

This parameter is a sequence of positional descriptors which have the value 1 when the corresponding peripheral is connected (0 otherwise).

Warning: The unit number (from 1 to 8) must be used for defining the logical disk in Unit 50 (Byte Stream).

* **HDCNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

HDCUNITSNUM: Specifies the number of units connected to the HDC controller by means of the ST506 interface.

This parameter is a sequence of positional descriptors which have the value 1 when the corresponding peripheral is connected (0 otherwise).

Warning: The unit number (from 1 to 8) must be used for defining the logical disk in Unit 50 (Byte Stream).

SASI DRIVER PARAMETERS: UNIT 6

* **SASNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

SASUNIT: Specifies the number of units connected to the SASI controller.

This parameter is a sequence of positional descriptors which have the value 1 when the corresponding peripheral is connected (0 otherwise).

Warning: The unit number (from 1 to 8) must be used for defining the logical disk in Unit 50 (Byte Stream).

ESDI DRIVER PARAMETERS: UNIT 7

* **ESDNUMOPERAT:** Specifies the maximum number of operations using the buffer pool of the driver at the same time. It is an integer value.

Warning: It can be defined by adding 2 to the number of programs which has been specified in the NUMPROGRAM parameter, Unit 10 (PMM).

ESDUNIT: Specifies the number of units connected to the ESDI controller.

This parameter is a sequence of positional descriptors which have the value 1 when the corresponding peripheral is connected (0 otherwise).

Warning: The unit number (from 1 to 8) must be used for defining the logical disk in Unit 50 (Byte Stream).

NUMCHANNEL: Specifies the number of channels required for family communication. It depends entirely on the application environment used. It is an integer value. The maximum value of the parameter can be calculated using the formula:

NUMCHANNEL = (USRFAMILY - 2)

-> A smaller number of channels than the maximum indicated above, does not penalize application environments in the case of the same channel identifier being assigned to several families; this is the case, for example, of the Grandpa.

-> A larger number has the effect of a higher D.A.M. occupation for the allocation of the relative descriptors.

See Appendix A for memory allocation.

NUMPROGRAM: Specifies the number of programs which can be loaded into memory at the same time. It depends entirely on the application environments being used.

Warning: It is an integer value and usually twice NUMCHANNEL. It is generally greater than or equal to the number of processes specified by the parameter NUMPROCESS in Unit 1. A value of NUMPROGRAM equal to NUMPROCESS permits making a number of programs Loads followed by as many processing Starts.
It must, however, be stressed that the two parameters are unrelated: in one case the application may involve the loading in advance of n programs (by executing the Load primitives n times) in the same family, each Load followed by the sequence "Start, WaitResult, Unload", each time with a different progId input at the Start. In another case the application may use the sequence "Spawn, Load, Call".

-> A number too small for the requirements (number of Loads) of the application environments intended to be used prohibits their activation.

-> A number too large has the effect of occupying more D.A.M. area than is necessary for the allocation of the relative process descriptors.

See Appendix A for memory allocation.

FILENUMBER: This indicates the maximum number of active files (that is connected) that the File System must handle at the same time. The Memory Volume files are included in this number.

Each connected file has a certain number of internal files associated with it, depending on its type (byte stream, positional, keyed).

Appendix B gives the correspondence between user files and internal files, and shows which entries are made in which tables when a connection is made. The FILENUMBER parameter is also used by the File System to calculate how many Extent Tables are needed.

It must be remembered that the number of simultaneously active connections depends entirely on the application environment used - see Chapter 3, "Processes, Families, Channels, Programs and Files". Obviously, if a large number of files are active at the same time, the work load is increased.

Warning: It may be useful to note that this parameter is in no way related to the number of files declared in MKVOL, which determines the creation of the PDD table (a structure resident on the volume itself) and, therefore, the maximum number of files that can be generated in the volume.

FILENUMBER cannot be linked to the other parameter NUMSID of the same Unit (see following), as everything depends on how many, and which, files the application environment uses.

-> A number too small for the requirements (number of simultaneous Connects) of the application environments intended to be used prohibits their activation.

-> A number too large has the effect of occupying more D.A.M. area than is necessary for the allocation of the relative file descriptors.

See Appendix A for memory allocation.

NUMSID: Depends entirely on the application environments used. With this parameter the user defines the maximum number of 'sids' (internal structures of the F.S. corresponding one by one with the system-id generated with the Connect or Open operation) active at any moment in the system.

Warning: There is no link with the FILENUMBER parameter. The following considerations can be taken into account when evaluating the NUMSID parameter value:

1. A user may decide to make 'm' Connect, and subsequently (at any moment) 'n' Open.

2. The different types of files handled by the File System require a different number of 'sids', that is to say the user's 'm' Connect are translated into 'M' (Connect/Open) internals (see Appendix 'B').

Furthermore, remember that only the I/O on a file requires the Connect and the Open, other File System functions (for example GetType, GetAccessRight, ...) only require the Connect. To sum up, the only constraint is that:

(M + n) = NUMSID

-> Too small a number of 'sids' for the requirements (maximum number of system-ids that can be generated simultaneously) of the application environments intended to be used prohibits their activation.

-> Too large a number has the effect of occupying more D.A.M. area than is necessary for the allocation of the relative descriptors.

See Appendix A for memory allocation.

NUMSYSBUFF: Specifies the number of buffers in the system buffer pool destined for synchronous and asynchronous I/O operations.

Every I/O operation requires a number of linked buffers formed as follows:

#buffers = ((size of I/O data object): (buffer size [multifile of 512]))
+ 1 [for fractions of buffer size]

It is possible to define private pools or buffers reducing the need for large system buffer pools. See Chapter 3 "I/O Buffers".

-> A number too small will result in delay in making the resource available.

-> A bigger buffer size reduces the access to disk, as it permits the holding of a larger amount of data in memory; it also permits an increase in the number of operations to a driver (see the "...NUMOPERAT" parameters of Units 6, 11, 13, 14, 15, 20 and 21). There is, however, a limit to the size of the increase, due to the allocation of the necessary memory [#buffer * buffer size], which reduces the space of the Memory Volume.

See Appendix A for memory allocation.

SYSBUFFSIZE: Specifies the size of each buffer in the system buffer pool. It depends on the application environments used. It must be a multiple of 512.

When defining the size and number of system buffers the effects on the availability of memory space must be taken into consideration. See Chapter 3 "I/O BUFFERS".

Warning: It is interesting to note that this parameter together with the NUMSYSBUF parameter may reduce the memory volume size thus reducing the number and size of files which may be created in it.

-> The use of large buffers implies a large quantity of data in memory, they are particularly useful for applications having large sections of contiguous data to avoid data fragmentation.

-> Small buffers are more appropriate for systems in which small sections of contiguous data are transferred, allowing the same memory area to be used for different files and thus decreasing the level of swapping.

See Appendix A for memory allocation.

NUMPRIVBPOOL: Specifies the maximum number of private buffer pools which the user may define for synchronous and asynchronous I/O operation.

They do not cause a delay in release of resources (buffer), as is the case with system buffers, provided that this parameter and the successive NUMPRIVBUFF are correctly dimensioned.

-> Too small a value penalizes the applications programs in I/O, in that it causes a delay in release of the necessary pool.

-> Too high a value costs only in terms of greater allocation for the pool headers.

-> A correct value - able to avoid conflicts on the pool - depends on the maximum number of user, who, at any moment, are carrying out I/O operations using SetBufferOptions.

NUMPRIVBUFF: Defines the total number of buffers which the user intends to keep in the maximum number of pools already defined.

The cost of the buffer size itself is not of interest since they are allocated to free memory blocks via the SetBufferOptions primitive.

When calculating the total memory size on disk the following should be taken into account:

- System files. The size of these depend on the environment they are used for. The Shell contexts for example have a minimum of 236 bytes which may increase if external connections are made before running the programs and the number of contexts which are present concurrently depend on the way in which the system is used, that is number of work stations, ...

- The size of memory files requested by application programs.

- The private buffer pool mechanism.

See Chapter 3 "I/O BUFFERS".

Warning: This parameter must not be lower than NUMPRIVBPOOL, otherwise there will inevitably be a delay in the assignment of a buffer to a pool.

-> Too high a value costs only in terms of greater allocation for the buffer headers; allocation of memory takes place at run time.

-> Too small a value will incur a delay in the release of a buffer.

See Appendix A for memory allocation.

FILFORMEMVOL: Specifies the number of files in the memory volume. It depends on the application environments used. These comprise system and user files.

The following files are created by the system at system initialisation:

- /DEV directory: this contains the names of all the system devices.

- /CONTEXT$ directory: this contains the context files for programs.

- /TMP directory: this allows the user to create memory files.

- /IPL volume: this is used to contain the mounted IPL disk.

The /DEV directory contains the symbolic names of all system devices, used independently of the environment selected. For example, TTYA and TTYB are created for each terminal which opens a working session; a symbolic name is created for each disk used in the system, the memory disk (VD, that is Virtual Disk), the Streaming Cartridge Tape and magnetic tape and each system printer.

The /CONTEXT$ directory contains the context file created by the PMM to be passed to Grandpa; a context file is created for each process started.

Other files are created in the memory volume when environments are activated by Grandpa and the activities within them launched.

A context file must be created for each environment activated by Grandpa. The number of files implicitly created in the memory volume varies for different environments.

The Shell environment, for example, must have a context file for each instance which is activated, in the same way as each program which is activated from the Shell environment.

Each suspend operation causes the creation of a new context file.

The value of this parameter is thus dependent on three factors:

- the physical configuration of the system (the number of FDU, HDU and the mode in which they are configured, the number of terminals, tapes, ...)
- the intrinsic policy of environments which may be selected (Shell needs a different number of memory files to the TH)
- the number of files explicitly requested by the applications activated from the various environments.

The value of this parameter affects the memory space available for the user in that it is reduced when the number of files is increased.

The memory files used to describe the physical configuration of the system are at least 10 for a system having only an FDU, and no HDU, the value must be incremented if the devices available increase. When calculating the value of this parameter the number of files needed for the various environments must be added to the system files. See Chapter 3 "I/O BUFFERS".

O_T_SIZE: Specifies the number of entries which make up the Object Table. Each entry contains the identifier of the distributed system global object. See Chapter 3 "Servers in a Distributed System".

In a distributed configuration, the Object Table of the Name Server system also contains the entries of the Application/Terminal systems. When configuring a Name Server system, it is therefore suggested that a reasonably high value be specified for this parameter.

LOGHD: Links several physical hard disks together to make up a logical hard disk. The maximum number of hard disk units possible is 8.

The value of this parameter may be logically seen as an 8 by 8 matrix. Each row of this matrix represents a logical HDU. Each element in the row contains the number of a physical HDU. All the physical HDUs in a row make up the logical HDU, whose identifier is the first value in the row. The elements of the matrix are requested row by row starting from the top one.

0 indicates fields which are not used: the grouped units are recognised by the name of the first unit.

The following is an example of the parameter:

```
0 0 0 0 0 0 0 0
3 1 0 0 0 0 0 0
2 4 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

This structure represents a system having four physical hard disk units (1,2,3,4), these are grouped into 2 logical HDUs, the first of which comprises HDU units 3 and 1, the second 2 and 4. A logical hard disk takes the name of the first physical hard disk. In this case the two descriptors generated in the /DEV directory are called HD3 and HD2.

When the user has no hard disk the matrix includes 64 words set to 0.

Hard disk units connected to the SMD controller are numbered 1,3,5,7 instead of the normal numbering (1,2,3,4) which applies to other kinds of hard disks.

This parameter influences only the number of files that are created under the /DEV directory of the memory volume, each file created reduces the total number of files that may still be created on the memory volume by one. The creation of these files does not occupy any unnecessary memory area.

Warning: this parameter must be set as though only one hard disk were present when configuring a system. It may then be modified, if necessary, as described above (using the MODCON Shell command), but only after the LINKDEV Shell command has been used (see Part V).

See Appendix A for memory allocation.

NSERVERTYPE: Identifies the machine type. Possible values are:

2: if the Global Name Server is resident on the machine (in the case of a single network or a principal network)

1: if the the Domain Name Server is resident on the machine (this is the case of the machine which joins a subnetwork to a principal network)

0: for all the machines on which the Name Agent is resident.

See Chapter 3, "Limits to Software Configuration".

The two parameters that follow are only significant for interconnected local networks. They allow the system which is connected to both the subnetworks to know where its own Global Name Server resides.

GLOBNSNET: Only significant on machines where the Domain Name Server is resident (for example, NSERVERTYPE=1). It specifies the Network Name (see Unit 1) of the machine where the Global Name Server is resident.

GLOBNSHOST: Is significant on machines where the Domain Name Server is resident. It specifies the Host Name (see Unit 1) of the machine where the Global Name Server is resident.

FSERVERINST: Specifies the number of instances of the File System Server (FSERVER) that are to be activated in the machine. As each instance serves a File System request, the greater value this parameter has the more File System requests may be served in parallel.

The suggested values for this parameter are:

0:	if the files in this machine are not to be accessed by other machines in the network and is therefore incorrect for a distributed system.
1:	for all machines that do not have File Server functions.
other values:	for machines that have File Server functions and for which it is convenient to increase the parallelism in serving file access requests from remote systems. The optimal value of the parameter, as regards performance, is at present 3-4.

See Chapter 3, "Remote Access Components", for the components that use FSERVER.

Warning: The sum of FSERVER and GSERVER (see following) must not exceed the value of 15.

GSERVERINST: Specifies the number of instances of the Remote Procedure Call Server (GSERVER) that are to be activated on the machine. This depends on the number of services that reside on the machine.

If this parameter is increased the number of simultaneously executed services also increases. If there is no service on the machine, the parameter must be set to 0. Generally, in a distributed system, it has the value 1.

See chapter 3, "Remote Access Components", for the components that use GSERVER.

RFA_TIMEOUT: specifies the time (in seconds) that the RFA must wait at the private reception port. When this time has expired RFA signals timeout. The same value is used by FSERVER when the next blocks of a message are being received at the private port (see the manual MOS Structure and Functioning).

MULTPLFILES: Expresses the number of different multiple files to be concurrently connected in the system. It depend entirely on the application environments used. "Multiple" refers to the complete set of positional and keyed files regardless of their characteristics (record deletion, secondary indices, ...)

The following points should be noted:

- 1 positional file no deletion = 2 byte stream files
- 1 positional file yes deletion = 3 byte stream files
- 1 positional file packed = 3 byte stream files
- 1 keyed file = 3 byte stream files

Warning: The sum of all the various types of multiple files, at any moment, must not exceed the value of this parameter, otherwise a table overflow error will be signalled. It is stressed that this is not connected to the PACKPOSFILE and KEYEDFILES parameter of the same Unit (see following).

-> A number too small for the requirements of the application environments intended to be used prohibits their activation.

-> Too large a number has the effect of occupying more D.A.M. area then is necessary for the allocation of the relative descriptors.

See Appendix A for memory allocation.

PACKPOSFILES: It depends entirely on the application environments used. This parameter serves to dimension the Lock Table of the F.S., a table with a number of entries composed of:

PACKPOSFILES + KEYEDFILES + LOCKEDREC + LOCKEDPHAN

The Key Table of the F.S. has the same number of entries as the Lock Table.

Warning: The limit of this parameter is always represented by the MULTPLFILES parameter described above.

Example: If MULTPLFILES has the value 30 and there are no other types of multiple files, a user may therefore create up to 30 packed positional files.

See Appendix A for memory allocation.

<u>**KEYEDFILES:**</u> Specifies the number of keyed files. The considerations noted for PACKPOSFILES apply also to this parameter.

See Appendix A for memory allocation.

<u>**LOCKEDREC:**</u> Indicates the number of locked records which the user wishes to maintain concurrently in the system (regardless of the type of files).

See Appendix A for memory allocation.

<u>**LOCKEDPHAN:**</u> Specifies the maximum number of locked "phantom" or non-existent records (used for locking records in keyed files prior to assigning values to the record).

See Appendix A for memory allocation.

SECONDARYIND: Specifies the number of different secondary indices to be connected concurrently to the system. A connection with a secondary index is established whenever the keyed file to which the index belongs is connected.

See Appendix A for memory allocation.

$DEB (RAM DEBUGGER) PARAMETERS: UNIT 54

TERMNUM: Specifies the WS identifier that drives the debugger or a channel dedicated to a debugger terminal.

In the first case, a work station is defined to drive the debugger.

In the second case, a channel is dedicated to the debugger: that is, a VT100 type terminal can be physically connected to this channel for debugging purposes only.

It is a two byte hexadecimal value. The most significant byte can be:

%00 : if the work station is of the KDC type
%01 : if the work station is of the VT100 type connected via RS232, or if the channel dedicated to the terminal debugger is being configured

The least significant byte can be:

- for KDC, a value from %00 to %0F, which specifies one of the work station

- for VT100:

  %01 : if the work station is connected via RS of CPU

  %02 : if the work station is connected via the first channel of Twin 1

  %03 : if the work station is connected via the second channel of Twin 1

  %04 : if the work station is connected via the first channel of Twin 2

  %05 : if the work station is connected via the second channel of Twin 2

  and so on.

Warning: On L1 MOS multiprocessor systems, for VT100 like terminals the only admitted value is %0101, which means connection via RS of CPU; for KDC terminals the only admitted values are %000n, where 'n' is in the range from 0 to F.

KEYCODE: This parameter specifies the key that will initiate the debugger.

For KDC type work stations it must be the hexadecimal positional code of the key.

For VT100 type work stations it must be the ASCII code of the key.

The Keyboard Key position codes are given below; the key code are valid for all keyboard layouts and do not depend on the key level (key pressed with SHIFT or CONTROL).

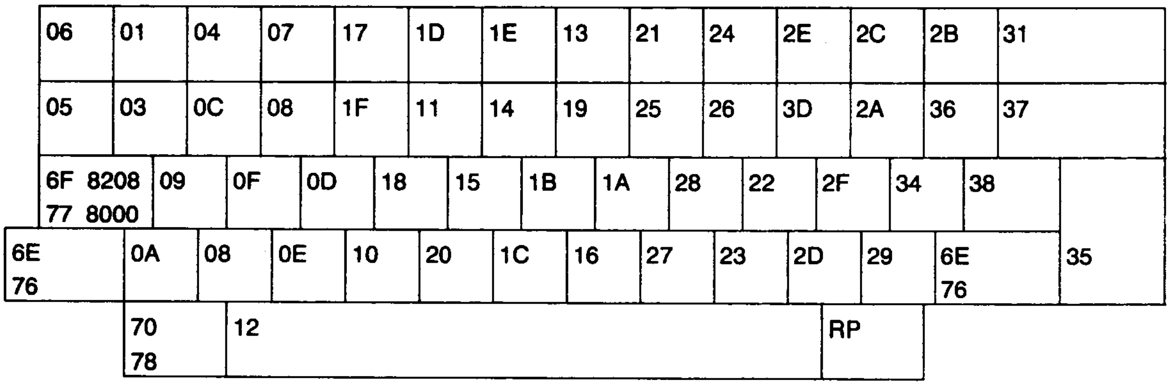


Fig. 4-5 Alphanumeric Area

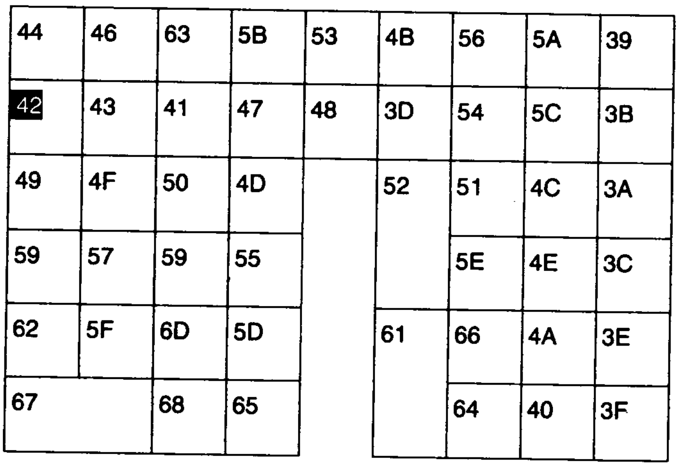


Fig. 4-6 Function Area

3

PART III : GUIDE TO THE GENERATION PROCESS

INTRODUCTION TO PART III

This part of the manual describes the procedures used to generate L1 MOS systems for customers with specific hardware installations.

It is divided into chapters on configuring the COBOL run time support (Chapter 5), compiling configuration tables (Chapter 6), and generating products for customers (Chapter 7).

5. RUN TIME COBOL MODULAR CONFIGURATION

A modular run time can be configured as an alternative to a standard COBOL run time, using the MCRTS (Modular COBOL Run Time System) tool.

Configuring the COBOL run time means the selection of optional modules. These are:

- SCREEN HANDLING
- SORT-MERGE
- REPORT WRITER
- EMULATED VISA

and they will be added to those which are always present (that is, NUCLEUS and I/O). Contents of these modules are listed below.

The following is required for use of MCRTS:

- An L1 MOS system configured with HDU's, FDU's and Shell
- Availability of OLINK
- Enough memory space to allow allocation of OLINK temporary files
- The variable %PATH set to the value '/IPL/DPC/CMD'

RUN TIME COBOL FUNCTIONS

Run time COBOL functions are the standard COBOL modules plus S6000 VISA.

These are as follows:

- NUCLEUS	contains the support for all "NUCLEUS MODULE" statements
- I/O	contains the support for all the "SEQUENTIAL MODULE", "RELATIVE MODULE", and "INDEXED MODULE" statements
- SCREEN HANDLING	contains the support for the "SCREEN HANDLING MODULE"
- SORT-MERGE	contains the support for the "SORT-MERGE MODULE"
- REPORT WRITER	contains the support for the "REPORT WRITER MODULE"
- EMULATED VISA	contains the support for the S6000 VISA

COBOL PROGRAM LINKING INDEPENDENCE FROM RUN TIME

With the COBOL run time modular configurator, programs do not have to be linked again after modifications have been made to run time, as all the references to run time entry points are expressed in a virtualized form, excluding only the following functions:

- PROGRAM STARTUP
- ACCEPT LINE-PARAM
- STOP RUN or STOP LITERAL
- CALL DATA-NAME WITH DYNAMIC LOADING
- LMS

Their presence in a program does not require it to be relinked.

Programs do not have to be linked again even if the run time is re-configured, as long as the new configuration is either the same as the old one, or contains it.

This facility is made possible by the mechanism m-tables/p-tables. A p-table is generated for each run time function, and this is linked at run time (see <u>PASCAL+ Program Preparation and Execution</u>).

RUN TIME COBOL STRUCTURE

This is as follows:

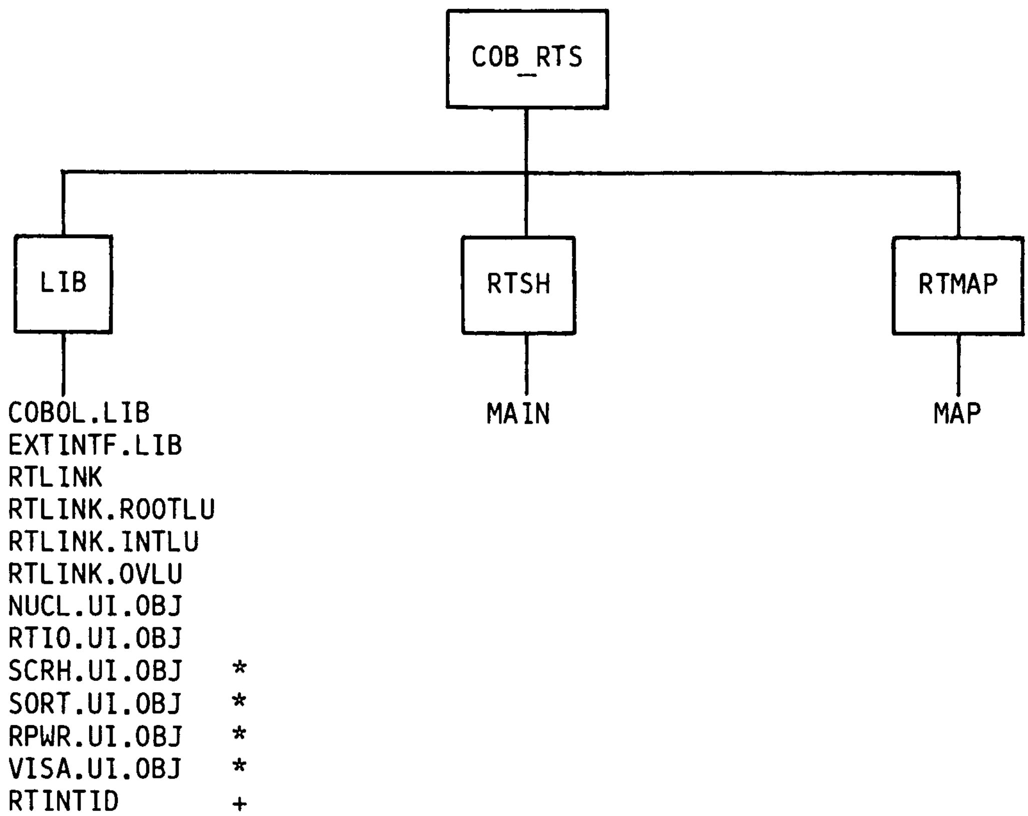


Fig. 5-1 Run Time COBOL Structure

The asterisked files are only present if requested in configuration.

The file marked with the plus (+) is the mask that identifies the configuration.

The directory LIB contains the files that are used in application program linking.

The directory RTSH contains the shared module (MAIN) that is used when the application program is run.

The directory RTMAP contains the run time linkage map; it is only generated in order to have a linkage map of the run time that has been requested.

The user is recommended to print this file for reference purposes.

The shared module (COB_RTS/RTSH/MAIN) in the RUN system must be consistent with the library version in the PROGRAM PREPARATION system (directory LIB); that is, the configuration of RUN must be the same as or contain the configuration of the PROGRAM PREPARATION system.

Consistency is controlled at run time, and checks are made to see that all the functions that have been called in the application programs have been configured.

The root directory of the product must have the path-name IPL/DPC/COB_RTS, therefore if the user wishes to keep the standard run time and/or previously generated run times he must copy them to other directories with different path names.

When a new version of run time is used the system must be initialised before executing any further COBOL programs, even if they are configured with the same functions.

COBOL RUN TIME CONFIGURATION COMPONENTS

These are as follows:

- Input files and libraries
- A generator of configuration control masks
- A tool configurator

INPUT FILES AND LIBRARIES

A library, a p-table that is linked to the run time and a user interface that is linked to the COBOL program, are supplied for each COBOL run time function.

These elements are shown in the following table, divided according to the function.

function	library	p-table	user interface
NUCLEUS	NUCL.LIB	NUCL.PTBL.OBJ	NUCL.UI.OBJ
I/O	RTIO.LIB	RTIO.PTBL.OBJ	RTIO.UI.OBJ
SCRH	SCRH.LIB	SCRH.PTBL.OBJ	SCRH.UI.OBJ
SORT	SORT.LIB	SORT.PTBL.OBJ	SORT.UI.OBJ
RPWR	RPWR.LIB	RPWR.PTBL.OBJ	RPWR.UI.OBJ
VISA	VISA.LIB	VISA.PTBL.OBJ	VISA.UI.OBJ

The following files which contain the commands for linking the run time program:

- LINK.BASE0
- LINK.BASE1
- LINK.SEGMSORT
- Working files for the configurator

and the following libraries, that are not configured, are also supplied:

- COBOL.LIB
- EXTINTF.LIB

CONTROL MASK GENERATOR

The control mask generator RTMAKEID, which is part of the configuration, is a program that generates the RTINTID mask that identifies the configuration.

This mask is linked both to the run time and to the application programs, and is used to carry out consistency checks between the RUN and PROGRAM PREPARATION versions of the run time program.

MAKECRTS CONFIGURATOR

The MAKECRTS configurator is used to create the required COBOL run time programs.

A SHELL configurator interprets the specified options, activates the RTMAKEID control generator, dynamically constructs a command file and starts-up OLINK. It also creates the directories for PROGRAM PREPARATION and RUN and the default command file RTLINK for linking the application programs.

The format is as follows:

MAKECRTS options INDIR = dirname1 OUTDIR = dirname2 COMMONDIR = dirname3

---

where:

- options are as follows:

  SCRH to include Screen Handling
  SORT to include Sort-Merge
  RPWR to include the Report Writer
  VISA to include S6000 VISA

  These are all optional and can be used in any order.
  If they are omitted the MINIMAL LIBRARY is configured which gives support for the NUCLEUS and I/O functions.

- dirname1 is the complete path name of the MCRTS components (that is /IPL/RELn.l/DPC/MCRTS , where 'n' is the release number and 'l' is the release level).

- dirname2 is the path name of the directory that identifies the product root (that is /IPL/FREE/DPC).

  This path name is only significant during the process of run time configuration. At program execution time the product root must have the value 'IPL/DPC/COB_RTS'.

- dirname3 is the path name of the volume where the common library LIB/SYS.LIB is (that is /IPL/RELn.l/DPC/COMMON , where 'n' is the release number and 'l' is the release level).

Note: Path names indicated above for dirname2 and dirname3 (/IPL/FREE/DPC and /IPL/RELn.l/DPC/COMMON respectively) should be used when generating within the user product.

Different path names, instead of these, should be used in order to generate a COBOL run time separate from the user product.

ERROR MESSAGES

At link time:

REFERENCE TO UNRESOLVED SYMBOL xxxxx -- <<0>> 0000 USED

No corresponding functions are in the PROGRAM PREPARATION library for one or more functions in the source library.
This message is contained in the map file produced by OLINK.

At run time:

INCONSISTENT RTSH MODULE VERSION

The result of the check carried out at run time, as to the consistency between the RUN and PROGRAM PREPARATION configurations, is negative.

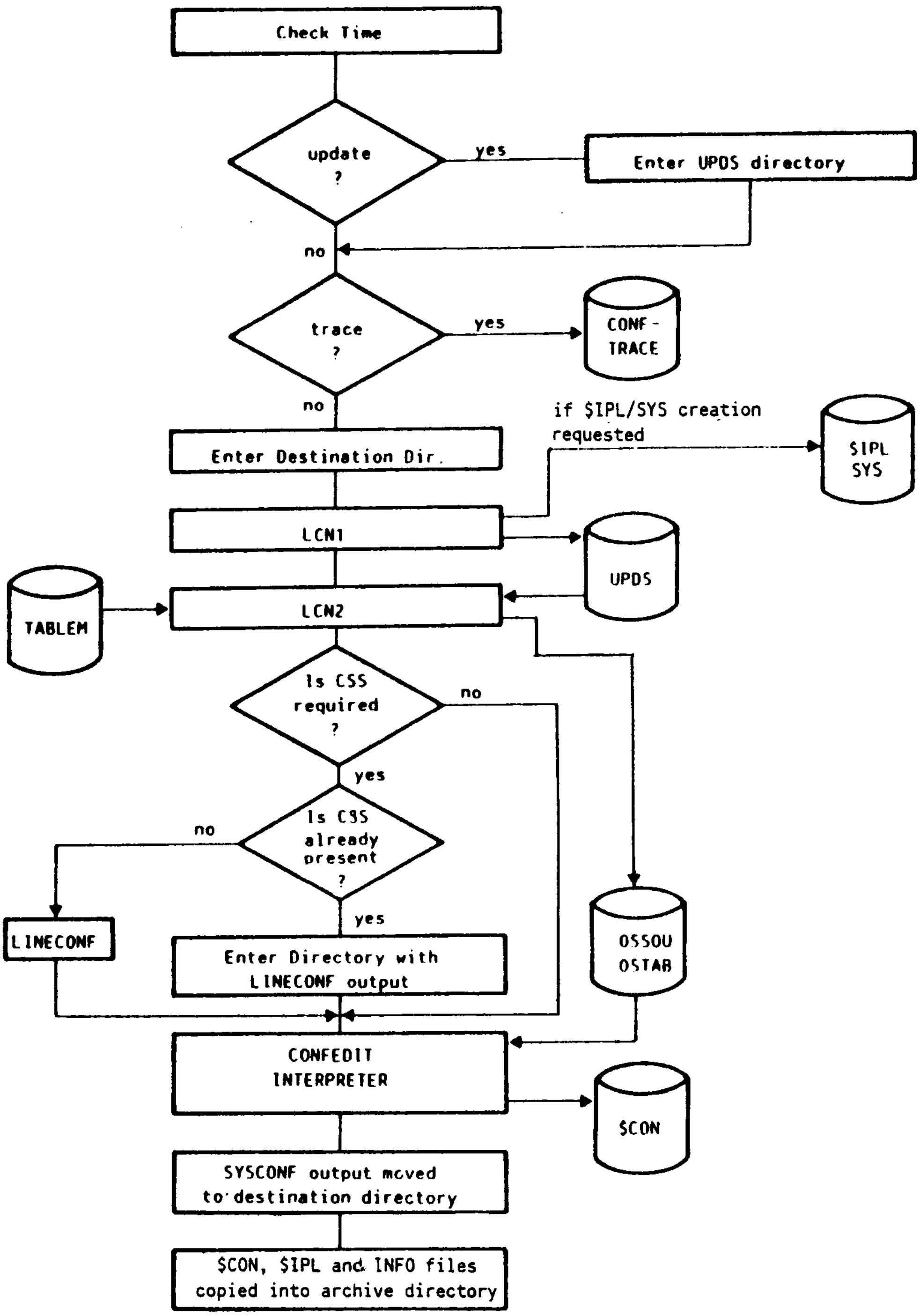


Fig. 6-1 SYSCONF Logic

6. THE SYSCONF PROGRAM

Before generating the SYS volume on hard disk which is to contain the customer's operating system, the configuration parameters for the system must be specified. These parameters are used at run time to initialise l-modules in the installed system. The parameters are specified, in the subsidiary company, using the SYSCONF command which:

1. asks the system timer to be set up (This is due to the fact that the last access time for some files is used by the makefile mechanism.)

2. asks for the destination user directory into which the configuration output will be placed (the specified directory must exist and be empty)

3. selects the create or the update function

4. if the selected function is update, asks the directory containing the UPDS file (for the configuration to be updated)

5. asks if the trace is required

6. creates the WORKFILE, FILEGEN and CUPCOM directories

7. starts the LCN1 program

8. asks if the CSS (Communication SubSystem) is present and starts, if needed, the LINECONF program

9. starts the LCN2 program

10. generates, if required, the $IPL l-module and/or the SYS volume

11. starts the generation phase of the configuration source file

12. starts the CONFEDIT program

13. starts the INTERPRETER, which produces the $CON file

14. moves all the created files ($CON, and possibly $IPL, CONFTRACE, ...) into the user defined directory.

15. copies the $CON and the $IPL files in the /IPL/RELn.1/ARCH_IPL_CHM/SC_LC/ARC_IPL_CON directory, together with a file which contains a brief description of the particular configuration. These three files have a common numeric suffix: for example, $CON_10, $IPL_10 and INFO_10.

Note: If the generation of the $IPL l-module is not required, it is the responsibility of the SYSCONF user to guarantee congruency between the

$CON generated using SYSCONF and the $IPL 1-module which will be used to initialize the system.

Chapter 4 is a detailed description of the operating system configuration parameters. Chapter 3 is a guide to configuration decisions.

The following figure shows the file system structure which sets up the SYSCONF program execution environment.

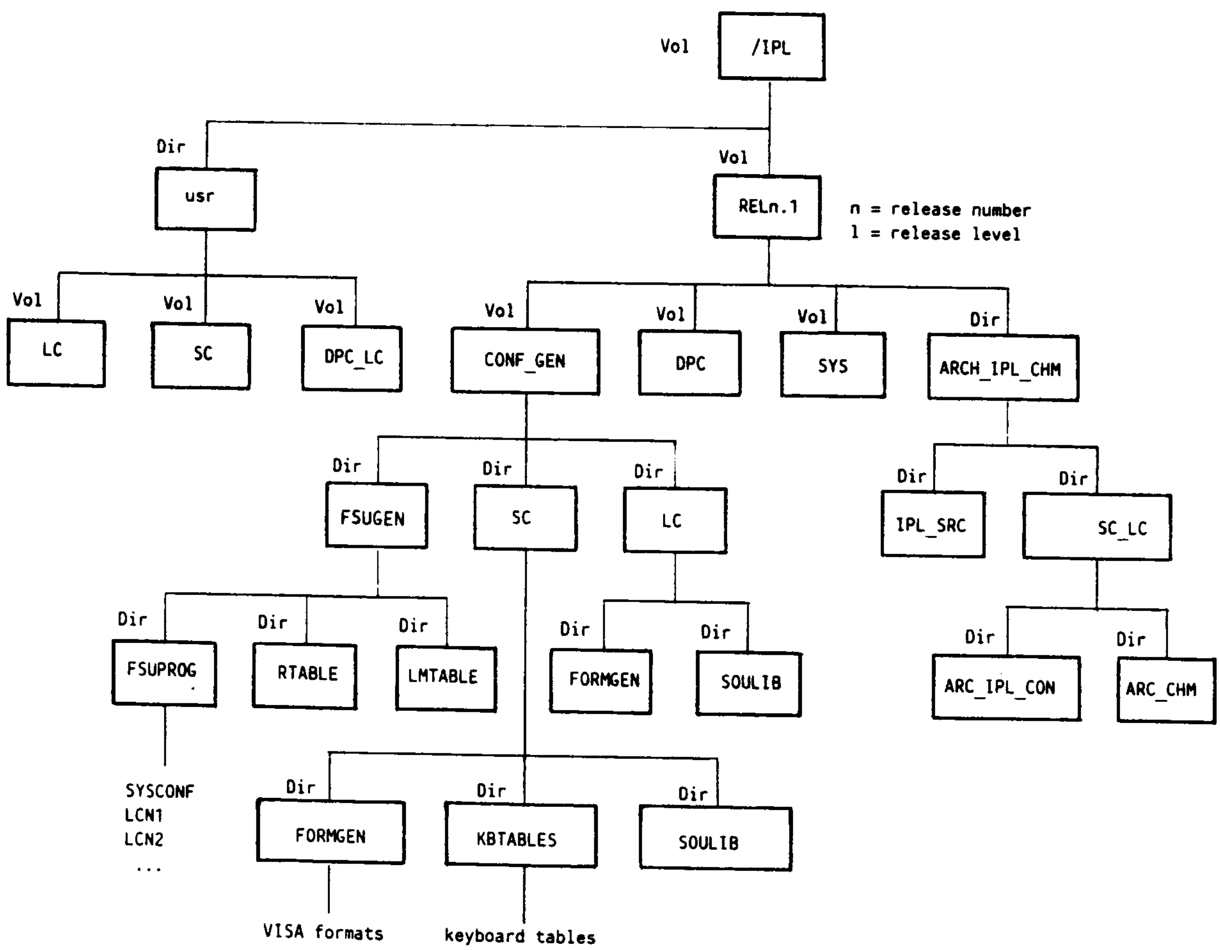


Fig. 6-2 SYSCONF Disk File System

EXECUTION

When the SYSCONF program is activated, an interactive session is started in order to:

1. give a general definition of the system to be configured, thus selecting the necessary Units for the required configuration and the command string for generating the $IPL file (LCN1 utility)

2. enter all the values for the parameters relevant to the Units selected in the previous phase (LCN2 utility).

When the user wants to update a configuration, he only needs to confirm the information which is not to be changed and enter the new information.

USER INTERFACE

To run SYSCONF, the working directory is set to the FSUPROG directory (that is, **SETWDIR /IPL/RELn.l/CONF_GEN/FSUGEN/FSUPROG** , where **n** is the release number and **l** is the release level) and the program SYSCONF is run (that is, **SYSCONF**).

Note: It is also possible to activate the SYSCONF program with a debug option (that is, **SYSCONF D**): in this case some extra files will be produced in output, apart from those normally created ($CON and UPDS). These files are:

OSSOU which contains the Units selected by the user and is the input file for the CONFEDIT program

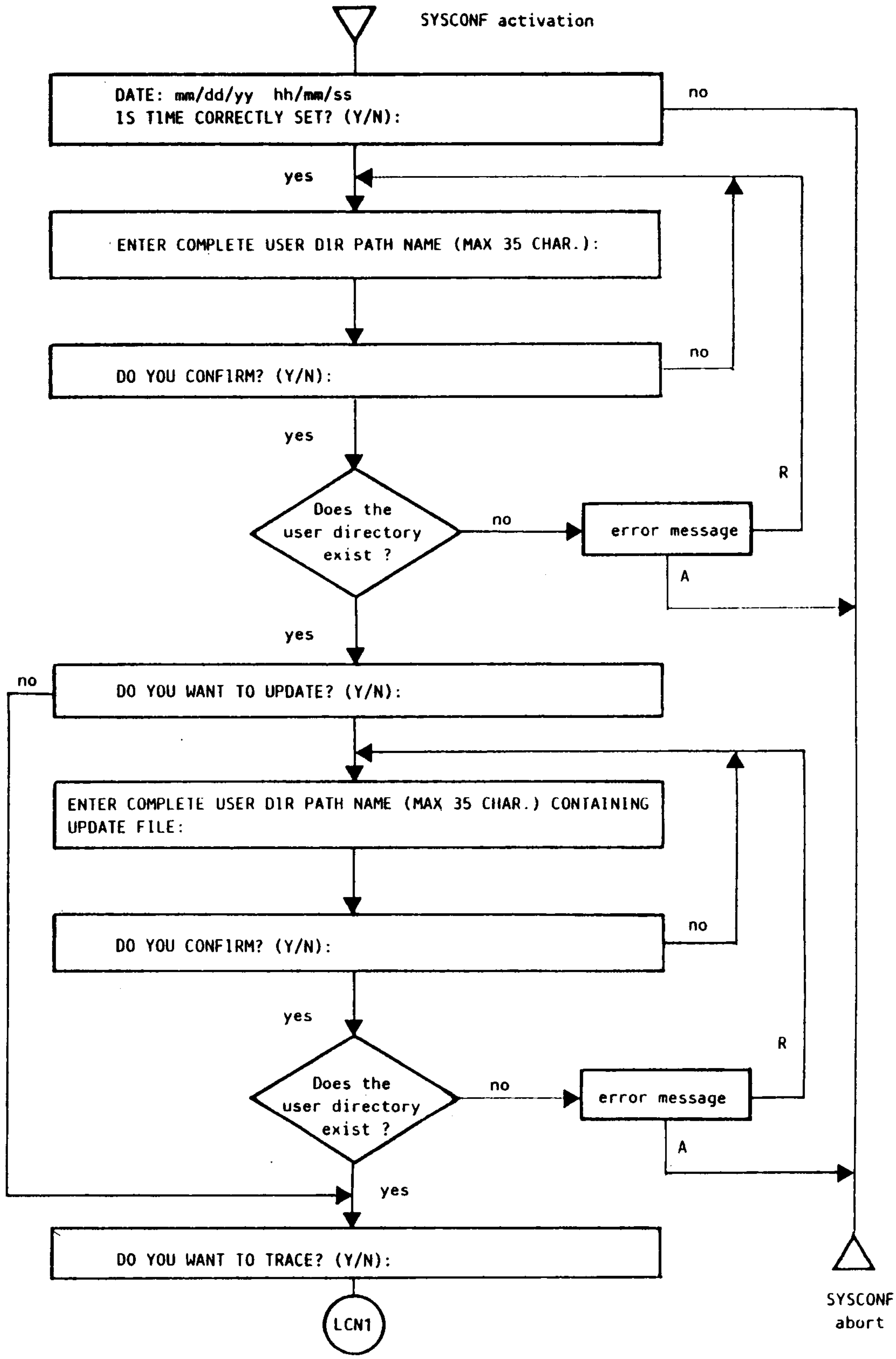
OSTAB which is the configuration parameters table, containing the values entered for each parameter. This is the input file for the batch phase which generates the file $CON

and, if generation of the $IPL module has been requested:

$IPL.map which is the map file produced by the "zloc" linker at $IPL creation time

The interactive part of the SYSCONF program (consisting of the LCN1 and LCN2 utilities) is a series of menus, which are preceded by the requests described below.

Flow of SYSCONF Messages Preceding LCN1



DATE: mm/dd/yy hh/mm/ss

IS TIME CORRECTLY SET? (Y/N):

Y	An affirmative response gives rise to the next prompt.
N	The SYSCONF program is aborted.

ENTER COMPLETE USER DIR PATH NAME (MAX 35 CHAR.):

/x..xx	The complete path name of the directory where the SYSCONF output files must be placed.

DO YOU CONFIRM? (Y/N):

Y	If the indicated directory does not exist a message is displayed and a new path name can be entered or the SYSCONF program can be aborted.
N	The path name is requested again.

DO YOU WANT TO UPDATE? (Y/N):

Y	The update function is requested, and the next prompt is displayed.
N	A new configuration is assumed to be carried out.

ENTER COMPLETE USER DIR PATH NAME (MAX 35 CHAR) CONTAINING UPDATE FILE:

/y..yy	The complete path name of the directory where the support file for the update function (UPDS) is contained.

DO YOU CONFIRM? (Y/N):

	Y	If the indicated directory does not exist a message is displayed and a new path name can be entered or the SYSCONF program can be aborted.
	N	The path name is requested again.

DO YOU WANT TO TRACE (Y/N):

	Y	A record of the configuration choices will be kept into the CONFTRACE file.
	N	The CONFTRACE file will not be created.

THE LCN1 UTILITY

The LCN1 utility prompts the user sufficiently to determine the software items that will be included in the $CON file. The items to be included determine which parameters will be requested by the LCN2 utility and the command string for the generation of $IPL.

The user does not directly select items, but specifies the required system configuration by means of a series of screen formats.

LCN1 produces a file (called UPDS) containing the list of Units to be included in the $CON file. The UPDS file is stored in the WORKFILE directory contained in the user defined directory specified as destination directory.

When a configuration is updated, the UPDS file generated in the previous configuration is used for identifying the default values of the screen formats.

When operating in **update** mode, on completion of a menu, the user may:

- enter **N** to the confirmation prompt and return to the first question of the menu, where the default values will be those entered during the previous SYSCONF session
- enter [↖] to the confirmation prompt and return to the first question of the menu, where the default values will be those entered during the previous SYSCONF session, but including possible modifications done in the current session.

The menus displayed by LCN1 are the following:

```
                    PRG : LCN1
                  OPERATING MODE

Do you want to generate SYS  volume.............(Y/N):

Do you want to generate $IPL module.............(Y/N):




                                         OK (Y/N):
                      SC100
```

where:

- the first answer is whether the SYS volume is to be created, or not, during the current SYSCONF session. The user is reminded that the SYS volume can be created:

  1. using the SYSGEN procedure described in Chapter 7

  2. during the SYSCONF session (answering **Y** to this question). This choice guarantees the absolute coherence among the $IPL, $CON, $CHM (if present) and the SYS volume.

- the second answer is whether the $IPL module is to be created or not.

Note: Neither the SYS volume nor the $IPL module for M31 systems can be generated using SYSCONF. The SYS volume for such systems must be generated using the SYSGEN procedure (as described in the "Generating Products for Customers" Chapter) and selecting the appropriate Functional Set.

```
          PRG : LCN1
         MACHINE TYPE

     1......M3X

     2......M30 COLOR GRAPHICS / M31

     3......M4X

     4......M54

     5......M64

     6......M60

     7......M60 MULTIPROCESSOR

     8......M70

     9......M70 MULTIPROCESSOR

     ENTER CHOICE :

                                    OK (Y/N):
                SC101
```

where the choice indicates the machine type for which the operating system is being configured.

PRG : LCN1
HARDWARE INFORMATION

HARD DISKS................(Y/N):	SCT5........................(Y/N):
NUMBER OF SASI CONTROLLER....:	FLOPPY DISKS................(Y/N):
NUMBER OF SMD CONTROLLER.....:	F.D. (1 MB,8")................:
NUMBER OF INT. CONTROLLER....:	M.F.D. (1 MB,5" OR OF SLIM)....:
NUMBER OF ESDI CONTROLLER....:	M.F.D. (320 KB,5")............:
NUMBER OF ST506 CONTROLLER...:	PERFORMANCE MONIT. EVAL. ...(Y/N):
MAGNETIC TAPE UNIT........(Y/N):	DO YOU USE PIN CHECK DRIVER.(Y/N):
	REDAC board...............(Y/N):
STREAMING CARTRIDGE TAPE..(Y/N):	DO YOU USE L.M.S............(Y/N):

OK (Y/N):

SC102

where:

- the first thirteen questions refer to the type of magnetic peripherals: if they are present (**Y**) or not (**N**), which controller handles them (for hard disks only) and how many units of each there are (for disks only)

- the last four answers indicate whether:

  . the performance evaluator program

  . the PIN Check driver (which also handles the Real Time Clock on M3X/M4X machines, that is allows the system time and date to be rembered when the system is switched off even if the REDAC board is not present)

  . the software handling the REDAC board (to handle the Real Time Clock on M64 systems)

  . the Local Management System

  must be included.

```
                         PRG : LCN1
              SYSTEM PERIPHERALS and CONTROLLERS


Number of WORK STATIONS.............:
Number of CAT work stations.........:   Number of BADGES................:
Do you use WORK STATIONS                Number of PINPADS...............:
       for dynamic login.......(Y/N):   Number of CASH ADAPTER..........:
Number of KDC  CONTROLLERS..........:   Number of MICRS.................:
Number of TWIN CONTROLLERS..........:
Number of MUX  CONTROLLERS..........:   Do you use GHAPHICS........(Y/N):
Do you use RSCPU...............(Y/N):   Do you use COLOR/GRAPHICS..(Y/N):

Number of SYSTEM PRINTERS...........:
Number of WS PRINTERS...............:   Number of RS232/CL DRIVERS......:

                                                              OK (Y/N):
                            SC103
```

where:

- the first answer indicates the number of work stations
- the second answer indicates the number of CAT (Customer Activated Terminal) work stations
- the third answer indicates whether work stations for dynamic login are to be handled or not (see the "Work Stations For Dynamic Login" section in the "Preparation of the MOS User Environment" Chapter)
- the following three answers indicate the number of some controllers (KDCs, Twin boards, MUXs)
- the seventh answer indicates whether the RS232 channel provided by the CPU board is to be used (for example, for a system printer) or not
- the last two answers in the left column indicate the number of system and work station printers, respectively
- in the right column, the answers indicate the number of some work station peripherals (badge readers, ...) and whether some facilities are used (graphics, colour); these answers are only meaningful if the $IPL module is being created
- the last answer indicates how many instances of the RS232/CL driver are required.

```
                    PRG : LCN1
                 OPTIONAL MODULES

POSITIONAL ACCESS METHOD....[$POS]..........(Y/N):

KEYED ACCESS METHOD.........[$KEY]..........(Y/N):

AUXILIARY FUNCTIONS.........[$AUX]..........(Y/N):

REMOTE SERVICE..............................(Y/N):

COMMUNICATION SUBSYSTEM.....[$CHM]..........(Y/N):


                                         OK (Y/N):
                      SC104
```

where:

- the first three answer indicate whether the positional access method, the keyed file handling and the auxiliary functions are required (**Y**) or not (**N**)
- the fourth answer indicates whether the Remote File Access facility is required (**Y**) or not (**N**). This feature must be selected for on-line systems (belonging to distributed configurations or clusters), but it is not necessary for stand-alone systems.
- the last answer indicates whether the CSS (Communication Subsystem) is required (**Y**) or not (**N**). Reply **Y** since every system (stand-alone or on-line) needs the Port mechanism to handle the master work station (also, the LINECONF program must be used to configure at least the UNIT 4, which refers to Ports). See also the "The LINECONF Program (Optional)" section below.

```
                         PRG : LCN1
                    SYS VOLUME GENERATION

  $VSH  and $CONVSH .......(Y/N):        VOLSR ...............(Y/N):
  $VISA and $VMN ..........(Y/N):        FLDUMP ..............(Y/N):
  $SERVER .................(Y/N):        FILETAR .............(Y/N):
  $FSERVER ................(Y/N):
  $GSERVER ................(Y/N):
  $BOOT_x .................(Y/N):
  $LOGN_...................(Y/N):
  $LOGS ...................(Y/N):


                                                   OK (Y/N):
                           SC104
```

where:

- answers on the left (above) indicate system component requirements (see the "Remote Access Component" section in Chapter 3)
- answers on the right (above) indicate save/restore command requirements.

THE LCN2 UTILITY

The LCN2 utility prompts the user to enter all the values for the parameters relevant to the selected configuration.

According to the information entered by the user when running the LCN1 utility, LCN2 calls VISA primitives to execute all the VISA forms related to each Unit involved. These forms have the following characteristics:

1. Four windows: one window containing a format header, two central windows for the questions (or one large window if more convenient), one "help" window for additional information.

2. Cursor positioned on current field.

The cursor is replaced at the beginning of the field if a unacceptable value is entered, and a new value can be keyed-in.

After completing a format, the user may:

- confirm all the data

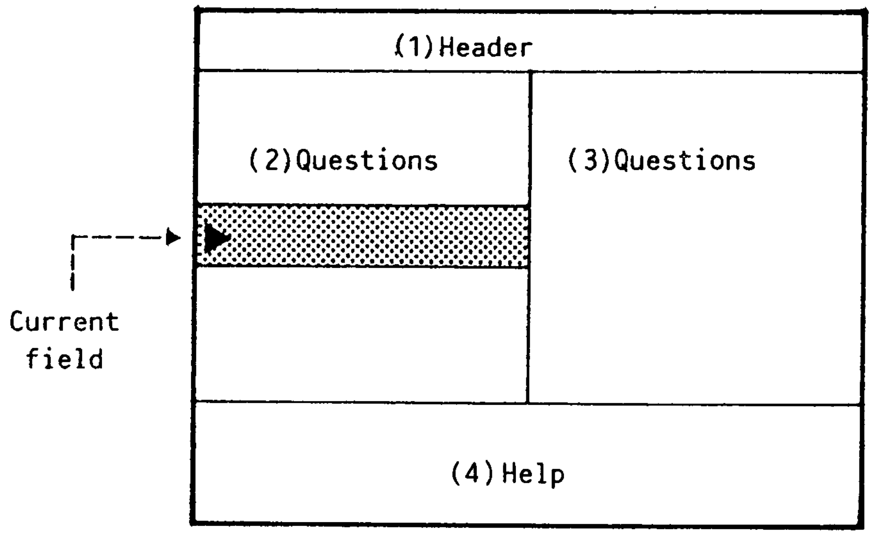
- display ranges for UNIT 5 values (PICK key). Such ranges are continuously updated according to the previously entered values.

- change some values, using the cursor

- abort the LCN2 (or LCN1) program (F8 key)

- exit from LCN2 (CTRL + Exit).

The raw data may be processed before being stored by the validation procedure associated with the form. Each form has a single procedure assigned to it, but one procedure may serve more than one form.



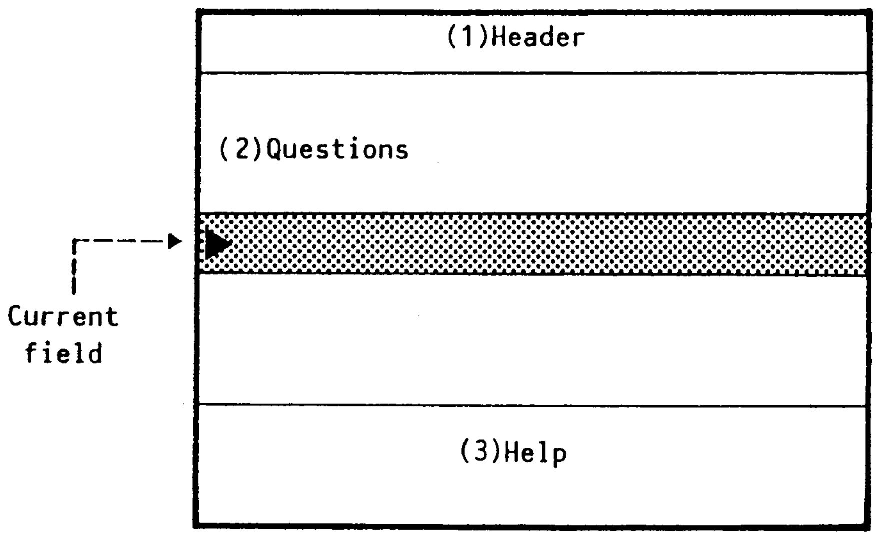


Fig. 6-3 Form Layouts

Besides the help displayed in each form, useful information and greater detail on the configuration parameters may be found in Part II of this manual.

Notes About LCN1 and LCN2 Internal Organization

Each Unit may have more than one format. Each format (previously created using TFORM) is stored in the /IPL/RELn.l/CONF_GEN/SC/FORMGEN directory ('n' is the release number and 'l' is the release level).

By means of these formats the user is prompted to enter values for all the parameters of the selected Units. Values entered are then associated (by VPL procedures) with the suitable variables.

If more than one format is relevant to a single Unit, they are called in the following sequence:

- The name of the first format to be executed is FORMxx0 (where "xx" is the number of the Unit to be configured, and is read in the UPDS file).

- The other formats are selected according to information contained in the FORMDATA buffer, which is written by the validation program common to all the formats associated with the Unit. The names of the formats associated with the Unit "xx" generally are:

  FORMxx1, FORMxx2, FORMxx3, ..., FORMxxn

The following table summarises the relationship between a Unit, its formats and its validation program.

UNIT NUMBER	NAME OF THE FORMATS	NAME OF THE VALIDATION PROGRAM
xx	FORMxx0, ..., FORMxxn	VPxx

Existing forms can be modified by the user (for instance, by translating the text of the questions). For a description of how to handle VISA forms and programs see <u>TFORM/VPL Form Preparation Tools, Reference Manual.</u>

THE LINECONF PROGRAM (OPTIONAL)

The LINECONF program must be used to configure:

- the Port mechanism (UNIT 4) for every system (never mind if on-line or stand-alone) in order to correctly handle the master work station

- the CSS for on-line systems.

After the LCN1 and LCN2 utilities, if the CSS (Communication SubSystem) is to be configured (that is, if the user had answered **Y** to the last question of the menu SC104 of LCN1), SYSCONF asks if the CSS has already been configured.

If the answer is affirmative, the user is prompted to enter the path name of the directory containing the output of a previously created CSS configuration.

If the answer is negative (and, therefore, the CSS is to be configured within the current SYSCONF session) then the LINECONF program is run. The complete description of how to configure the CSS using the LINECONF program is to be found in the CSS Generation and Configuration, User Guide.

If LINECONF does not terminate correctly, the user is asked whether he wishes to continue or not. He may decide to stop the whole generation activity (entering **N**), or to proceed and configure only the operating system (entering **Y**). In any case no output concerning the Communication Subsystem is produced.

PRODUCING THE $IPL L_MODULE AND THE SYS VOLUME

If it has been requested, the $IPL l-module and/or the SYS volume are now generated according to the configuration parameters given.

This generation phase is executed in batch mode, and a message is displayed when the batch execution is started.

The generated $IPL module is placed in the /IPL/RELn.l/ARCH_IPL_CHM/SC_LC/ARC_IPL_CON directory (n.l are the current release number and level), where the SYSGEN procedure expects it to be (see Chapter 7).

When producing the SYS volume, SYSCONF checks if all the required modules are present. If not, the message with which the user is notified about the SYSCONF operation (see below the "SYSCONF Output" section) will contain all the information necessary to regenerate the volume (see the REBUILD command described in Part V).

PRODUCING THE $CON FILE

Having so far received all the information relevant to the required system configuration, a non interactive phase is started, which consists of running the following programs:

- CONFEDIT

- INTERPRETER

The final output of this phase is the $CON file, which includes all the selected configuration parameters (including, if needed, those relevant to the CSS).

SYSCONF OUTPUT

After the correct execution of the SYSCONF program, the following files exist:

- $CON
- UPDS

plus, if the debug (D) option has been used at SYSCONF activation time (see the "User Interface" section above):

- OSSOU
- OSTAB

plus, if the trace function has been requested:

- CONFTRACE

plus, if its generation has been requested:

- $IPL (and $IPL.map if the debug (D) option has been used)

and plus, if its generation has been requested:

- SYS volume

These objects are stored in the directory specified by the user at SYSCONF activation time.

Their contents are summarized in the following table.

FILE	CONTENTS
$CON	Configuration file used by the system at IPL time
UPDS	Support file for the update function
CONFTRACE	Trace of the requests made to the user and of the answers which has been given
$IPL	Operating system l-module containing kernel and drivers
$IPL.map	Map file produced by the "zloc" linker at $IPL creation time
SYS volume	System volume generated according to the choices made by the user

Tab. 6-4 SYSCONF Output Files Contents

Warning: If the $IPL l-module and/or the SYS volume creation has been requested, the destination directory specified by the user must not be altered until the user has been notified of the end of the batch phase. The notification arrives in the form of a "wrtuser": "There is a message" is displayed on the system line, and information concerning the batch phase can be found in the .MESG file in the HOME directory of the user who ran SYSCONF.

$IPL IDENTIFICATION

In order to identify the $IPL modules generated by SYSCONF (which are placed in the /IPL/RELn.l/ARCH_IPL_CHM/~~IPL~~_SC_LC/ARC_IPL_CON directory, where n is the release number and l is the release level, and which are named $IPL_02, $IPL_03, and so on), the user can:

a) run the LISTMSG Shell command

b) run the CONFMAN utility (option 'L').

In this way their header message is listed to obtain information on their contents. The following table lists the characters which can be present in the header message and their meaning.

CHARACTERS	MEANING
B,	Badge reader driver
Ca,	Cash adapter driver
E,	Performance evaluation monitor
F,	Real time clock (REDAC)
Fb,	Floppy disk driver
Fm,	1 Mbytes mini floppy disk driver
Ft,	320 Kbytes mini floppy disk driver
Ha,	SASI 14 Mbytes hard disk driver
Hd,	ST 506 hard disk driver
He,	ESDI hard disk driver
Hg,	Hard disk (via GIPO) driver (no longer used)
Hi,	Integrated hard disk driver
Hs,	SMD hard disk driver
Id,	Color graphic physical driver
Ig,	Graphic work station physical driver
Im,	MUX physical driver
Ir,	CPU RS232 driver
It,	TWIN physical driver (SIC)
Iw,	Screen/keyboard physical driver
L,	Local Management System (LMS)
M,	Cheque reader driver
M48,	M54, M64, M70 and M70 multiprocessor machine support
M60,	M60 machine support
Or,	RS232 work station logical driver
Ot,	TTY like work station logical driver
Pl,	Printer driver
Pp,	PIN Pad driver
Rw,	Dynamic login support
S,	Streaming cartridge tape driver
S5,	SCT5 Streaming cartridge tape driver
T,	Tape unit driver
Wr,	RS232/CL driver (general driver)
Y,	PIN Check driver (encryption)
Z,	M60 multiprocessor support

Tab. 6-5 Meaning of the Characters in the Header Message of $IPL Modules

---

PATH NAME DOES NOT EXIST, RETYPE OR ABORT (R/A):

The directory specified as destination for SYSCONF output or as containing the UPDS file for an update session does not exist.

R The message requesting the path name is displayed again and a new path name can be entered.

A The SYSCONF program is aborted.

... LCN1 STATUS ERROR ... or

... LCN2 STATUS ERROR ...

The LCN1 (or LCN2) program has not been terminated correctly, due to an error in its execution or because it has been aborted by the user using the F8 key. No output files have been produced. The SYSCONF program must be run again.

ERROR IN EXEC LINECONF or

LINECONF STATUS ERROR

DO YOU WANT TO CONTINUE THE O.S. PARAMETER GENERATION? (Y/N)

The LINECONF program has not been terminated correctly, due to an error in its execution. No output files have been produced.

Y The SYSCONF session continues, but only output related to the operating system is produced.
No output concerning the Communication Subsystem is produced.

N The SYSCONF program is aborted.

Besides the above messages, other error conditions detected during execution of a Shell command invoked by SYSCONF (for example, COPY) may cause a Shell error message to be displayed. See the SHELL Commands, Reference Manual or the Message Book for their meaning.

7. GENERATING PRODUCTS FOR CUSTOMERS

The generation of a product on the subsidiary's machine consists of two operations:

1. Generating the system volume SYS, using the SYSGEN procedure.

2. Generating the DPC volume, using the DPCGEN procedure.

These operations are independent of each other. They may only be performed after the starter system and the Release Library have been installed on the subsidiary's machine, as described in the Release Guide, and after the tables containing configuration parameters have been compiled, as described in Chapter 6.

Before generating the product, a temporary volume called FREE must be created on the mostro to contain the product's SYS and DPC volumes.

Both the SYSGEN procedure for generating the operating system, and the DPCGEN procedure for generating the dependent components volume, are in the /IPL/RELn.l/CONF_GEN/FSUGEN/FSUPROG volume on the hard disk of the subsidiary's machine ('n' is the release number and 'l' is the release level).

The SYSGEN procedure includes in the system volume the $CON and the $IPL l-modules previously created using the SYSCONF program, and the $CHM l-module previously created using the LINECONF program.

CREATING THE TEMPORARY PRODUCT VOLUME 'FREE'

The following table shows how the FREE volume is created on the subsidiary's machine.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Create a temporary volume for the product (operating system and application environments). Its size and capacity depend on which functional sets (system and application) the product will contain.	MKVOL	
		FREE	volume name
		/IPL	directory name
		5000000	size in bytes
		10	number of files
		(typical size and no. of files; values set by user according to product characteristics)	
		/CR/	(user code not used)
		n	release number
		n	release level

Tab. 7-1 Creation of FREE Volume for Product

Note that the size of the volume given here is only an example.

SYSGEN: CONFIGURATION AND GENERATION OF SYSTEM VOLUME SYS

The SYSGEN procedure, which effects the generation of the customer's SYS volume, consists of creating the SYS volume containing the operating system for the customer.

It copies the system modules from the archive system volume to the user system volume.

If any sets of alternative modules are encountered while the modules are being copied into SYS, a list of the names is displayed and the user must choose only one of the alternatives. Similarly, if any optional module is encountered, its name is displayed and the user must decide whether to include it or not.

Note that if a user CHM module (previously created using the LINECONF procedure) is to be included on the volume SYS, then the module $BST_2 must be included. In this case the optional modules $CHM_1 and $BST_1 must not be included.

Note also that the choice of either the VOLSR module, the FILETAR module, or the FLDUMP module must always be made. Systems with SCT (and HD) need the VOLSR command, systems with MT (and HD) need the FILETAR command, whereas systems with HD in addition to FD need the FLDUMP command.

SYSGEN MESSAGES

To run SYSGEN, enter the Shell environment and type "SYSGEN". The messages displayed are listed and described below.

***** GENERATE SYS USER VOLUME *****

***** TYPE COMPLETE USER VOLUME PATH_NAME : (DEFAULT : /IPL/FREE)

Type the complete path name of the product image volume (/IPL/FREE).

***** THE VOLUME PATH_NAME IS : path_name *****

***** DO YOU CONFIRM (Y/N) ?

The path name of the product image volume typed in response to the last prompt is displayed. The user may type Y to confirm the name or N to select another.

***** VOLUME GENERATION IS : SYS *****

***** TYPE THE VOLUME SIZE:

The size of the system volume which contains the system l-modules must be input. This is usually set at just less than 1 Mbyte, the size of a floppy disk. However, if $CHM, $SERVER, ... are present (on line systems) a size greater than 1 Mbyte may be required.

***** THE VOLUME SIZE IS : size *****

***** DO YOU CONFIRM (Y/N) ?

The size of the system volume typed in response to the last prompt is displayed. The user may type Y to confirm the size or N to select another.

***** TYPE THE MAXIMUM NUMBER OF FILES :

***** DO YOU CONFIRM (Y/N) ?

The user is prompted to type in the number of files needed in the SYS volume, calculated from the functional set structure information given in the Release Guide and then to confirm the number in the same way as described above.

***** CREATED : /IPL/FREE/SYS *****

The /IPL/FREE/SYS volume has been created, and the selected system l-modules copied into it. (A list of these modules is also displayed.)

***** SELECT AN ALTERNATIVE MODULE *****

The list of $IPL modules present in the /IPL/RELn.1/ARCH_IPL_CHM/SC_LC/ARC_IPL_CON directory ('n' is the release number and '1' is the release level) is displayed on the screen.

***** TYPE SELECTION CODE TO SELECT :

The user must type the number corresponding to the required $IPL module.

***** SELECTED MODULE NAME IS : $IPL_x
***** DO YOU CONFIRM (Y/N)?

The name of the $IPL module which corresponds to the number typed in response to the last prompt is displayed. Type Y or N.

```
*****   SELECT AN ALTERNATIVE MODULE   *****

+-----------------------------------------------------------------------+
| SELECTION | SELECTED |                  MODULE                        |
|   CODE    |  MODULE  |                DESCRIPTION                     |
+-----------+----------+------------------------------------------------+
|    01     |  $LOGS   | $LOG with splitting login                      |
|    02     |  $LOGN   | $LOG without splitting login                   |
+-----------+----------+------------------------------------------------+

*****   TYPE SELECTION CODE TO SELECT :
```

The user must type the number corresponding to the required $LOG module.

```
*****   SELECT AN ALTERNATIVE MODULE   *****

+-----------------------------------------------------------------------+
| SELECTION | SELECTED |                  MODULE                        |
|   CODE    |  MODULE  |                DESCRIPTION                     |
+-----------+----------+------------------------------------------------+
|    01     |  $BST_1  | $BST without remote file access                |
|    02     |  $BST_2  | $BST including remote access to system objects |
+-----------+----------+------------------------------------------------+

*****   TYPE SELECTION CODE TO SELECT :
```

The user must type the number corresponding to the required $BST module.

```
*****   SELECTED MODULE NAME IS : $BST_x
*****   DO YOU CONFIRM (Y/N)?
```

The name of the $BST module which corresponds to the number typed in response to the last prompt is displayed. Type Y or N.

***** SELECT AN ALTERNATIVE MODULE *****

SELECTION CODE	SELECTED MODULE	MODULE DESCRIPTION
01	$BOOT_4	Bootstrapper for M30/M40
02	$BOOT_6	Bootstrapper for M60

***** TYPE SELECTION CODE TO SELECT :

The user must type the number corresponding to the required $BOOT_x module.

***** SELECTED MODULE NAME IS : $BOOT_x
***** DO YOU CONFIRM (Y/N)?

The name of the $BOOT module which corresponds to the number typed in response to the last prompt is displayed. Type Y or N.

***** SELECT AN ALTERNATIVE MODULE *****

SELECTION CODE	SELECTED MODULE	MODULE DESCRIPTION
01	$PMM_1	Process & Memory Management for M30/M40
02	$PMM_2	Process & Memory Management for M60 mono & M54/M64
03	$PMM_3	Process & Memory Management for M60 multiprocessor

***** TYPE SELECTION CODE TO SELECT :

The user must type the number corresponding to the required $PMM module, remembering that $PMM_1 or $PMM_2 can be selected if $BOOT_4 has been selected in response to the last prompt, while $PMM_2 or $PMM_3 can be chosen if $BOOT_6 has been selected in response to the last prompt.

***** SELECTED MODULE NAME IS : $PMM_x
***** DO YOU CONFIRM (Y/N)?

The name of the $PMM module which corresponds to the number typed in response to the last prompt is displayed. Type Y or N.

$CHM = OPTIONAL MODULE * YOUR ANSWER TO SELECT A MODULE TO COPY ? (Y/N)

Asks if you want to include in the system a $CHM module. Type Y or N.

***** SELECT AN ALTERNATIVE MODULE *****

The list of $CHM modules present in the /IPL/RELn.l/ARCH_IPL_CHM/SC_LC/ARC_CHM directory ('n' is the release number and 'l' is the release level) is displayed on the screen.

***** TYPE SELECTION CODE TO SELECT :

The user must type the number corresponding to the required $CHM module.

***** SELECTED MODULE NAME IS : $CHM_x
***** DO YOU CONFIRM (Y/N)?

The name of the $CHM module which corresponds to the number typed in response to the last prompt is displayed. Type Y or N.

---

module = OPTIONAL MODULE * YOUR ANSWER TO COPY (Y/N)?

Asks if you want to include in the system the optional modules shown in the prompts.
If $KEY is included, $POS is also included.
The VOLSR command is needed for systems which install from SCT; FLDUMP is needed for systems which install from FDU; FILETAR is needed for systems which install from MTU.
Type either Y or N in response to each question.
(See also the note on optional modules immediately before the description of this procedure.)

***** PROCEDURE CORRECTLY ENDED ***

At the end of the generation procedure, the /IPL/FREE volume on the hard disk of the subsidiary's machine contains:

- The SYS volume containing the selected modules

- The FSU directory containing the index for the modules in /IPL/FREE/SYS

DPCGEN: GENERATION OF THE DPC VOLUME

The generation of the DPC volume within the temporary volume FREE is managed by the DPCGEN procedure. This procedure can be executed without having first generated a SYS volume with SYSGEN.

DPCGEN MESSAGES

To run DPCGEN, enter the Shell environment and type DPCGEN. The messages which prompt the user during the execution of this procedure are shown below.

```
*****   GENERATE USER PRODUCT   *****
1.......DISPLAY FUNCTIONAL SET LIST
2.......RUN COPYENV
3.......EXIT FROM PRODUCT GENERATOR
*****   ENTER CHOICE :
```

The menu is displayed (see above) and the user must type a number from 1 to 3.

1 : A complete list is displayed of the functional sets available in the release for application environments and programming languages.

2 : The COPYENV utility is run. It takes one or more functional sets from the Release Library, one at a time, as selected interactively by the user, and copies modules into the customer system image volume (/IPL/FREE): the DPC volume is created and the modules belonging to the selected functional sets are copied into it.
If there are optional modules, the user is asked whether they should be included or not.

3 : The DPCGEN procedure is terminated.

***** TYPE FUNCTIONAL SET NAME :

Type the name of the functional set to be copied copied (one of the application functional sets, for example FS33).

***** DO YOU CONFIRM (Y/N) ?

Asks the user to confirm the data entered. Type Y or N.

***** TYPE COMPLETE USER VOLUME PATH_NAME :

Type the complete path name of the product volume (/IPL/FREE).

***** VOLUME GENERATION IS DPC *****
***** TYPE THE VOLUME SIZE :

Type the number of bytes needed in the DPC volume, calculated from the functional set memory occupation figures given in the Release Guide.

***** THE VOLUME SIZE IS : nnnn *****
***** DO YOU CONFIRM (Y/N) :

Asks the user to confirm the value entered (and by DPCGEN). Type Y or N.

***** TYPE THE MAXIMUM NUMBER OF FILES :

Type the number of files needed in the DPC volume, calculated from the functional set structure information given in the Release Guide.

***** DO YOU CONFIRM (Y/N) ?

Check the value entered then type Y or N.

***** CREATED : /IPL/FREE/DPC *****

The /IPL/FREE/DPC volume has been created and the selected 1-modules copied into it.

Note: A list of all the directories created in the DPC volume for the customer is displayed, and for each directory, a list of the 1-modules copied is displayed.

If there are optional modules in any selected functional sets, the user is asked in each case whether the modules should be copied or not.

***** DO YOU WANT TO COPY AN OTHER FUNCTIONAL SET?

Type Y or N. If the response is Y, the copying procedure (COPYENV) starts again from the beginning. Otherwise the next message is displayed.

***** END OF FUNCTIONAL SET GENERATION *****

After the last message, the initial menu of the procedure is displayed again. To exit from the procedure and return to Shell, select option 3 on this menu.

The resulting organization of the FREE volume on the mostro is shown in the figure below.

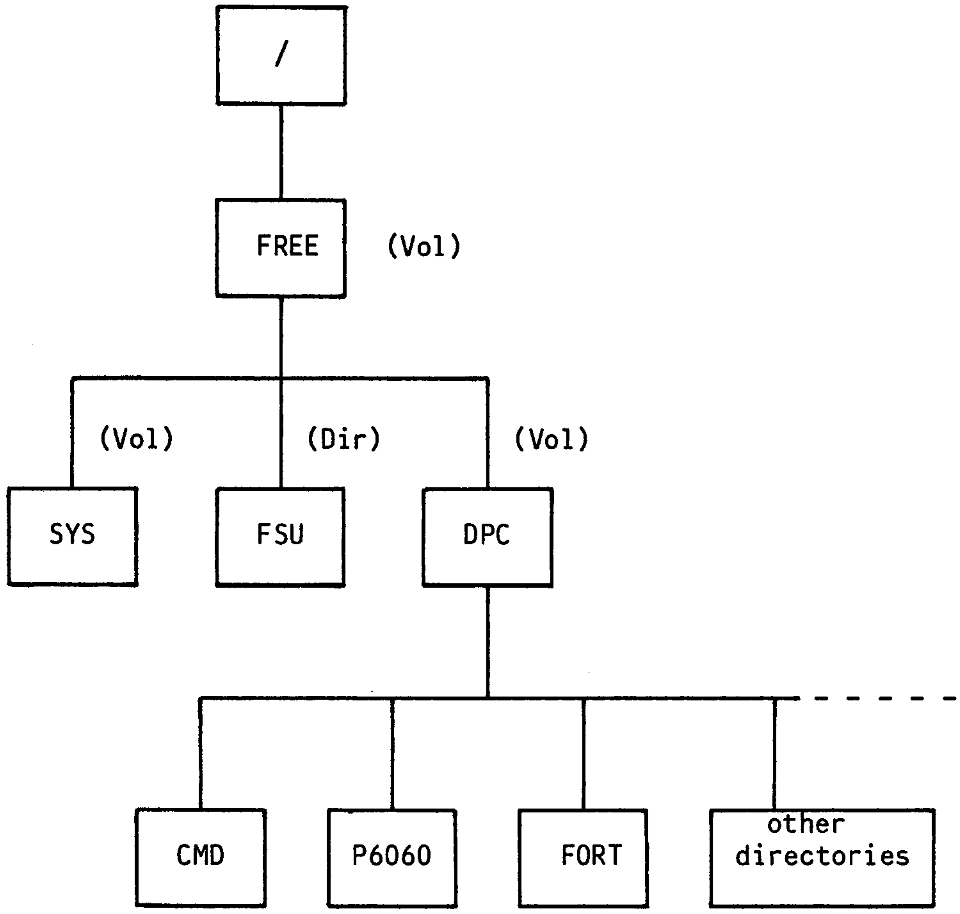


Fig. 7-2 Organization of the Customer System Image Volume

ERROR MESSAGES

When the user makes an error in entering parameters to SYSGEN or DPCGEN, no message is output, and control is automatically passed to the Shell environment. The user must then reactivate the program, and the interactive process begins again. If however, an error occurs during the execution of the COPYENV procedure, an error message may be output; these error messages are described here.

```
*****  user_path_name/SYS ALREADY EXIST   *****
*****  DO YOU WANT TO OVERRIDE IT? (Y/N)   *****
```

The user volume path name entered already exists. Type "Y" if it can be overwritten in the current SYS generation, or "N" if it cannot. In the latter case SYSGEN is ended, and the following message is displayed:

```
*****  END OF THE PROCEDURE  *****
```

ERROR IN PARAMETER(S) THE SYS VOLUME CANNOT BE CREATED

One or more mistakes have been made when giving values for the SYS volume size and maximum number of files.

ABNORMAL PROGRAM TERMINATION

COPYENV has terminated having carried out all its functions correctly, but it has not been able to delete its work files.

RELEASE PATHNAME INCORRECT

ERROR - (R = RETRY / A = ABORT)

The user has typed an incorrect release path name. If "R" is typed in response to the prompt shown above, the prompt

TYPE COMPLETE RELEASE PATHNAME :

is repeated. Otherwise, the program aborts, and control is returned to the Shell environment.

THE FOLLOWING DIRECTORY/VOLUME CANNOT BE SCRATCHED

When the volume has been created, and the operation of copying a new functional set has not succeeded, the volume SYS is not deleted.

THE FOLLOWING DIRECTORY/VOLUME DOES NOT EXIST

The directory or the volume does not exist in the release archive.

THE FOLLOWING DIRECTORY/VOLUME WILL BE SCRATCHED

When a fatal error occurs during the execution of COPYENV, COPYENV aborts, and the above message is output.

THE FOLLOWING FILE CANNOT BE COPIED

The file cannot be found in the archive.

THE FOLLOWING FILE CANNOT BE SCRATCHED

When COPYENV aborts, there may be files or program directories which may be unable to be cancelled due to a file system fault.

THE FOLLOWING VOLUME CANNOT BE CREATED

There is not sufficient space on the disk to create the volume.

WRONG, TRY AGAIN - (R = RETRY, A = ABORT)

When the operator makes a mistake in typing the release path name, or the user volume, COPYENV outputs this message. If the user selects retry, COPYENV repeats the prompt which was answered incorrectly. If the user selects abort, the following message is output:

FATAL ERROR , PROGRAM ABORTED

ONE OR MORE ERRORS DURING THE PROCEDURE

SEE /IPL/TRACE

One or more modules selected in the SYSGEN procedure has not been found or has not been copied successfully (for example, due to lack of space). Details about the failed operation are stored in the /IPL/TRACE file.

4

PART IV : SYSTEM DELIVERY, INSTALLATION AND USE

INTRODUCTION TO PART IV

This part of the manual concerns the delivery of systems to customers, the installation process and the necessary preliminaries for the use of L1 MOS systems.

Chapter 8 describes the preparation on the mostro of distribution kits containing customer systems to be delivered.

Chapter 9 describes the installation of customer systems on hard disk, using distribution kits.

Chapter 10 describes the preparation of configuration files for Grandpa (user environment initialisation process), lists the configurable application environments, and discusses system startup.

8. PREPARATION OF CUSTOMER DISTRIBUTION KITS

Customer distribution kits may be prepared for hardware installations having:

- floppy disks only (8" or 1 Mb 5 1/4"; 320 Kb 5 1/4" for M31 systems only)
- floppy disks and one or more hard disks
- streaming cartridge tapes and one or more hard disks
- magnetic tape and one or more hard disks.

FD-Only Systems

The number of floppy disks in a customer kit for a FD-only system depends on the size and nature of the MOS software selected. If the customer has a machine with few floppy disk drives you may wish to make use of the removable system disk facility. For small systems, all the software may fit on a single floppy disk.

Customer Distribution Kits

There are four different types of kit, classified according to the steps taken to prepare them. These are:

Type 1 : Kit for FD-only systems
Type 2 : Kit for FD-only systems intending to make use of the
         removable system disk facility
Type 3 : Kit for FD/HD systems
Type 4 : Kit for SCT/HD systems
Type 5 : Kit for MT/HD systems.

The starting conditions, file organization and generation of each type of kit are described separately in this chapter. For this purpose, the release library is assumed to be directly under the /IPL volume (it could have been placed anywhere in the file system). In all cases preparation starts with the subsidiary's machine logged in under the ROOT user and the console switch at PRIMARY.

The installation commands INSTALLQM, MKBOOT, MKLOGIN and MKSYS are described in detail in Part V of this manual.

Note

In the following text the use of 8" floppy disks is described. If 5 1/4" are used then the device name MF1 must be given in place of FL1.

STANDARD KIT FOR FD-ONLY SYSTEMS

The volume on hard disk from which the distribution kit is generated, /IPL/FREE, must contain:

- the system volume, SYS
- the dependent components volume, DPC, containing selected working environments, ...
- a reduced command volume, CMD, created from the "reduced Shell commands" functional set if needed.

The kit will consist of at least **one or two floppy disks** as follows:

first floppy disk : SYS volume;

USR and ETC directories (from MKLOGIN);

CMD volume containing the reduced set of commands:
MNT, UNMNT, COPY, SHDIR, PRY, MKVOL
(if dependent components on another disk)

or DPC volume containing dependent components;

$QM and BE directories, created by INSTALLQM (if batch environment is present);

SP directory, created by SPCONF, (if spool environment is present).

second floppy disk (if present) : dependent components.

The steps taken to prepare a standard floppy disk kit are listed in Tab. 8-1.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Insert the first floppy disk in drive 1.	-	
2	Prepare the disk for use.	MKENV FL1	
		Y	confirmation to clear disk
3	Create a volume on the disk (entire disk).	MKVOL	
		/CR/	volume name=default, VOL
		FL1	device name (assume 1)
		/CR/	default values for size
		/CR/	and number of files
		/CR/	(user code not used)
		n	release number
		/CR/	(release level not used)
4	Copy the system bootstrapper onto the disk.	MKBOOT /IPL/FREE/SYS FL1	
5	Mount the floppy disk under the root directory.	MNT FL1 /VOL	
6	Copy the system volume and the reduced command volume (if present) into the floppy disk volume.	COPY /IPL/FREE/SYS /VOL/SYS COPY /IPL/FREE/CMD /VOL/CMD	
7	Activate the user login mechanism for the system on floppy disk.	MKLOGIN /VOL	
8	Copy the Shell initial procedure file into the new root user's directory from the release library on hard disk.	COPY /IPL/REL5.2/DPC/CMD/.INIT /VOL/USR/ROOT/.INIT	

Tab. 8-1 Preparation of FD Kit on Subsidiary's Machine (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS
9	Install the spooling system, if required, using SPCONF (I option).	See Part V
10	Install the batch processing system if required.	INSTALLQM /IPL/REL5.2/DPC/SERVICE /VOL TYPE=BE ["R] (omit "R option if spooling already installed)
11	Unmount the floppy disk.	UNMNT /VOL
12	Remove the floppy disk and label it.	-
13	If kit contains a second floppy disk, insert the next disk.	-
14	Prepare the next disk for use.	MKENV FL1 Y confirmation to clear disk
15	Create a volume on the disk (entire disk).	MKVOL parameters as in Step 3
16	Mount the disk.	MNT FL1 /VOL
17	Copy selected dependent components from volume /IPL/FREE/DPC (contents are configuration-dependent) into /VOL	for example, COPY /IPL/FREE/DPC/DMS /VOL/DMS (Elements are copied individually so that DPC is not on the disk; then the disk can be mounted as /IPL/DPC)
18	Unmount the disk.	UNMNT /VOL

Tab. 8-1 Preparation of FD Kit on Subsidiary's Machine

REMOVABLE SYSTEM DISK KIT FOR FD-ONLY SYSTEMS

The /IPL/FREE volume on hard disk must contain:

- the system volume SYS
- the dependent components volume DPC containing selected working environments, ...
- a reduced command volume, CMD, from FS01 if needed

The kit will consist of **at least two floppy disks** as follows:

first floppy disk : system volume SYS;

second floppy disk : USR and ETC directories (from MKLOGIN);

either dependent components
or (reduced) command volume CMD;

$QM and BE directories, created by INSTALLQM (if batch environment is present);

SP directory, created by SPCONF, (if spool environment is present).

third floppy disks (if present) : dependent components

Note: At system startup time, in $CONFGP, the NEWVOL function is used when replacing the system disk (first FD in the kit) by the second floppy, see Chapter 10.

The steps taken to prepare a removable system disk kit are listed in Tab. 8-2.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Insert a floppy disk in drive 1.	-	
2	Prepare the floppy disk for use.	MKENV FL1	
		Y	confirmation to clear disk
3	Create a volume on the disk (entire disk).	MKVOL	
		/CR/	volume name=default,VOL
		FL1	device name (assume 1)
		/CR/	default values for size
		/CR/	and number of files
		/CR/	(user code not used)
		n	release number
		/CR/	(release level not used)
4	Copy the bootstrapper from hard disk onto the floppy disk.	MKBOOT /IPL/FREE/SYS FL1	
5	Mount the floppy disk under the root directory.	MNT FL1 /VOL	
6	Copy the system volume onto the disk.	COPY /IPL/FREE/SYS /VOL/SYS	
7	Unmount the first floppy disk.	UNMNT /VOL	

Tab. 8-2 Preparation of Removable System Disk Kit on Subsidiary's Machine (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS
8	Remove the first floppy disk and label it; insert another disk.	-
9	Prepare the second disk for use.	MKENV FL1 Y confirmation to clear disk
10	Create a volume on the disk (entire disk).	MKVOL same parameters as in Step 4
11	Mount the floppy disk under the root directory.	MNT FL1 /VOL
12	Activate the user login mechanism for the system on floppy disk.	MKLOGIN /VOL
13	Copy the Shell initial procedure file into the new root user's directory from the release library on hard disk.	COPY /IPL/REL5.2/DPC/CMD/.INIT /VOL/USR/ROOT/.INIT
14	Copy the dependent components volume or the command volume onto the floppy disk.	COPY /IPL/FREE/DPC /VOL/DPC or COPY /IPL/FREE/CMD /VOL/CMD

Tab. 8-2 Preparation of Removable System Disk Kit on Subsidiary's Machine (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS
15	Install the spooling system, if required, using SPCONF (I option)	See Part V
16	Install the batch processing system, if required.	INSTALLQM /IPL/REL5.2/DPC/SERVICE /VOL TYPE=BE ["R] (omit "R option if spooling already installed)
17	Unmount the floppy disk and label it.	UNMNT /VOL
18	If another disk is required, insert and prepare another disk.	MKENV FL1 MKVOL parameters as in Step 4
19	Mount next disk and copy dependent components onto it (separately, as disk to be mounted into /IPL/DPC.)	MNT FL1 /VOL for example, COPY /IPL/FREE/DPC/BASIC /VOL/BASIC
20	Unmount the disk.	UNMNT /VOL

Tab. 8-2 Preparation of Removable System Disk Kit on Subsidiary's Machine

KIT FOR FD/HD SYSTEMS

The /IPL/FREE volume on hard disk must contain:

- the system volume SYS, also containing the utility FLDUMP
- the dependent components volume DPC.

The kit will consist of **at least two floppy disks** as follows:

first floppy disk : system volume SYS;

a CMD directory containing the commands needed for installation;

the USR and ETC directories (from MKLOGIN);

second to nth floppy disk : dump of DPC volume.

The steps taken to prepare a FD/HD kit are listed in Tab. 8-3.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Insert a floppy disk in drive 1.	-	
2	Prepare the floppy disk for use.	MKENV FL1	
		Y	confirmation to clear disk
3	Create a volume on the disk (entire disk).	MKVOL	
		/CR/	volume name=default, VOL
		FL1	device name (assume 1)
		/CR/	default values for size
		/CR/	and number of files
		/CR/	(user code not used)
		n	release number
		/CR/	(release level not used)
4	Copy the bootstrapper onto the floppy disk (from the starter OS on hard disk).	MKBOOT /IPL/FREE/SYS FL1	
5	Mount the floppy disk under the root directory.	MNT FL1 /VOL	
6	Copy the system volume onto the disk.	COPY /IPL/FREE/SYS /VOL/SYS	
7	Create a command directory on the floppy disk.	MKDIR /VOL/CMD	
8	Set the working directory to the commands directory in the release library.	SETWDIR /IPL/REL5.2/DPC/CMD	
9	Copy the Shell commands needed for installation into the directory created in Step 7.	COPY MKVOL MKBOOT MNT COPY MKLOGIN UNMNT /VOL/CMD	

Tab. 8-3 Preparation of FD/HD Kit on Subsidiary's Machine (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
10	Reset working directory to the root user of the hard disk system.	SETWDIR /IPL/USR/ROOT	
11	Activate the user login mechanism for the system on floppy disk.	MKLOGIN /VOL	
12	Unmount the floppy disk.	UNMNT /VOL	
13	Remove the floppy disk, label it and insert another.	-	
14	Dump the dependent components onto floppy disks, labelling the disks with sequence numbers from 2 as you remove them.	FLDUMP	
		S	function is "save"
		/IPL/FREE/DPC	disk volume path name
		/CR/	default FDU: FL1
		/CR/	confirm floppy disk inserted
		N	no check
		/CR/	confirm next floppy disk inserted
		...	

Tab. 8-3 Preparation of FD/HD Kit on Subsidiary's Machine

KIT FOR SCT/HD SYSTEMS

The /IPL/FREE volume on hard disk must contain:

- the system volume SYS, also containing the command VOLSR
- the dependent components volume DPC.

The kit will consist of **two streaming cartridge tapes** containing the system volume SYS and the dependent components volume, DPC, respectively.

The steps taken to prepare a SCT/HD kit are listed in Tab. 8-4.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Insert an SCT.	-	
2	Prepare the SCT; it then contains the SCT bootstrapper the system bootstrapper and the system volume (SYS).	MKSYS SCn	n = 4 for SCT4; n = 5 for SCT5 (see MKSYS in Chapter 11)
		"owner-name"	enter information
		"date"	enter information
		nn	release number
		nn	release level
		Mx0	machine type
		/IPL/FREE/SYS	system volume name
		/CR/	DPC volume name; must be /CR/ if the DPC volume is to be stored on a second tape; it can be the DPC volume name if the DPC volume is to be stored on the first tape, together with the SYS volume (in this case they cannot exceed the tape capacity, 19 Mbytes, and the following steps 3, 4 and 5 are omitted)
		S or D	single or double recording (only for SCT4)
		Y	confirmation to continue
		Y	request to erase tape
3	Remove the first SCT and insert another.	-	
4	Prepare the second tape and dump the dependent components onto it.	VOLSR FUNC=LABEL	TID=DPC OWN=xxxxx TAPEMODE=S RECMODE=D
			function is "labelling"
			xxxxx is the owner identifier
			single-tape volume
			double recording
		VOLSR FUNC=SAVE	TID=DPC SRC=/IPL/FREE/DPC
			function is "save"
			/IPL/FREE/DPC volume is dumped on the DPC tape
5	Remove the second SCT.	-	

Tab. 8-4 Preparation of SCT/HD Kit

KIT FOR MTU/HD SYSTEMS

The /IPL/FREE volume on hard disk must contain:

- the system volume SYS, also containing the command FILETAR
- the dependent components volume DPC.

The kit will consist of **two magnetic tapes:**

first MTU : the system volume SYS

second MTU : the dependent components volume DPC

The steps taken to prepare a kit on tape for an MTU/HD system listed in Tab. 8-5.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Insert an MTU.	-	
2	Prepare the first MTU; it then contains the MTU bootstrapper, the system bootstrapper and the system volume (SYS).	MKSYS MT1	
		"owner-name"	enter information
		"date"	enter information
		nn	release number
		nn	release level
		/IPL/FREE/SYS	system volume name
		/CR/	default no. of files: 1000
		/CR/	DPC volume name; must be /CR/
		/CR/	default unit number: 1
		Y	confirmation
3	Remove the first MTU and insert another.	-	
4	Prepare the second tape and dump the dependent components onto it.	FILETAR	
		C	function is "create"
		/CR/	default drive number: 1
		Y	confirmation
		xxxxx	owner identifier
		FILETAR	
		S	function is "save"
		/IPL/FREE/DPC	volume to be dumped
		/CR/	default drive number: 1
		/CR/	default block size: 2048
		/CR/	no options
5	Remove the second MTU.	-	

Tab. 8-5 Preparation of MTU/HD Kit

9. INSTALLATION PROCESS

Installation of the customer system is only necessary when the system has to be loaded onto hard disk. The installation process varies according to the medium on which the product is delivered, which is one of the following:

- Floppy disks (8" or 1 Mb 5 1/4"; 320 Kb 5 1/4" for M31 systems only)

- Streaming cartridge tapes

- Magnetic tapes

For systems with floppy disks only, the product will be delivered on floppy disks which are ready for direct use. The three installation processes are described separately in this chapter.

The resulting structure of files on the customer's machine varies only in as much as there are two possible positions for CMD. The generic structure after the system has been installed is shown in Fig. 9-1.

The installation commands INSTALLQM, SPCONF, MKBOOT and MKLOGIN are described in detail in Part V of this manual. FILETAR is described in the Shell Commands, Reference Manual.

Note that:

- The spooling system is needed if a terminal emulator is present in the configured customer system.

- A further installation process is necessary for BASIC ESE (emulated P6066 BASIC), see L1 MOS Extended System Environment, Software Installation User Guide.

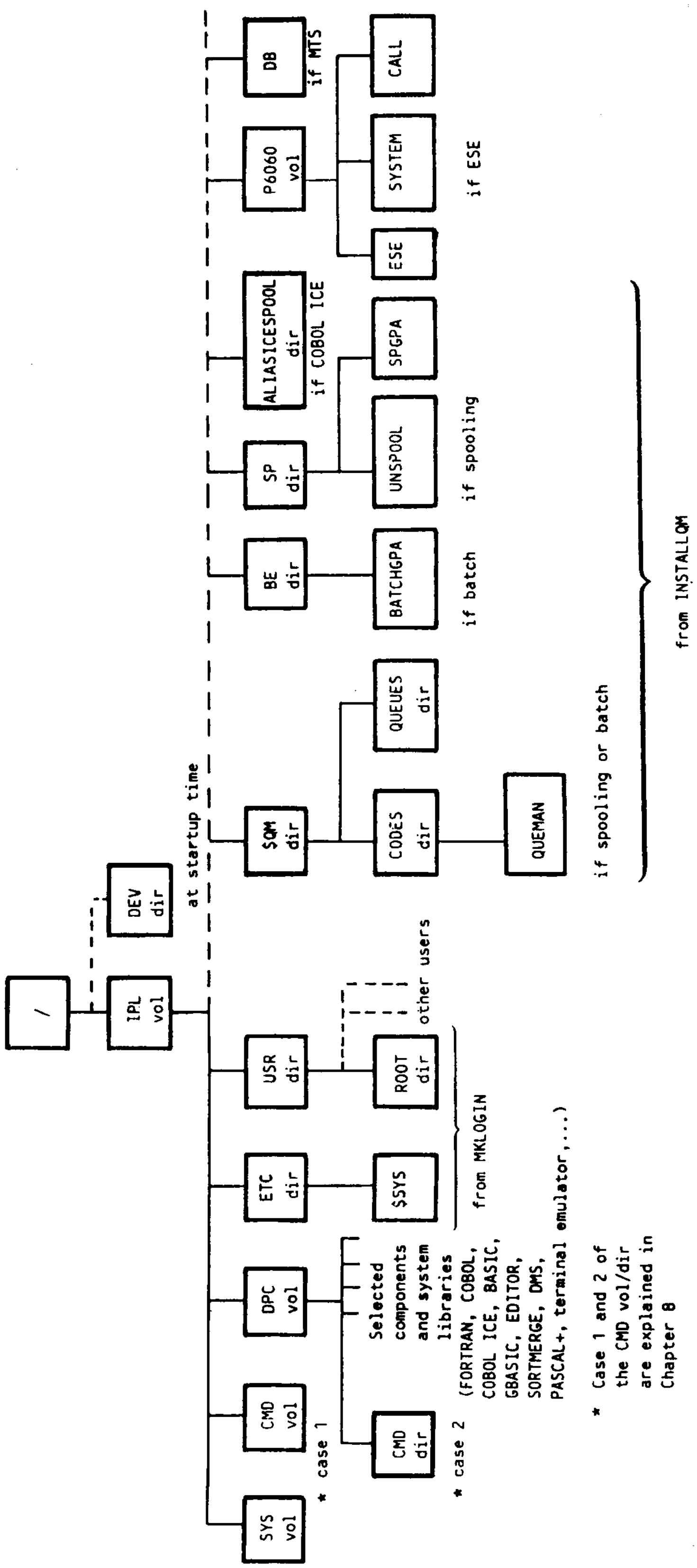


Fig. 9-1 Customer System Structure (User Volume)

INSTALLATION FROM FLOPPY DISKS

The steps taken to install a system from floppy disks onto hard disk are listed in Tab. 9-2.

STEP DESCRIPTION	COMMAND(S) AND PARAMETERS	
1 With console switch at ISL2, reinitialize cylinder 0 of HD using the suitable floppy disk of the distribution kit (see "Index of Software Distribution Media" in the Release Guide). Note: a complete reinitialisation deletes all the previous contents of the HD. It is possible to omit step 4, (with console switch at ISL2 but avoiding the reinitialization of the cylinder 0 of HD) if the existing volume dimensions are still adequate, overwriting SYS and DPC contents.	-	
2 Insert first floppy disk in floppy disk drive 1.	-	
3 Select MCL (option 2 on initial menu) and log in to MCL as root user.	2 ROOT	
4 Create a volume on a hard disk (here we assume only HD1 is used; otherwise similar volumes would be created on all hard disks; this is in any case necessary before these disks can be used).	MKVOL	
	VOL	volume name
	HD1	device name
	nnnnnnnn	size in bytes (See below table 9-5)
	1000	max. number of files
	/CR/	(user code not used)
	n	release number
	/CR/	(release level not used)

Tab. 9-2 Installation of Customer System on HD from FD (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS
5	Copy bootstrapper onto hard disk.	MKBOOT /IPL/SYS HD1
6	Mount the hard disk volume.	MNT HD1 /VOL
7	Copy the system volume from floppy disk into the hard disk volume.	COPY /IPL/SYS /VOL/SYS ["R] The "R option must be used only if an existing SYS volume is to be replaced. or (if a larger SYS volume is required): MKVOL SYS volume name HD1 device name nnnnnnnn size in bytes (user defined) 50 max. number of files /CR/ (user code not used) n release number /CR/ (release level not used) CPTREE /IPL/SYS /VOL/SYS
8	Activate the login mechanism for the system on hard disk.	MKLOGIN /VOL
9	Unmount the hard disk volume.	UNMNT /VOL
10	Log out of MCL and then out of MOS.	LOGOUT LOGOUT
11	Set console switch to ISL1 to work from hard disk, and press RESET.	-
12	Select MCL (option 2 on initial menu) and log in to MCL as root user.	2 ROOT
13	Set working directory.	SETWDIR /IPL/SYS

Tab. 9-2 Installation of Customer System on HD from FD (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
14	Restore dependent components volume DPC onto hard disk from floppy disks.	FLDUMP	
		R	function is "restore"
		/IPL/DPC	disk volume path name
		/CR/	default FDU: FL1
		Y	overwrite (only if the DPC volume is being replaced). This answer is not requested if the DPC volume is being created for the first time.
		/CR/	confirm (next) floppy inserted
		repeat for all floppies	
		Q	function is "quit"
15	Hard disks linking, if required.	LINKDEV IPL HD1 ... HDn	
		MODCON $CON	
		E	display and modify option
		50	unit number
		modify correspondence between physical hard disks and logical hard disks according to the LINKDEV command executed (see the LOGHD parameter in Chapter 4)	
16	Log out of MCL and then out of MOS.	LOGOUT LOGOUT	
17	Set console switch to ISL1 to work from hard disk, and press RESET.	-	
18	Select MCL (option 2 on initial menu) and log in to MCL as root user.	2 ROOT	
19	Install the spooling system, if required, using SPCONF (I option).	See Part V	
20	Install the batch processing system if required.	INSTALLQM /IPL/DPC/SERVICE /IPL TYPE=BE ["R] (omit "R option if spooling already installed)	

Tab. 9-2 Installation of Customer System on HD from FD (cont.)

STEP DESCRIPTION	COMMAND(S) AND PARAMETERS
21 Create the alias mechanism for COBOL to use system spooling, if required.	MKDIR /IPL/ICEALIAS

Tab. 9-2 Installation of Customer System on HD from FD

INSTALLATION FROM SCT

The steps taken to install a system from SCT onto hard disk are listed in Tab. 9-3.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Set console switch to ISL2.	-	
2	Insert the first SCT and press RESET.	-	
3	For the "SYSTEM LOADING" phase, specify HD to be loaded onto, and confirm (tape label VERIFY information is then displayed).	1 /CR/	number of HD to be used confirmation to continue Note: if using a WSL1, check that its speed is set to 9600 baud. If a different speed is set, use the WSL1 set-up to modify it.
4	If the disk is empty the size of the volume is requested.	nnnnnnnn	size in bytes (See below table 9-5)
5	If HD is not empty (already contains a SYS volume), confirm overwrite.	/CR/	confirms complete overwrite (or "1" to overwrite SYS volume only or "0" to abort)
6	If /CR/ typed, the volume size is requested, as in 4 above. If 1 is typed (to overwrite SYS): - if the new SYS takes less space, the excess should be recovered (compacting /IPL) - if the new SYS takes more space, SYS LENGTH DIFFERENCE message is displayed, and the installation phase is aborted.	nnnnnnnn	size in bytes (See below table 9-5)

Tab. 9-3 Installation of Customer System on HD from SCT (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
7	When END OF LOADING message appears (indicating HD initialised and containing the system volume), remove first SCT and insert second (if the DPC volume resides on another tape; do not remove the SCT if the DPC volume is there).	-	
8	Set switch to ISL1 and press RESET in response to prompts.	-	
9	Select MCL (option 2 on initial menu).	2	
10	Set working directory.	SETWDIR /IPL/SYS	
11	(Only if the DPC volume resides on a second tape) Restore the dependent components volume DPC to HD from SCT. Remove second SCT.	VOLSR FUNC=RESTORE TID=DPC SRC=/IPL/FREE/DPC DEST=/IPL/DPC OCCNUM=1	function is "restore" /IPL/FREE/DPC volume on the DPC tape is restored on disk with the name /IPL/DPC
12	Activate login mechanism for the new system.	MKLOGIN /IPL	
13	Log out of MCL, then back in again to have Shell commands directly accessible.	LOGOUT 2 ROOT	logout of MCL MCL option on initial menu

Tab. 9-3 Installation of Customer System on HD from SCT (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS
14	Hard disks linking, if required.	LINKDEV IPL HD1 ... HDn MODCON $CON E display and modify option 50 unit number modify correspondence between physical hard disks and logical hard disks according to the LINKDEV command executed (see the LOGHD parameter in Chapter 4)
15	Log out of MCL and then out of MOS.	LOGOUT LOGOUT
16	Set console switch to ISL1 to work from hard disk, and press RESET.	-
17	Select MCL (option 2 on initial menu) and log in to MCL as root user.	2 ROOT
18	Install the spooling system, if required using SPCONF (I option).	See Part V
19	Install the batch processing system, if required.	INSTALLQM /IPL/DPC/SERVICE /IPL TYPE=BE ["R] (omit "R option if spooling already installed)
20	Create the alias mechanism for COBOL to use system spooling, if required.	MKDIR /IPL/ICEALIAS

Tab. 9-3 Installation of Customer System on HD from SCT

INSTALLATION FROM MTU

The steps taken to install a system from MTU onto hard disk are listed in Tab. 9-4.

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
1	Set console switch to ISL2.	-	
2	Insert the first MTU and press RESET.	-	
3	For the "SYSTEM LOADING" phase, specify HD to be loaded onto, and confirm (tape label VERIFY information is then displayed).	1 /CR/	number of HD to be used confirmation to continue Note: if using a WSL1, check that its speed is set to 9600 baud. If a different speed is set, use the WSL1 set-up to modify it.
4	If the disk is empty the size of the volume is requested.	nnnnnnnn	size in bytes (See below table 9-5)
5	If HD is not empty (already contains a SYS volume), confirm overwrite.	/CR/	confirms complete overwrite (or "1" to overwrite SYS volume only or "0" to abort)
6	If /CR/ typed, the volume size is requested, as in 4 above. If 1 is typed (to overwrite SYS): - if the new SYS takes less space, the excess should be recovered (compacting /IPL) - if the new SYS takes more space, SYS LENGTH DIFFERENCE message is displayed, and the installation phase is aborted.	nnnnnnnn	size in bytes (See below table 9-5)

Tab. 9-4 Installation of Customer System on HD from MTU (cont.)

STEP	DESCRIPTION	COMMAND(S) AND PARAMETERS	
7	When END OF LOADING message appears (indicating HD initialised and containing the system volume), remove first MTU and insert second.	-	
8	Set switch to ISL1 and press RESET in response to prompts.	-	
9	Select MCL (option 2 on initial menu). Ignore "UNDEFINED WORKDIR" message, due to lack of login mechanism.	2	
10	Restore the dependent components volume DPC to HD from MTU. Remove second MTU.	/IPL/SYS/FILETAR R /IPL/FREE/DPC /IPL/DPC /CR/ /CR/ /CR/	 function is "restore" volume name on tape path name on disk default drive number: 1 default occurrence number on tape: 1 no options
11	Activate login mechanism for the new system.	MKLOGIN /IPL	
12	Log out of MCL, then back in again to have Shell commands directly accessible.	LOGOUT ROOT	
13	Hard disks linking, if required.	LINKDEV IPL HD1 ... HDn MODCON $CON E 50 modify correspondence between physical hard disks and logical hard disks according to the LINKDEV command executed (see the LOGHD parameter in Chapter 4)	 display and modify option unit number
14	Log out of MCL and then out of MOS.	LOGOUT LOGOUT	

Tab. 9-4 Installation of Customer System on HD from MTU (cont.)

STEP DESCRIPTION	COMMAND(S) AND PARAMETERS
15 Set console switch to ISL1 to work from hard disk, and press RESET.	-
16 Select MCL (option 2 on initial menu) and log in to MCL as root user.	2 ROOT
17 Install the spooling system, if required using SPCONF (I option).	See Part V
18 Install the batch processing system, if required.	INSTALLQM /IPL/DPC/SERVICE /IPL TYPE=BE ["R] (omit "R option if spooling already installed)
19 Create the alias mechanism for COBOL to use system spooling, if required.	MKDIR /IPL/ICEALIAS

Tab. 9-4 Installation of Customer System on HD from MTU

Sizing the First Volume when Installing a System

When installing a system, from whatever magnetic support (floppy disks, SCT or MTU), one of the first steps to be carried out is to create a volume on the hard disk.

The table below gives the values to be entered for this volume, according to the type of hard disk being used.

HARD DISK UNIT(S)		SIZE OF THE VOLUME (in bytes)
ST506	(10M bytes)	9721156
SASI	(14M bytes)	14689280
HDU	(18M bytes)	17397760
ST506	(20M bytes)	19919360
ST506	(27M bytes)	27965440
ST506	(40M bytes)	41234944
SMD	(60M bytes)	61523456
ST506	(65M bytes)	65994752
SMD	(120M bytes)	123238400
ESDI	(140M bytes)	140000000
SMD	(275M bytes)	274676160

Tab. 9-5 Size of the Volume Created at Installation Time

Values given above do not take into account the hard disk space which could be dedicated to diagnostic programs.

As diagnostic programs vary from one installation to the other, the dimension of those involved in the current installation should be considered. Typically, a normal diagnostic environment size is 1.5 Mb.

If a diagnostic environment is required, the disk space used by diagnostic programs must be subtracted from the values given above in order to calculate the number of bytes to be entered as volume size.

Note: If the LDHMU2 program (environment activator) is used for the installation phase, 16384 bytes must be subtracted from the values given above in order to calculate the number of bytes to be entered as volume size.

10. PREPARATION OF THE MOS USER ENVIRONMENT

GENERAL

The environment directly visible to end-users of a MOS system at run time is provided by programs which run in "user mode". This term reflects the fact that these programs are users of the operating system. The user environment can be preconfigured to suit individual hardware installations and application requirements before an initial system startup. It can also be reconfigured to adapt to changes in hardware or usage of the system.

Configuring the user environment involves:

- providing information about the required general characteristics including which application facilities are present within the environment

- providing information about each configurable application environment.

The information about the MOS user environment in general is interpreted at startup time by Grandpa, the control program for all programs which run in user mode. Grandpa is the first process to become active in user mode when the system is started up.

The information about each configurable application environment is subsequently interpreted by its main environment monitor when it is first activated.

This chapter describes the overall configuration of the MOS user environment for Grandpa, and gives references to documentation on the configuration of the application environments that may be present.

THE MOS USER ENVIRONMENT

The elements associated with the MOS user environment that are directly visible to end-users of the system are:

- work stations, one of which is master

- application environments, each available at one or more work stations

- any commands directly activated by Grandpa instead of from Shell

- special functions, for example, date-setting, shutdown.

An application environment may provide:

- interactive programming (in Interpreted BASIC or COBOL ICE)
- emulation (ESE, terminal emulators)
- transaction handling (the MTS system)
- general purpose facilities (Shell or BEAM - all other environments can be entered via Shell or BEAM, programs in any available language can be run, and utilities such as SORTRUN and EDITOR can be called up)
- data handling (DMS).

The configuration of the user environment entails specifying:

- the usable work stations
- the facilities associated with each usable work station
- the work stations that have master status
- the interactive program that is started when an application environment is entered, for each application environment
- the non-interactive programs that are run in association with application environments
- system facilities such as spooling and batch
- function packages (system or user) available for general use, if any
- the Shell commands for direct activation, if any
- the programs that initialise the user environment
- the programs that close down the user environment.

This information is supplied to Grandpa in the form of a configuration file which is potentially different for each installation and may also vary from one system session to the next.

THE USER ENVIRONMENT CONFIGURATION FILE

The MOS user environment configuration file is an object file in the format expected by Grandpa. It contains configuration records; one for each l-module (program or package) that needs to be defined plus one for each work station that has master status. This file must always be present in the system volume (/IPL/SYS) at system startup time, with the fixed name $CONFGP.

Prior to the first system startup, a source file should be prepared in a predefined format using EDITOR. This is then converted to object format using the Shell command PARSER (see Part V of this manual). If your requirements subsequently change, this procedure should be repeated and the system restarted.

The information in this file is used by Grandpa to:

- define a master work station
- initialise the user environment
- supply a logging activity (by cooperating with the Local Management System) to monitor any activity run under Grandpa's control, gathering information useful for system maintenance
- monitor and respond to such events as user login/logout and environment selection throughout the system session
- carry out an orderly system shutdown on user request.

USER ENVIRONMENT CONFIGURATION PARAMETERS

Each record in the user environment configuration file contains up to five parameters. These are:

- keyword
- name
- path name
- initial string
- working directory.

These parameters are described in the paragraphs that follow.

THE KEYWORD PARAMETER

This parameter specifies the configuration record type, which corresponds to a predefined category of l-module, having certain operational characteristics. Thus the keyword used to declare a particular l-module indicates the type of activation it requires, its priority, time slice, and so on are appropriate.

For interactive programs, there is a set of possible keywords which are in fact work station identifiers. Thus the association of interactive programs with particular work stations is implicit in the records which describe them.

The work station identifier keywords are also used to assign other functional characteristics to work stations.

The l-module categories and their corresponding keywords are listed below:

CATEGORY OF L-MODULE		KEYWORD
Initial programs		INIT, PINIT
User or system packages		CALL, PCALL
Non-interactive programs:	server	SERV
	foreground	FG, PFG
	standard	START, PSTART
	background	BG, PBG
Interactive programs		TTYA, TTYB, ... (local work station identifiers) RTTY (work stations for dynamic login identifiers) ALLT (identifier for all the work stations: local and for dynamic login)
Final programs		TERM

The order in which Grandpa activates (that is, loads, and starts if appropriate) the l-modules (programs or packages) inserted in its configuration file is as follows:

```
SERV   programs
PINIT  programs
PCALL  modules
PFG    programs
PSTART programs
PBG    programs
LOGIN  program (if the login data base is present)
INIT   programs
CALL   modules
FG     programs
START  programs
BG     programs
interactive programs
TERM   programs
```

Within each category except interactive programs, l-modules are loaded (and started if appropriate) in the order in which they appear in the configuration file. Interactive programs are loaded and started as necessary according to the user's activities. The use of each keyword is explained below.

If the login data base is present, and if the NOMLOG label is not present in the Grandpa configuration file (see below "The NOMLOG Function" section), the prompt for login is displayed on the master work station after execution of the directives identified by the SERV, PINIT, PCALL, PFG or PSTART keywords.

At this stage only a user belonging to the privileged group may log in, and he must interact with the DATE function (and, possibly, with the NEWVOL function).

When this phase is completed, Grandpa continues by executing the remaining directives present in its configuration file.

Note: In a distributed configuration, after any user packages have been loaded and server programs activated, remote programs can be loaded and started by inserting their remote path names in the Grandpa configuration file. As this feature uses the remote access facility, it is necessary to activate a remote program only when the server environment (see below) is available. In order to to ensure this, the remote program activation should be preceded by a suitable directive which suspends the Grandpa configuration file execution for a certain time, as shown in the following example:

```
PINIT=/IPL/DPC/CMD/WAITNET,<W>nn;
INIT=/../machinename/PROGRAM;
```

The optional W parameter specifies an integer number of seconds the system will wait before continuing to execute the Grandpa configuration file (the default value of 60 seconds is used if the W parameter is missing).

The INIT, or PINIT, Keyword

This keyword is used to declare initial programs. Initial programs are those that must be executed before any others, that is, once only at the beginning of a system session. Their activation is one of the initial actions carried out by Grandpa at startup time in association with the master work station. Typically, such programs initialise some aspect of the application, for example, a program to mount a disk that will be used by certain interactive programs.

The CALL, or PCALL, Keyword

This keyword is used to declare user or system packages. Such packages provide libraries of functions for general use, for example, mathematical functions, run-time support routines. The mechanism of loading and initialising such libraries separately from the programs that use them relieves the individual programs of this task. This mechanism is intended only for general libraries; those specific to particular application environments should be the responsibility of the monitor which manages the environment. Packages are loaded and initialised at startup time as one of Grandpa's initial actions.

Up to a total of sixteen CALL or PCALL entries may be specified. More entries will be ignored.

The SERV Keyword

This keyword is used to declare server programs. These are non-interactive programs in the highest priority class (only Grandpa has a higher priority) which must run to completion whenever called. Their priority range is from 500 to 999 and their time slice is infinite (that is, their execution cannot be interrupted by another program). These programs are loaded and started at startup time as one of Grandpa's initial actions.

The FG, or PFG, Keyword

This keyword is used to declare foreground programs. These are non-interactive programs with high priority which must run to completion whenever called. Their priority range is from 1000 to 1999 and their time slice is infinite (that is, their execution cannot be interrupted by another program). These programs are loaded and started at startup time as one of Grandpa's initial actions.

The START, or PSTART, Keyword

This keyword is used to declare non-interactive standard programs. These are programs with normal priority (2000 to 2999) and time slice 100 ms. Typically these are system services, for example, the spooling system dispatcher, a terminal emulator. These programs are loaded at startup time and executed according to scheduling policy.

The BG, or PBG, Keyword

This keyword is used to declare background programs. These are non-interactive programs which carry out low priority activities, for example, the batch system monitor, a system statistics routine. Their priority is from 3000 upwards and their time slice is 100 ms. These programs are loaded at startup time and executed according to scheduling policy.

Up to a total of sixteen SERV, PFG, FG, PSTART, START, PBG or BG entries may be specified. More entries will be ignored.

WS Identifier Keywords

Work station identifiers are used to associate interactive programs (up to a maximum of nine) with work stations and to assign functional characteristics to work stations. In particular:

- Potential master work stations are identified.

- The interactive programs that are first activated in the application environments are allocated to the work stations where these environments are available.

- Login programs are assigned to work stations, if desired.

If work stations are declared individually, only those explicitly named will be usable (up to a maximum of thirtytwo). Alternatively, the general WS keyword ALLT declares all work stations to be usable, but also renders them functionally identical. If all work stations are required to provide the same facilities, it is appropriate to use the general identifier. But, if there are to be any differences at all, this form cannot be used, as specific WS keywords cannot appear in a file containing the general keyword.

If a work station results as having only a single interactive program ("monofunction" work station) the program is loaded and started automatically by Grandpa when the user logs in. In this case the user has no visibility of Grandpa at all, but sees only the single application environment provided for his use.

At a "multifunction" work station, on the other hand, when the user logs in he is presented with an initial menu of symbolic names, from which he can choose an application environment or execute a command or special function as applicable. The user types the number which is displayed alongside the symbolic name to make his selection. The corresponding interactive program is then loaded and started by Grandpa, or the requested special function is executed by Grandpa (see also the discussion of the symbolic name parameter below).

Note that any Shell command can be configured as a program selectable via the initial menu. Typically this is useful for minimum configurations where the Shell interpreter is not present but certain Shell commands are present because their functions are required. Since the Shell environment does not exist in such configurations, the commands can only be activated directly as programs. Note also that the LOGOUT function is inserted automatically by Grandpa on all menus.

Interactive programs belong to the standard class of programs (c.f. "START" above), with priority range 2000-2999 and time slice 100 ms.

The TERM Keyword

This keyword is used to declare final programs. These are the programs that must be executed after all other programs, that is, once only at the end of a system session in association with shutting the system down. They are loaded and executed when Grandpa has received the shutdown request.

THE NAME PARAMETER

This parameter specifies a symbolic name. The symbolic names of the main interactive programs in the application environments appear on the initial menu displayed at the work station. Special functions are identified by predefined symbolic names, as are the "master" status and the definition of a login program, but the predefined names do not appear on the work station menu, except for "SHUTDOWN". The appropriate symbolic name may also appear in a warning message if Grandpa is unable to start a program (interactive or non-interactive) or to call a package.

There are currently seven predefined values for this parameter:

MASTER	master status function
GMASTER	global master function
LOGIN	login program function
DATE	date and time-setting function
NEWVOL	system disk removal function
NOMLOG	system start up without operator login
SHUTDOWN	shutdown procedure function (this name does appear on the work station menu)

The first and second values are the means of assigning master status to work stations, the third is the means of defining a login program for a terminal, and the others are the special functions implemented by Grandpa. When any of the predefined names except LOGIN is used, no path name, initial string or working directory need be specified (they will be ignored in any case). The LOGIN name requires parameters in the same way as any interactive program definition. An explanation of the use of the various predefined names follows.

MASTER and GMASTER Functions

Master status is conferred by MASTER or GMASTER. It may be assigned to one terminal, more than one, or none. If more than one work station is assigned master status, then the first of these to be switched on is recognized as the current (local) master by Grandpa. If no master is specified, all usable terminals are given master status automatically and the first to be switched on is recognized as the current master.

Note: It is advisable to define MASTER (or GMASTER) more than one terminal. If only one is specified, and this is not working (for any reason: switched off or disabled), the system cannot start up.

At startup time, Grandpa's initial actions are carried out in association with the local master terminal. These initial actions therefore cannot be started until a terminal with explicit or implicit master status is switched on. No other terminals will accept user login until all initial actions have been completed.

As soon as a master has been established, Grandpa can display any warnings, and then carry out the initial actions. If the master goes down or is switched off before this phase is complete, a new attempt is made with another master if this is feasible.

During the initial phase the master (and only the master) can:

- set the date and time (see DATE below)
- remove the system disk (see NEWVOL below)
- receive asynchronous messages from the operating system.

After the initial phase, asynchronous messages are sent to the master terminal (defined MASTER or GMASTER). For this reason, all work stations given master status must have a login program which requests the splitting of the screen to give the "system line". Alternatively, at least one terminal in the configuration must have a screen-splitting login program, so that it can become the master. If the master terminal goes down or is switched off, a new master is established. This will be the second terminal with master status to have been switched on, if explicit masters were specified, or just a terminal with split screen if no explicit masters were specified.

There must always be a master in existence, so that asynchronous system messages are not lost.

A terminal belonging to the master system of a cluster, or to a system in a local network, can be defined global master of the distributed configuration. The necessary global status is conferred by GMASTER which is identical to MASTER in all other respects.

NOMLOG Function

The system is started up without the operator identity check, because the login program is not activated before initial operations (IPL warnings display, date setting, and so on) on the master work station. Only after this initial phase is the login program started as on any other work station.

LOGIN Function

To assign a login program to a terminal, the program is defined with the symbolic name LOGIN. If no explicit assignment is made, the default program used is /IPL/SYS/$LOG which contains LOGS (screen splitting) unless this has been changed to LOGN (no splitting). For the login mechanism to be effective, the command MKLOGIN must have been executed to create the login database structures.

A login program may be user-written, or one of two programs supplied by Olivetti may be used:

LOGS which splits the screen to give the system line (last row of screen) as a separate (logical) terminal.

LOGN which does not split the screen.

(The actual screen-splitting operation is carried out by Grandpa as directed by the login program.) These supplied programs can either be specified as the default program during system generation, or assigned explicitly via LOGIN.

Both the previous programs can be made non-interactive if they are started with a parameter specifying the user name, as shown in the following example:

```
WSid:LOGIN=/IPL/SYS/$LOG,<NAME>JOHN!MCL=/IPL/SYS/$VSH;
```

To disable the login mechanism, the system is used without the login data base structures, and all users are then effectively superusers. If there are login database structures but no login program is defined (either explicitly or by default), this is an error condition.

DATE Function

The system date and time are set as the first action after system initialisation, when a master work station has been established. The date and time-setting function is permitted only to the master terminal. Input is requested from the operator as follows:

```
ENTER DATE (MM/DD/YY) :
```

The date should be entered with two digits each for month, day and year, each pair separated from the next by a slash.

```
ENTER TIME (HH/MM/SS) :
```

The time in hours, minutes and seconds should be entered in an analogous format.

If an incorrect value is entered, the prompt is repeated.

NEWVOL Function

The system disk is mounted as directory /IPL in the memory volume at boot time. If it is to be removed, this must take place before any user programs are run, as it involves changes to the basic file system tree structure. The NEWVOL function enables these changes to be made. This function is permitted only to the master work station. There are three possibilities open to the operator:

1. To remount the system disk into a different directory, but still within the memory volume created at boot time.

2. To mount another disk into directory /IPL.

3. To mount another disk into a new directory in the memory volume.

Grandpa looks for the directory name in its configuration file. A complete path name must be specified as positional parameter in 'initial string'. If this parameter is missing, Grandpa prompts the operator for the directory name as follows:

```
ENTER DIRNAME FOR NEW DISK :
```

A complete path name should be entered. If the IPL device is a floppy disk drive, Grandpa then prompts:

```
CHANGE DISK - HIT <CR>
```

The operator should insert a new disk and then press carriage return to tell Grandpa to proceed.

SHUTDOWN Function

The shutdown procedure is by its nature necessary in all systems. It can be initiated implicitly by the sending of a completion code at the end of a program (see the Chapter "Activating the Programs and User Subsystems" in the MOS Programmer Guide) or explicitly via the work station menu. The shutdown function can be explicitly assigned to one, more than one, or no work stations. It can only be requested by a privileged user, that is a user belonging to the SYSTEM group (typically the system administrator - ROOT).

When the shutdown request is received, Grandpa looks for the waiting time specified as positional parameter in its configuration file. For example, the directive:

```
TTYA:MCL=/IPL/SYS/$VSH!SHUTDOWN,<>60;
```

associates with the TTYA work station, besides the MCL application environment, the SHUTDOWN activity with a delay time of 60 seconds.

If this parameter is missing, the requesting work station has to specify a delay time. Input is requested from the operator as follows:

```
ENTER TIME (<CR> = 90") :
```

The required delay time should be entered as a number of seconds; default is 90 seconds. After this time, Grandpa starts the actual shutdown procedure. This essentially involves the reversal of initialisation.

The actions taken to shut the system down are as follows:

1. All user programs that are running are interrupted and their execution is not resumed. (They are able to respond to the interrupt before being terminated as it is implemented as a soft interrupt brought about by the PMM interrupt primitive.)

2. The termination procedures of the user packages are called and they are unloaded in the reverse order to that in which they were activated.

3. Final programs are loaded and run in the order in which they appear in the configuration file.

4. The requesting work station is notified that shutdown is complete (after which the system no longer responds).

If a non interactive program, or a program started via the PINIT keyword, requests the shutdown, then the blinking string "77" is displayed as console code when the system is closed down.

THE PATHNAME PARAMETER

This parameter identifies the file corresponding to the program or package which is described in the configuration record. It must be a full path name starting with the "/" character which, taken singly, identifies the local root directory. If the first component is in fact "/..", the reference is instead to the global root directory. The path name must not be longer than 160 characters.

THE INITIAL STRING PARAMETER

The initial string is passed directly to the program or function package described in the configuration record when this is started. It consists of pairs of ASCII strings which are Pascal+ keys (parameter names) and associated values. The string is not previously decoded by Grandpa and its format is therefore of no relevance to the system but the total string length must not exceed 160 characters.

A non-empty string is compulsory only for function packages.

THE WORKING DIRECTORY PARAMETER

This parameter allocates a directory to be used as working directory by the program specified in the configuration record. (This parameter is not applicable to function packages as functions always run in the context of the calling program). A complete path name should be specified. The default is the local root, "/".

SOURCE FILE CREATION

To obtain a user environment configuration file, it is first necessary to create a source file for the PARSER utility using the Editor. The source file comprises a series of configuration directives, which can be entered in the file in any order. These directives contain the configuration parameters in a compacted format which largely eliminates duplication.

CONFIGURATION DIRECTIVES

Each directive in the source file associates one or more 1-modules (programs or packages) with a keyword, or possibly more than one keyword if these are WS identifiers. The number of configuration records generated in the object file by each source directive is:

```
number of keywords    *   number of programs/packages
 in the directive          declared in the directive
```

The total number of records generated must not exceed fifty.

Each directive consists of the following elements:

- one or more keywords (separated by exclamation marks if more than one)

- a delimiter (colon)

- one or more program or package descriptors, each containing the name, path name, initial string and working directory parameters as appropriate, in the format:

  name=pathname,initial string(working directory)

- an end delimiter (semi-colon).

Each field in the directives is limited to 160 characters. When the directives are decoded by PARSER, blanks have no special significance (that is, blanks within fields are preserved as part of the value). The move to a new line at the end of, or in the middle of, a directive, is recognized implicitly.

THE ENVIRONMENT LABEL

Before the normal directives, an additional general directive may appear which consists only of a user environment label which has ten characters and is enclosed between % signs:

%UNATTENDED%

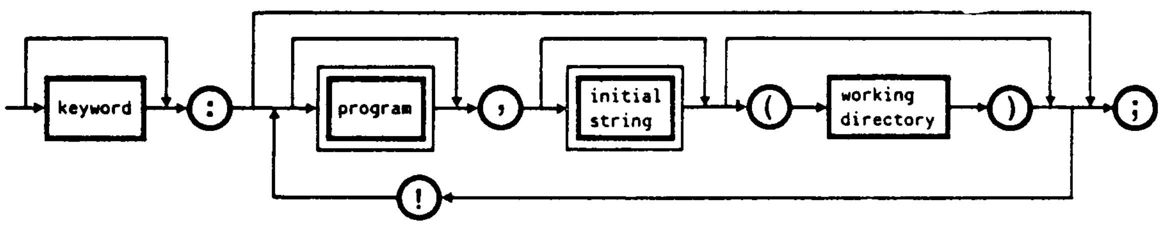
The unattended label specifies that the system has been started up remotely or using a timing mechanism. The alternative is local startup and this is the default assumed if this directive is missing.

Alternative modes of operation for the same system started up in different ways can be specified by splitting the user environment configuration file into blocks. Each block defines a different user environment, one of which is selected by Grandpa at startup time, according to information from the operating system on how the system has been activated. The first set of directives in the configuration file has no preceding label and defines a user environment with normal startup. Then, an unattended mode label must precede the set of directives for remote startup, if any.

SYNTAX OF CONFIGURATION DIRECTIVES

The syntax of the configuration directives can be represented by a set of syntax diagrams.

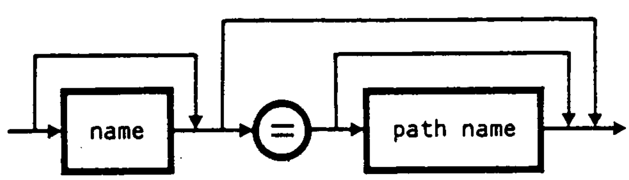
The general syntax of source directives is given by the following diagram.



The "program" field is a fixed combination of the "name" and "pathname" fields, which has been used for convenience as this part of the syntax is always the same.

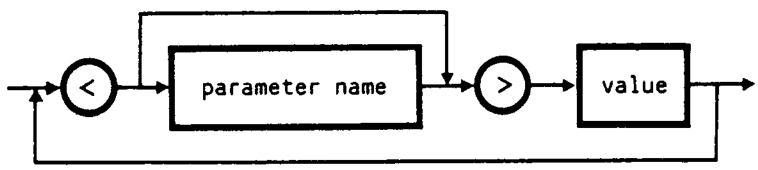
The expansion of "program" is shown in the diagram below.

Program



The expansion of the "initial string" field is shown below.

Initial String



In "initial string", the parameter name between the pointed brackets is the PASCAL+ "key" for a keyed parameter. Thus omitting the key is the means of specifying a PASCAL+ positional parameter.

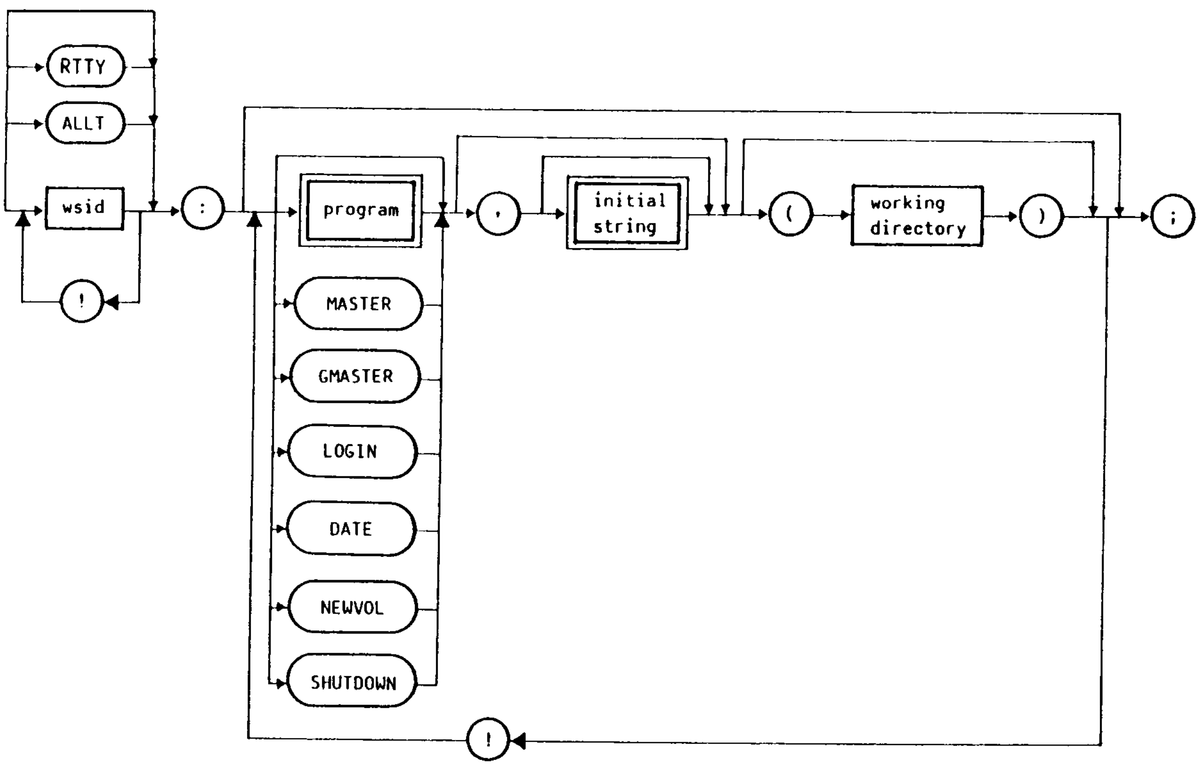
In all other cases, the omission of parameters in the configuration directives results in the use of default values. These are discussed under Directive Fields below.

Warning: Blanks are not automatically skipped in the parameter list; each parameter is defined as the character list (blanks included) delimited by '<' and '>' or ';' or '!' or '('.

WSid Directives

It is likely that more than one WSid directive will be needed to assign all initial interactive programs to work stations and to identify any masters.

The syntax of these directives is given by the diagram below.



For 1-module types other than interactive programs, a single directive with the appropriate keyword (INIT or PINIT, TERM, CALL or PCALL, FG or PFG, START or PSTART, BG or PBG) can declare all programs or packages of the corresponding type (or more than one directive can be used, as convenient).

Work Stations For Dynamic Login

Grandpa can handle intelligent work station connected in a OLILAN local area network in a manner similar to the way it handles those directly connected to the system where Grandpa resides.

These work stations must be notified to Grandpa using the RTTY keyword. This allow to define the menu entries that are displayed by Grandpa on their screens after a user login. The menu is common to any work station for dynamic login.

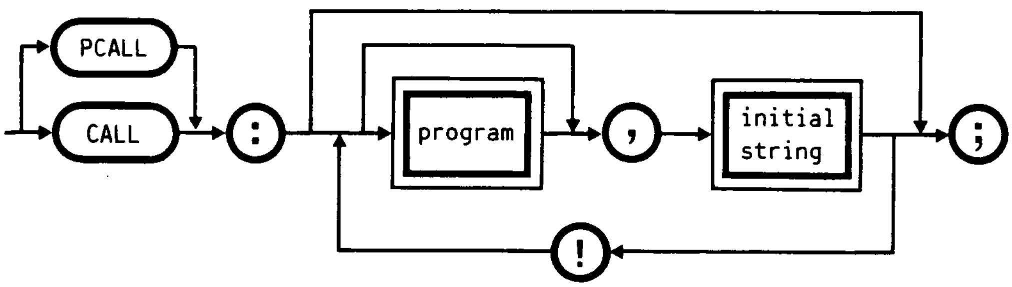
It is meaningless to define MASTER one of these work stations.

If one of these work stations is switched off, or if the OLILAN link goes down, Grandpa interrupts all that work station activities. The user is not disconnected: he is still logged in when the work station is switched on again or when the link is recovered.

To disconnect a work station from the system, a function key must be pressed at that work station when the login program is active, that is, when the prompt is displayed: the function key will have no effect at any other time, and the user will still be logged in.

CALL or PCALL Directives

The syntax of user package directives is given by the diagram below.

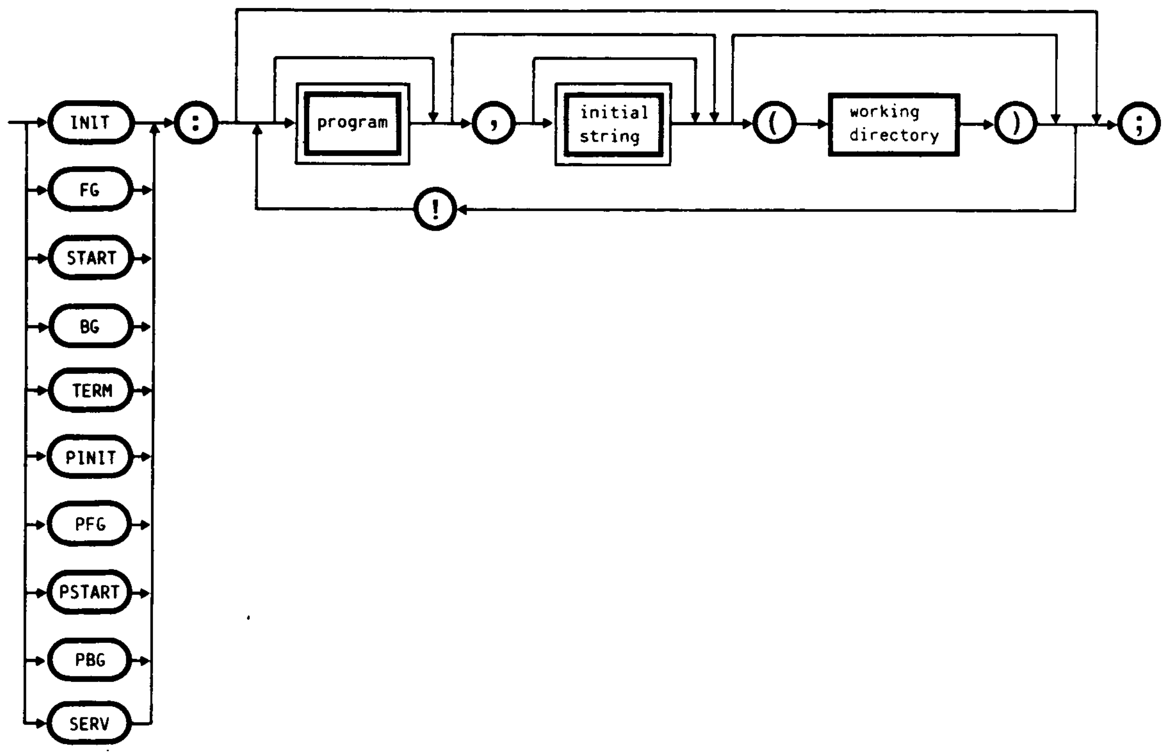


Note: In a distributed configuration, a user package which will be accessed from remote must be loaded before the $SERVER module is activated, as shown in the following example:

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>PAR;
PFG:/IPL/SYS/$SERVER;
```

Program Directives

The syntax of all other directives (INIT, PINIT, FG, PFG, START, PSTART, BG, PBG, TERM and SERV) is given by the diagram below.



DIRECTIVE FIELDS

The rules regarding character use in the various fields of the configuration directives are shown in the table below. (The "parameter name" and "value" fields belong to the initial string parameter.) Any characters that are unacceptable when used on their own can be used if prefixed with double quotes ("). The only exception is /CONTROL/ /D/ (shown in the table as ^D) which cannot be used at all.

FIELD	UNACCEPTABLE CHARACTERS
keyword	All special characters (that is, only alphanumeric characters accepted)
name	All special characters
path name	< ! ; ^D
parameter name	> ^D
value	< ! (; ^D

There is a comprehensive system of default values for individual fields and also for certain combinations of fields. These are shown in the table below. In this table, "empty" refers to the case where the preceding delimiter is present but the field is empty, while "missing" is when the preceding delimiter is also not present. Also, "previous" means the last specified value for the field in question, and is therefore a value from an earlier directive. Extensive use of defaults allows the directives to be presented in a highly compacted form, but is a more complicated exercise. Alternatively, the long format is the way Grandpa stores information internally and is easier to use.

FIELD(S)	DEFAULT IF EMPTY	DEFAULT IF MISSING	EXPLANATION OF MISSING IF APPLICABLE
keyword	previous keyword	-	-
descriptor	previous descriptor		
name	=pathname	-	-
pathname	previous pathname	-	-
name and pathname	previous name and pathname	-	-
parameter name and value	previous parameter name and value	the empty string	no preceding comma
working directory	-	local root directory	no preceding "(" (nor following ")")

GRANDPA CONFIGURATION FILE LIMITS

The Grandpa configuration file current limits are as follows:

GRANDPA CONFIGURATION FILE CHARACTERISTICS	MAXIMUM VALUE
Total number of characters in the source file	32767
Total number of directives	512
Total number of TTY directives (keywords: TTYA, ..., TTYZ, TTY0, ..., TTY5, RTTY, where A, ..., Z are all the letters in the English alphabet)	32
Total number of user packages directives (keywords: CALL, PCALL)	16
Total number of non interactive programs directives (keywords: SERV, PFG, PSTART, PBG, FG, START, BG)	16
Total number of entries in the menu displayed on the work station screen. (Each entry consists of at most 20 characters, which are the first 20 characters of the label in the terminal directive.)	32

DIRECTIVES FOR APPLICATION ENVIRONMENTS AND FACILITIES

The choice of application facilities to be included in the MOS user environment determines certain items which must be declared in the configuration file for Grandpa. The directives used to declare these items are largely predefined, according to known characteristics of the Olivetti-supplied software.

A list of the directives needed for each facility is given below, so that the user can see which modules need to be activated, their path names and any activation parameters.

The directives given below are syntactically correct and may be used in a Grandpa configuration file, substituting suitable values for the identifiers in lower case.

Depending on requirements, the user may substitute other keywords in these directives. See the preceding section "The Keyword Parameter" for the necessary information.

In particular, the user should refer to the relevant "Features and Functions" and/or "User/Configuration" guides when declaring in the Grandpa configuration file the environments MTS, BEAM, LMS, ONE and Terminal Emulators.

In the directives shown:

- "WSid" should reflect your choice of the work station(s) at which particular interactive facilities will be selectable via the initial menu and implies one or more directives, as required. If no such directive is given, these facilities will be available only via Shell.

- All symbolic names are just examples as these names are set by the user.

Note that in configurations that do not contain Shell, all interactive facilities must be selectable via the initial menu.

BASIC Environment

```
WSid:BASIC=/IPL/DPC/BASIC;
```

An initial string containing activation parameters (keyed) for the BASIC interpreter is optional.

BASIC Environment with Graphics

```
WSid:GBASIC=/IPL/DPC/GBASIC;
```

An initial string containing activation parameters (keyed) for the GBASIC interpreter is optional.

BASIC ESE

```
WSid:P6066=/IPL/DPC/CMD/SYS60,<>fparname;
```

The name of the configuration file passed to the P6066 BASIC emulator (**fparname**) is set by the user.

If the hardware configuration has more than one HDU which are not linked together (via the LINKDEV command) then the string to be entered in Grandpa's configuration file is:

```
WSid:P6066=/IPL/DPC/CMD/SYS60,<>fparname<>volname;
```

where **volname** is the user-defined name of the main volume of the HDU containing the P6060 volume.

Batch Facilities

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
START:/IPL/BE/BATCHGPA<ST>nnnn;
```

The execution of jobs in batch mode requires the activation of the queue manager and the batch monitor.

par is a dummy parameter as an initial string is required in a CALL directive but the queue manager requires no parameters. See also the "Queue Manager" section below.

The optional **ST** parameter specifies whether jobs can be enqueued but not executed until explicitly requested by the user (using the START option of the BATCH Shell command), in which case the value is STOP, or jobs can be enqueued and executed, in which case the value is START. If missing, the default START value is assumed.

BEAM Environment

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
PCALL:BEAMMON=/IPL/DPC/BEAM/BEAMMON,<BEAM>/IPL/DPC/BEAM/MAIN
                                    <BEAMDB>/IPL/DPC/BEAMDD;
START:BATCHGPA=/IPL/BE/BATCHGPA;
START:SPGPA=/IPL/SP/SPGPA,<>n<>m;
START:CMANAGER=/IPL/DPC/CMANAGER,<CMAN>COMx<LOGF>name1<STGA>stagesize
                                 <WFIL>name2<LOPT>m;
PFG:/IPL/SYS/$SERVER;
ALLT:BEAM=/IPL/DPC/BEAM/MAIN,<BNET>n;
```

On client systems, the last directive only is needed.

The queue manager and BEAM user packages are loaded in the server system. **par** is a dummy parameter as an initial string is required in a CALL directive but the queue manager requires no parameters. See also the "Queue Manager" section below.

BEAM is an optional parameter. It is the path name of the BEAM main program to be executed in background. If this parameter is missing, the BEAM main program must be under the path name /IPL/DPC/BEAM.
The **BEAMDB** parameter specifies the BEAM Data Base path name.

The **BNET** parameter in the last string is optional. If it is missing, the BEAM environment will use the BEAMMON and the BEAMDD of the current sub-network. If it is present, it specifies a particular BEAM Data Base (that is, the one belonging to the sub-network specified by the user - "n").

See below the "Spooling System", "Commit Facilities" and "Server Environment" sections for details concerning the fourth, fifth and sixth directives.

COBOL ICE

```
WSid:ICRTS=/IPL/DPC/ICRTS;
WSid:/IPL/DPC/ICOBOL;
```

The second directive is required for program preparation only, when the compiler is available as well as the interpreter.

Commit Facilities

```
SERV:/IPL/SYS/$SERVER;
START:CMANAGER=/IPL/DPC/CMANAGER,<CMAN>COMx<LOGF>name1<STGA>stagesize
                                  <WFIL>name2<LOPT>m<DUAL>name3;
```

These directives are only needed on the master system in a cluster configuration, or on the system where the Commit Manager resides in a distributed configuration.

The second directive starts the monitor which handles all information associated with the operation on files performed by the user transaction.
The **CMAN** parameter identifies the Commit Manager, making it independent from the machine name: its value must be COM followed by a fourth letter or digit.
The **LOGF** parameter passes the complete path name (starting with the "/" character) of the Commit Manager Log File to the monitor.
The **WFIL** parameter passes the complete path name of the Transaction Code File to the monitor. These two files are automatically created by the Commit Manager if they do not already exist.
The **STGA** parameter (5-character string) is needed to pass the size in bytes of the stage area to the Commit Manager. This size, which is user defined, must be large enough to contain the largest transaction in the application.
The **LOPT** parameter (1 character, either "Y" or "N") defines respectively append or rewrite mode on the Log File.
The **DUAL** parameter passes the complete path name (starting with the "/" character) of the duplicated Commit Manager Log File to the monitor. In order to use this duplicated Log file, the preceding LOPT parameter must be set to append mode (that is, LOPT=Y).

In a distributed configuration files handled by a Commit Manager must be stored on the system where the specified Commit Manager resides. Activities under Commit Manager control may be requested by application programs being executed on any system in the configuration.

In a cluster configuration files handled by a Commit Manager must be stored on the master system. Activities under Commit Manager control may be requested by application programs being executed on satellites or the master.

DMS Environment

```
WSid:DMS=/IPL/DPC/DMS;
WSid:ADMS=/IPL/DPC/ADMS;
WSid:INITDD=/IPL/DPC/INITDD;
```

The DMS interpreter, administrator and data dictionary initialiser are independently selectable, the last two in program preparation configurations only for the writing of procedures.

EDITOR

```
WSid:EDITOR=/IPL/DPC/EDITOR,<filename>;
```

GATEWAY

```
CALL:/IPL/DPC/MTS/MTSCATLG,<DBENV>SERVER<>f;
CALL:/IPL/DPC/MTS/MSWMAN,<MBOX>/IPL/mailbox_file_name;
START:/IPL/DPC/MTS/MSWDIS;
FG:MTS=/IPL/DPC/MTS/GMAN,<DMTS>MTSA;
```

Besides the above directives, the following must also be inserted, according to the required type of GATEWAY.

In the case of MOS GATEWAY for **COSMOS-MOS** Communication:

```
START:GATE=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/DPC/MTS/GMAN/ERRMSG
                              <OVLI>/IPL/DPC/MTS/OVLSMAN
                              <DMTS>MTSA<NAME>GATE<SYNC>nnnn<COSMOS>mm;
ALLT:MTSLOGON=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/DPC/MTS/GMAN/ERRMSG
                                 <OVLI>/IPL/DPC/MTS/OVLSMAN<DMTS>MTSA
                                 <COSMOS>mm<CMETRA>xxxx<GETN>yyyy;
```

The **COSMOS** parameter is an integer which specifies the number of servers to be dedicated to the work stations connected to the MOS system.
The **CMETRA** optional parameter specifies the METRA relevant to the COSMOS terminal. If this parameter is missing, then a module named METRAC is assumed to be present under the /IPL volume.
The **GETN** parameter specifies the name of the server program which handles "GET pending" type messages.

In the case of MOS GATEWAY for **T5_L1 NODE:**

```
START:GATE=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/GMAN/ERRMSG
                              <OVLI>/IPL/DPC/MTS/OVLSMAN
                              <DMTS>MTSA<NAME>GATE<SYNC>nnnn<T5NODE>kk;
ALLT:MTSLOGON=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/GMAN/ERRMSG
                                 <OVLI>/IPL/DPC/MTS/OVLSMAN
                                 <DMTS>MTSA<T5NODE>kk<TMETRA>zzzz;
```

The **T5NODE** parameter is an integer which specifies the number of servers to be dedicated to the work stations connected to the MOS system.
The **TMETRA** optional parameter specifies the METRA relevant to the L1_T5 NODE terminals. If this parameter is missing, then a module named T5METRA is assumed to be present under the /IPL volume.

The other directives and parameters are described later, in the "MTS Transaction Handler System" section.

LMS Local Management System

```
CALL:/IPL/$QM/CODES/QUE_LMS,<TFS>nnnn<TDN>pathname1<FSET>PM/FS/LN/TH/RC
                            <LDN>pathname2<LFS>mmmm<AGS>nn<AC>/i/j/k/l
                            <MC>/m/n/p/q<MOD>x;
FG:/IPL/DPC/SERVICE/SYSLOG;
```

The first directive makes the Local Management System monitor available, passing it some parameters.

TFS is the maximum number of bytes that a trace file may contain.
TDN is the complete path name of the directory containing the two trace files.
FSET is the functional set of primitives that are to be traced. The list in the first directive contain all possible primitives sets. Any subset, or all, may be selected.
LDN is the path name (either local, for example /IPL/LMS/LOG, or global, for example /../N1M1/IPL/LMS/LOG) of the log directory.
LFS (which is only meaningful if the log directory is on the local machine) specifies the maximum size in bytes for the Log File before a switch is made to the alternate file (the value applies to them both). The default value is 10200 for each of the two log file.
AGS specifies the logical number (decimal) of the machine on which the Agent is present (if this parameter is missing the Agent is assumed to reside on the local machine).
AC specifies the Agent Alarms Classes.
MC specifies the Master Work Station Alarms Classes.
MOD specifies the type of Log Mode (either 'E' for extended mode or 'N' for normal mode, which is the default type).

The second directive activates, with high priority, the system logger program.

Message Switching Facilities

```
PCALL:/IPL/DPC/MTS/MSWMAN,<MBOX>mailbox_file_name<APND>m;
START:/IPL/DPC/MTS/MSWDIS;
```

If the optional **MBOX** parameter is missing, in the first directive above, the default /IPL/MAILBOX file is used.

The optional **APND** parameter defines whether the "q_file" is to be written in append (the value is **Y**) or if it is to be overwritten (the value is **N**). If missing, the file is overwritten.

The second directive is only needed on systems belonging to distributed configurations. On stand alone systems, only the first directive is required.

```
PCALL:/IPL/DPC/MTS/MTSCTLG,<DBENV>SERVER<>f;
PCALL:/IPL/DPC/MTS/MSWMAN,<MBOX>/IPL/mailbox_file_name<APND>m;
PFG:/IPL/SYS/$SERVER;
FG:MTS=/IPL/DPC/MTS/GMAN,<INIT>/IPL/DPC/MTS/CONFILE<DMTS>MTSx;
START:/IPL/DPC/MTS/MSWDIS;
ALLT:MTSLOGON=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/ERRMSG<OVLI>/IPL/DPC/MTS/OVLSMAN
                                  <DMTS>MTSx<STGA>nnnnn;
ALLT:CONFMTS=/IPL/DPC/MTS/CMD/CONFMTS;
```

The first directive makes the MTS file-handling monitor available. The optional **DBENV** parameter must only be inserted in this directive if the Data Base has been configured. It specifies if the Data Base resides on the current machine (the parameter's value is 'SERVER') or if it resides on another machine (the value is 'CLIENT').

The second directive initialises the shared MTS monitors. The **MBOX** parameter is optional and its default value is /IPL/MAILBOX. The optional **APND** parameter defines whether the "q_file" is to be written in append (the value is **Y**) or if it is to be overwritten (the value is **N**). If missing, the file is overwritten.

The third directive starts the server activator in the operating system.

The fourth directive starts the MTS manager, passing it the name of the MTS configuration file. The **INIT** parameter may be omitted if the MTS configuration file is contained in the GMAN program directory. The **DMTS** parameter identifies the Transaction Handler System, making it independent from the machine name: its value must be MTS followed by a fourth letter or digit.

The fifth directive activates the Message Switching mechanism.

The sixth directive starts the interactive environment main program on all terminals, passing it the name of the error logging file, the name of the main program's overlay, the name of the Transaction Handler System used and the size of the stage area. This last parameter, which is optional, can be used to specify, for a specific work station (using the TTYn keyword) a value of the stage area lower than that defined in CONFILE. The name of the Transaction Handler System is usually the same as that specified in the fifth directive above (which activates the MTS manager), but it might be different, thus indicating that the interactive environment on the current machine refers to a Transaction Handler System resident on another machine of the distributed configuration.

The last directive assigns the MTS configuration utility to each terminal.

On agent systems the first and sixth directives are needed. The second and the fifth directives are needed if the message switching service is required.

For a system used as background node the sixth directive above must be substituted by the following:

```
START:MTSLOGON=/IPL/DPC/MTS/SMAN,<ERMS>/IPL/ERRMSG
                                 <OVLI>/IPL/DPC/MTS/OVLSMAN
                                 <DMTS>MTSx<STGA>nnnnn
                                 <NAME>xxxx<SYNC>mmmm;
```

which activates a background program (non connected with any work station) and which needs two new parameters: **NAME** is a four character string defining the logical name of the background node configured in CONFMTS, and **SYNC** is needed to delay the activation of the background environment by a specified number of seconds. An advisable value is 0120.

NEMOS

```
CALL:/dir/eea-name,<>password<>dirname<>opmode<$user>username<$maxl>maxl;
START:/dir/eea-server-name;
START:/dir/aa-name;
```

The first two directives activate the End to End Agent program. The third directive activates the Alarm Agent program.

dir is the MOS path name of the directory containing the programs to be activated.

eea-name is the name of the End to End Agent program.
password is an 8-character string containing the remote logon password associated with the TARA SYSMON command. This parameter is optional. If it is omitted, no password will be required to access the L1 MOS system.
dirname is a character string which specifies the name of a directory containing CMDFILE (the End to End Agent command file) and the ADM module of the End to End Agent program.
opmode is a numeric value that specifies one of the two following operating modes of interaction between End to End Agent and TARA: a value of 1 indicates a normal operating mode, a value of 2 indicates an interactive operating mode. This parameter is optional; if it is omitted, it defaults to 1.
username is a MOS login name.
maxl is a numeric value that specifies the maximum length for the data area that will be used by the End to End Agent Main module. It cannot be greater than 256 characters.

eea-server-name is the name of the End to End Agent server program.

aa-name is the name of the Alarm Agent program.

NMS Network Monitoring System

See the "Configurable Application Environments" section at the end of this Chapter.

OLILAN

```
START:/IPL/DPC/OLILAN/SERVER,<>/IPL/DPC/OLILAN<>instances;
START:/IPL/DPC/OLILAN/NSERVER,<>/IPL/DPC/OLILAN;
```

The **instances** parameter in the first directive is an integer representing the number of File Servers (DOS_SERVER) activated by SERVER. If missing, the default value of 1 is assumed.

Further information on installing OLILAN is given in the manual OLILAN User Guide.

ONE Open Network Environment

In the case of ONE for **X25** and **PORT:**

```
CALL:SLAM=/IPL/USR/one/slam,<ONEPORT>SL_PORT;
FG:ONE=/IPL/USR/one/cm,<>R(/IPL/USR/one);
START:FTF=/IPL/USR/ftr/FTRmonitor;
START:BNS=/IPL/OWS/MOSNET/NETBOOT,<>(/IPL/OWS/MOSNET);
```

The first directive renders the SLAM package available, passing a string which must be equal to that preceding "SLM port symbolic name" in the "one005.cnf" file.

The second directive activates the network server and is needed, therefore, only on the server machine in the net.

The third directive activates the File Transfer monitor.

The last directive activates the Batch Network Server (BNS) facility.

In the case of ONE for **SNA:**

```
START:SNAM=/IPL/USR/one/maindisp;
START:FTF=/IPL/USR/ftr/FTRmonitor;
START:BNS=/IPL/OWS/MOSNET/NETBOOT,<>(/IPL/OWS/MOSNET);
```

The first directive activates the program which allows the activation of user programs.

The second directive activates the File Transfer monitor.

The last directive activates the Batch Network Server (BNS) facility.

Queue Manager

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
```

The queue manager is used by any environment handling queues: batch, spooling system, Terminal Emulator, BEAM, MTS.

The directive need appear only once if more than one of them is present.

par is a dummy parameter as an initial string is required in a CALL directive but the queue manager requires no parameters.

Note that a particular version of the queue manager is provided for configurations comprising the **LMS** feature. In these cases, the above directive must be substituted by the following:

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUE_LMS,<LMS parameter list>;
```

where the **LMS parameter list** is that described above in the "LMS Local Management System" section.

Run Time Diagnostic

```
WSid:RTDIAG=/IPL/DPC/CMD/RTDIAG;
```

The run time diagnostic environment can be associated with any required work station.

Server Environment

SERV:/IPL/SYS/$SERVER,<T>nn<L>mm<N>b;

All distributed configurations require, on the server system, the activation of the server activator in the operating system, to allow remote object access (see the "Remote Component Access" section in Chapter 3) from the agent systems.

The three parameters are optional. For each of them, if it is missing in the directive, the relevant default value (specified below) is assumed.

The **T** parameter specifies the integer number of seconds between two request made by a Name Agent or Domain Name Server that are looking for a machine on the network to solve a naming problem. The default value is 50. A smaller value guarantees a faster connection to the network services.

The **L** parameter defines the number of remote requests which can be pending on the current machine (and whose execution is not immediately possible due to mutual exclusion reasons), that is, the number of remote file accesses simultaneously made on the current machine. The default value is 50.

The **N** parameter is a boolean value setting the notify mode for interconnected machines.

If N is set to 1, then the Name Agent of a machine is in notify state concerning its remote machine, while the Domain Name Server or the Global Name Server are in notify state concerning every system belonging to its distributed domain. To be in notify state means that an inquiry on the state of the other machine(s) is continuously carried out (by means of echo/ack-echo messages). When a machine is no longer connected to the network this technique detects the event immediately.

If N is set to 0, then the disconnection of a machine is not automatically detected. The inquiry on the state of a machine (and the update of the network state, if required) is only carried out when an access to an object resident on it fails. This "no notify" mode (N=0) is useful in distributed configurations with a high number of machines, in order not to overload the Domain Name Server or the Global Name Server for network monitoring only. The default value is 1.

Note: In a system where user packages are required, however, the server activator must be started using the PFG keyword, in order to allow the user packages to be loaded first (using the PCALL keyword), or using the FG keyword if there are user packages loaded via the CALL keyword.

Shell

```
WSid:MCL=/IPL/SYS/$VSH;
```

If a procedure or an executable program is to be activated by Shell at a specified work station, then the directive must contain the name of the procedure as parameters for Shell, and, optionally, a working directory:

```
WSid:MCL=/IPL/SYS/$VSH,<$cmd>proc(workdir);
```

If the login program has been changed from the default LOGS to LOGN, an extra directive is necessary for the terminal defined as master work station, or for those terminal where the possibility of inquiry is required (system line). This is because a login program with screen splitting is necessary:

```
WSid:LOGIN=.../LOGS;
```

SNA T5 NODE

```
START:T5=/IPL/DPC/T5NODE/STARTNODE
```

The SNA T5 node terminal concentrator has a specific start up program.

Sort/Merge Facilities

```
WSid:SORTRUN=/IPL/DPC/SORTRUN;
WSid:LANGUAGE=/IPL/DPC/LANGUAGE;
```

All sorting facilities are options of SORTRUN. The writing of procedures is done by selecting the LANGUAGE module.

Spooling System

```
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
PFG:/IPL/SYS/$SERVER;
START:/IPL/SP/SPGPA,<>n<>m<>...;
```

The queue manager is used by both the spooling system and the batch monitor; the directive (first above) need appear only once if both are present.

par is a dummy parameter as an initial string is required in a CALL directive but the queue manager requires no parameters. See also the "Queue Manager" section above.

The second directive is only needed in distributed configurations.

The third directive starts the spooling system.

The **n, m, ...** optional parameters are integer numbers identifying the spooling classes in the current spooling system. In a distributed configuration there can be up to eight queues and up to eight spooling systems. Each queue can only belong to one spooling system (which, on the other hand, can handle more than one queue). If no spooling classes are specified, eight classes are assumed by default.

2780/3780 Terminal Emulator

```
CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par1;
CALL:WBF=/IPL/DPC/$EMU/WBF,<>par2;
START:SERVER=/IPL/SYS/$SERVER;
START:SPGPA=/IPL/SP/SPGPA,<>n<>m;
START:EMUGPA=/IPL/DPC/$EMU/EMUGPA;
WSid:OBF=/IPL/DPC/$EMU/MAIN;
WSid:LOGIN=/IPL/DPC/CMD/LOGN;
```

These directives allow the terminal emulators software environment to be started together with the Queue Manager and the Spooler.

par1 is a dummy parameter as an initial string is required in a CALL directive but the queue manager requires no parameters. See also the "Queue Manager" section above.

par2 is a dummy parameter to provide the compulsory initial string for the work station batch facility package.

Activating the main program of the terminal emulator causes the associated terminal to operate as if it were a 2780 or 3780, sending batch jobs to an IBM mainframe.

The optional last directive is usually necessary because LOGN is not the standard (default) login program, and the emulator requires a login program without screen splitting.

```
PINIT:/IPL/DPC/CMD/WAITNET;
PCALL:LUINTERFACE=/IPL/DPC/$E3270/LUINTERFACE,<>par;
SERV:/IPL/SYS/$SERVER;
PSTART:DRIVER=/IPL/DPC/$E3270/DRIVER;
WSid:E3270=/IPL/DPC/$E3270/P3270x;
WSid:SHUTDOWN=SHUTDOWN;
```

The first directive above is only needed on satellite systems.

In the second directive, **par** is a dummy parameter to provide the compulsory initial string for the package which handles interactions between terminals and printers.

The third directive is only required on master systems in distributed configurations.

The Environment Printers manager program is activated in background.

In the fifth directive the lower case 'x' must be replaced with one of the following characters:

- **D** if the WSid is equipped with a DP keyboard

- **M** if the WSid is equipped with a multifunctional keyboard

The shutdown directive is required on stand alone systems. In distributed configurations, it must be modified as follows:

WSid:SHUTDOWN=SHUTDOWN!GMASTER; for master systems, or

WSid:SHUTDOWN=SHUTDOWN!MASTER; for satellites systems.

Note: The emulator requires a login program with screen splitting. In the above example LOGS is assumed to be the default login program.

SYSTEM STARTUP AND RELATED ERRORS

When the system is started up, the following sequence of initial actions is carried out by Grandpa:

1. The configuration file, $CONFGP, is read into internal tables.

2. The master work station for the system session is established.

3. The initial dialogue with the master work station takes place: IPL warnings are output; date and time are optionally set; the system disk is optionally removed.

4. User programs and packages are loaded and activated in the appropriate order.

During this phase, various error conditions may arise, as listed below:

1. Grandpa cannot find $CONFGP in the /SYS volume or there is a read error. The following message is displayed:

   GPA CAN'T ACCESS $CONFGP

   To permit recovery, the NEWVOL function is automatically called at this point, so that the operator is prompted:

   ENTER DIRNAME FOR NEW DISK

   If no system disk with $CONFGP in the system volume /IPL/SYS can be supplied, the system cannot be used.

2. There is a failure to complete the initial actions (no suitable functioning work station is available after all possibilities have been tried). The system cannot be used.

3. The remount following system disk removal fails. The error message displayed is:

   ERROR IN MOUNTING DISK

   An alternative is then requested using the original NEWVOL prompt:

   ENTER DIRNAME FOR NEW DISK

4. There is a failure in the activation of some program or package. This results in the message:

   WARNING: GPA UNABLE TO CALL/START <name>
   in particular, WARNING: GPA UNABLE TO CALL/START LOGIN PROGRAM

   The session is not aborted, it just proceeds without the component. If this is the login program, the system cannot be used.

CONFIGURABLE APPLICATION ENVIRONMENTS

The following application environments available to MOS users are configurable:

- MTS (Modular Transactional Support)
- BEAM (Business Environment Application Monitor)
- Terminal emulator for 2780, 3780, 3270 SNA or 3270 BSC
- NMS (Network Management System)
- ESE environment
- ONE (Open Network Environment)
- GATEWAY COSMOS-MOS
- OLILAN (Local Area Network)
- L1WSE (Line 1 Work Station Emulator)
- OLIEMU work station emulator

These environments must be configured before they are used. For the first three a configuration utility is available (as part of the relevant functional sets) to create the necessary configuration file. Each utility is documented as follows:

- CONFMTS for **MTS** is described in:

  MTS Configuration Guide - Code 4003450 L

- BEAMCONF for **BEAM** is described in:

  BEAM User Guide - Code 4003180 R

- CONFOBF for the **terminal emulators** is described in:

  Terminal Emulators, Configuration Guide - Code 4000930 U

The reader is referred to the manual concerning the application environment he wishes to configure.

For the **NMS** environment, the reader is referred to the manuals:

NMS, CMS Component, Operating Guide - Code 4000980 H
NMS, LMS Agent Component, Operating Guide - Code 4000960 X

For the **ESE** environment, the reader is referred to the manual:

ESE Environment Software Installation User Guide - Code 3985500 P

For the **ONE** environment, the reader is referred to the manual:

ONE Open Network Environment Network Management Guide - Code 4008550 C

For the **GATEWAY COSMOS-MOS,** the reader is referred to the manual:

CSS Generation and Configuration, User Guide - Code 4000750 H

For the **OLILAN** environment, the reader is referred to the manual:

OLILAN, Local Area Network, User Guide - Code 4021820 R

For the **L1WSE, OLIEMU** and **WSELAN** emulation environments, the reader is referred to Part V of the current manual.

5

PART V : CONFIGURATION AND INSTALLATION COMMANDS

INTRODUCTION TO PART V

The Shell commands used during the configuration and installation processes are described in this part of the manual in alphabetical order for reference purposes.

11. COMMAND DESCRIPTIONS

The commands described in this chapter, and the other chapters in which they are referred to are as follows:

- CONF : configure a PC to run the L1WSE program
- CONFMAN : display or modify existing configurations
- INSTALLQM : install spooling/batch systems (Chapters 8, 9)
- KEYCONF : create or modify a keyboard table
- LINKDEV : link devices to form a single logical device
- MKBOOT : copy bootstrapper onto system disk (Chapters 8, 9)
- MKLOGIN : activate user login mechanism (Chapters 8, 9)
- MKSYS : dump a system volume onto SCT or MTU (Chapter 8)
- MODCON : display, print or modify the $CON file
- OLICONF : configure a PC to run the OLIEMU program
- PARSER : create user environment configuration file (Chapter 10)
- REBUILD : regenerate the $CHM file, the $IPL file or the SYS volume (Chapter 6)
- SPCONF : install and configure the spooling system (Chapters 8, 9)
- WSECONF : configure a PC to run the WSELAN program

The CONF command configures the L1WSE emulator.

It is a menu-driven interactive utility with options to:

- define the type of connection between the PC and the L1 MOS system, and the relevant line parameters
- define the work station printer's parameters, if any
- define the other work station devices' parameters, if any
- select the version and type of keyboard table
- define further parameters, such as type of PC, type of emulated screen, etc.

CONF

CONF must be run on the PC to be used as an L1 work station, and it must reside in the MS-DOS directory containing the L1WSE emulator program.

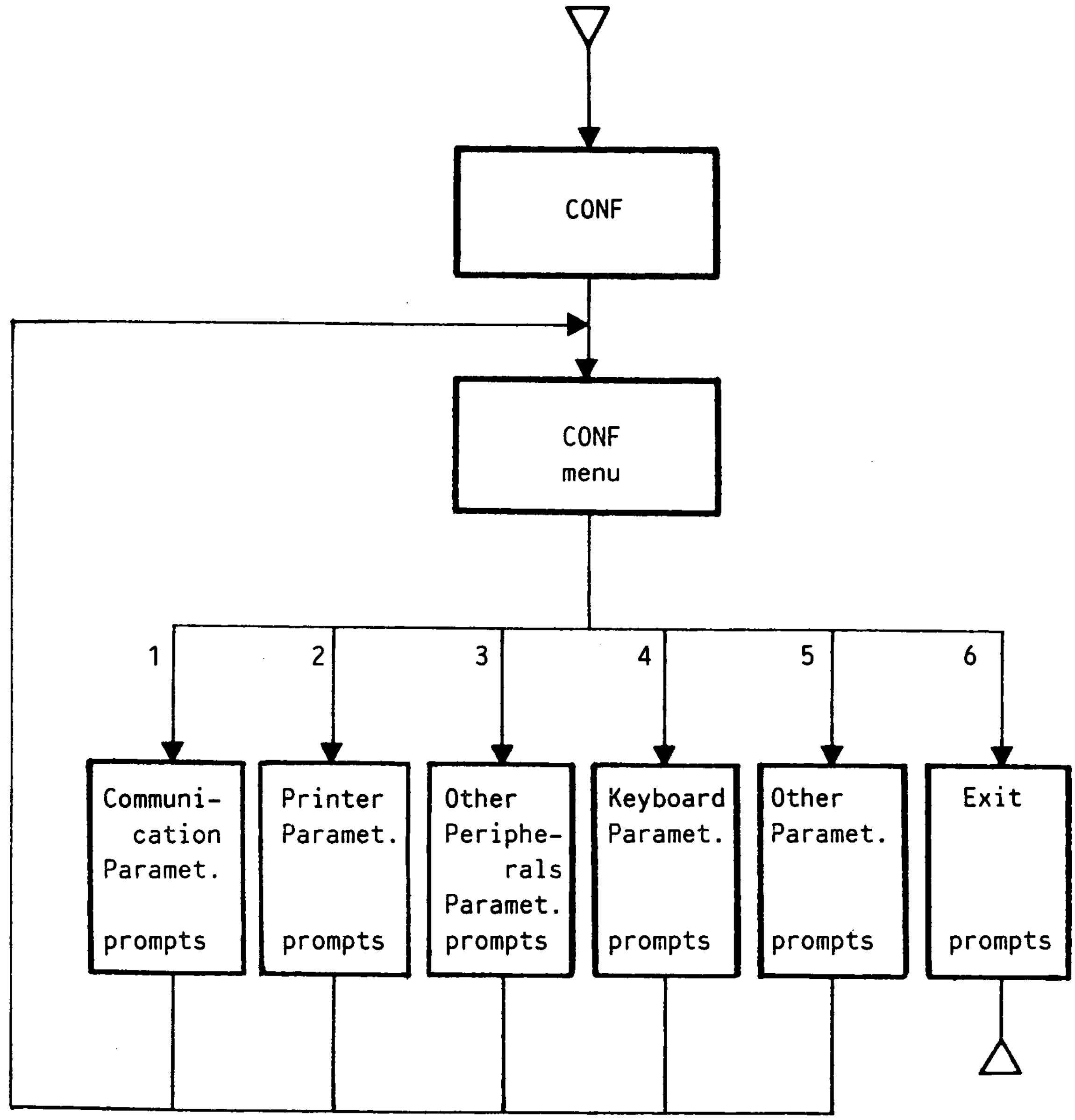


Fig. 11-1 General Logic of the CONF Command

CONF FUNCTIONING

The CONF menu offers six options:

1. to define the type of connection between the PC and the L1 MOS system and the relevant line parameters

2. to specify whether a work station printer is present and to define its interface type

3. to specify whether other work station devices are present and to define the relevant parameters

4. to specify the version and type of the keyboard table to be used

5. to specify the type of screen to be emulated (alphanumeric or graphic), the time out in receive mode, and whether the PC is an M19

6. to end the CONF run and return to MS-DOS.

The general logic of the program is shown opposite. Choosing an option (except **6**) gives rise to a single display or sequence of displays, which either prompt for information or present stored information.

Simple error messages appear if responses to prompts have invalid syntax or are incorrect, and the user must correct his input before he can continue. Each response accepted causes the next prompt to appear. The different displays are described below.

Note: Each menu is presented with default values. The last row of the menu itself asks if one or more of the displayed values is to be modified, as follows:

set up parameters (Yes-No) =

The user may:

- enter **Y** in reply to the set up prompt and return to the first question of the menu, where the default values can either be confirmed (entering **/CR/**) or modified as required

- enter **N** in reply to the set up prompt, thus confirming all the displayed values and giving rise to the next CONF menu.

OPTION 1 (COMMUNICATION PARAMETERS) PROMPTS

```
          **********  CONFIGURATION PARAMETERS  **********

async. comm. port ----------------------(1=motherb, 2=extens)  =  1
speed ------- (50-75-110-150-300-600-1200-2400-4800-9600-19200)  =  9600
current loop ---------------------------------------- (Yes-No)  =  N
ignore DSR ------------------------------------------ (Yes-No)  =  N
ignore CTS ------------------------------------------ (Yes-No)  =  N
parity ---------------------------------------(None-Odd-Even)  =  N
stop bit -----------------------------------------------(1-2)  =  1
character length ------------------------------------(5-6-7-8)  =  8
Transmit Control ---------------------(None-Xon-Break-DSR-CTS)  =  N
Receive  Control ---------------------(None-Xon-Break-DTR-RTS)  =  X
```

Current values for the RS232 line used to connect the PC to the L1 MOS system are displayed. These values can be either confirmed or modified, bearing in mind that:

- values entered are only accepted if belonging to the admitted ranges (which are displayed in brackets). Agreement between them and those given when configuring the RS232 line in question on the L1 MOS system is the the user's responsibility

- the recommended value for speed is 9600 bps (19200 bps can only be used if CONTSW will not be used)

- if the Current Loop connection is used, the DSR and CTS signals are ignored

- the character length must be 8 bits (as required on the L1 MOS system)

- the Xon(/Xoff) control cannot be specified in transmission.

OPTION 2 (PRINTER PARAMETERS) PROMPTS

No printer is currently supported. An acoustic warning is emitted if this option is selected, and another option can be chosen.

OPTION 3 (OTHER PERIPHERALS PARAMETERS) PROMPTS

No other peripherals are currently supported. An acoustic warning is emitted if this option is selected, and another option can be chosen.

OPTION 4 (KEYBOARD SELECTION PARAMETERS) PROMPTS

```
National keyboard -(DENMARK-FINLAND-FRANCE-GERMANY-HEBREW-ITALY-NORWAY
                    PORTUGAL-SPAIN-SPAIN2-SWEDEN-SWITZERLAND-UK-USA-) = USA

Keyboard layout ---------------------------(0 = PC   1 = PB 105 keys) = 0
```

aa...a/CR/	The name of the national keyboard used on the PC must be entered.
n	The type of keyboard used on the PC must be selected (entering 0 if it is a standard PC keyboard, or 1 if it is the PB keyboard with 105 keys).

OPTION 5 (OTHER PARAMETERS) PROMPTS

---

alpha/graphics terminal ------------------------------------(A-G) = A

timeout in receive (0 = infinite wait) ------------------------- = -1

M19 personal computer -----------------------------------(Yes/NO) = N

a The character identifying the type of emulated terminal: either A (if no graphics feature are required) or G (if graphics are used).

n/CR/ The number of polling cycles that the emulator must execute, when it is waiting for a character needed for a file transfer or a command, before signalling an error in reception. Accepted values are in the range from -1 to 65536, where -1 (which is the suggested value) means 65536 and 0 means infinite wait.

a Either Y, if the PC used is an M19, or N, if it is another PC.

OPTION 6 (EXIT)

Choosing this option ends the CONF run and returns to MS-DOS.

The CONFMAN menu-driven interactive utility operates on an existing configuration and allows the user to:

- display the Release messages contained in all the 1-modules present in the /IPL/SYS volume
- display and modify the current values of the system sizing parameters
- display and modify the current values of the magnetic peripherals parameters
- display and modify the current values of the parameters relevant to work stations, system printers and RS232/CL driver
- write any modification made into the specified MOS configuration file.

CONFMAN must be run in the directory containing the standard keyboard tables (that is, the command **SETWDIR /IPL/RELn.l/CONF_GEN/SC/KBTABLES** , where **n** is the release number and **l** is the release level, must be entered before running CONFMAN on the "mostro" machine; or, on a customer system, the working directory must be set to the directory containing the keyboard tables to be used).

CONFMAN pathname

---

where:

pathname can be:

- the complete path name (starting with "/" or relative to the current working directory) of a MOS configuration file (that is a $CON file produced by SYSCONF or SYSCONFR)
- the complete path name (starting with "/" or relative to the current working directory) of a MOS system volume (that is a SYS volume) which:

  1. must contain the $CON configuration file
  2. can contain system modules (whose list of contents may be displayed using the appropriate CONFMAN option).

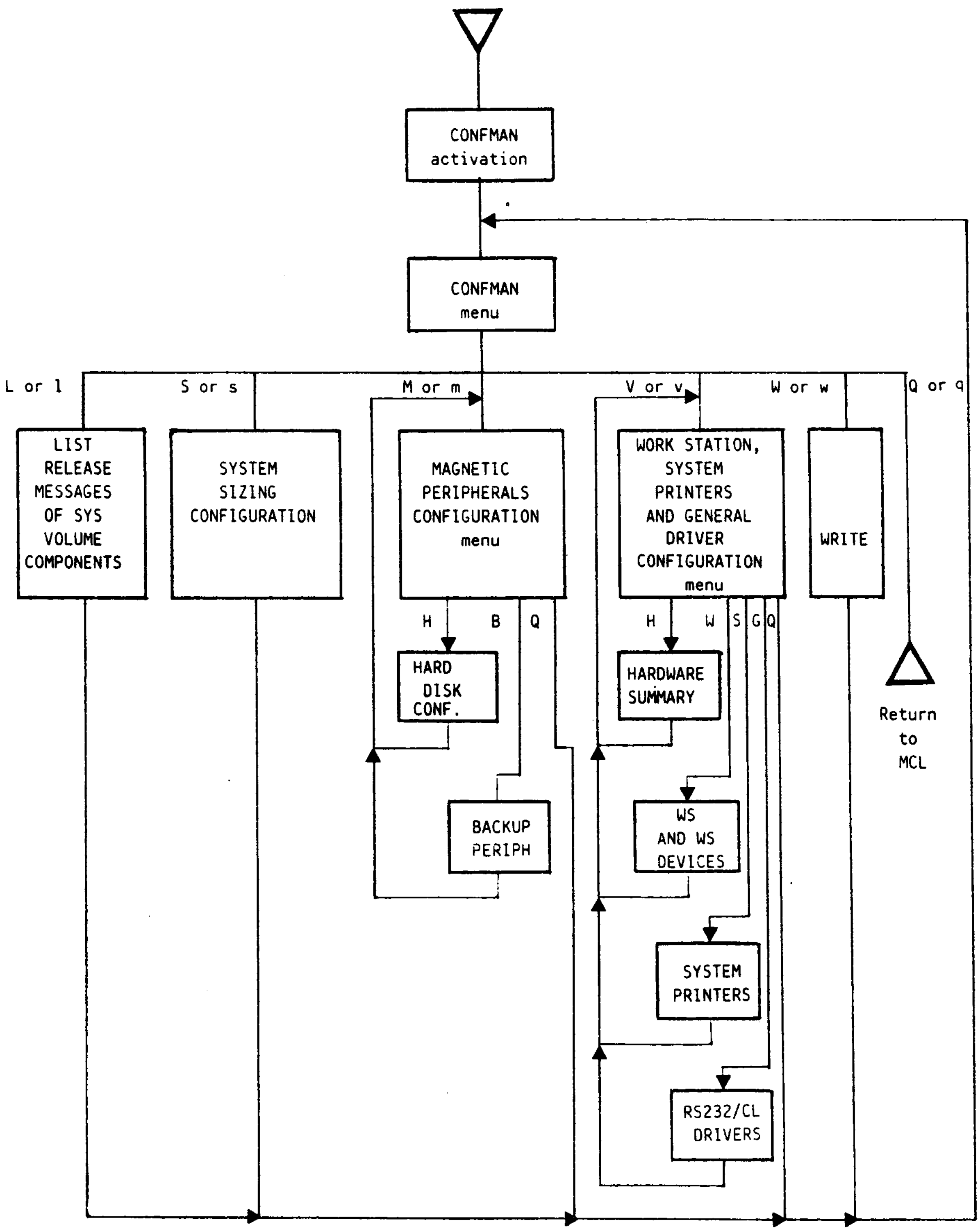


Fig. 11-2 General Logic of the CONFMAN Command

CONFMAN FUNCTIONING

Once activated, CONFMAN displays a menu which offers six options:

L or l	Displays the Release messages which are contained in each l-module present in the SYS volume.
S or s	Displays the current values for parameters relevant to system sizing and allows them to be modified.
M or m	Displays the current values for parameters relevant to magnetic peripherals and allows them to be modified.
V or v	Displays the current values for parameters relevant to work stations, system printer and RS232/CL driver, and allows them to be modified.
W or w	Any modification made in the current CONFMAN session is written into the $CON file.
Q or q	Exit.

The general logic of the program is shown opposite. Choosing an option (except Exit) gives rise to a single display or sequence of displays, which present the current configuration and permit it to be modified.

Modifications are not written directly to the $CON file, but to a copy residing in memory.

The option **W** or **w** must always be selected, after having modified some parameters' values, in order to write on disk the resulting $CON file.

The different options are described below.

OPTION L or 1 (DISPLAY RELEASE MESSAGES OF 'SYS' VOLUME COMPONENTS)

This option can only be selected if the path name entered when CONFMAN has been activated refers to a 'SYS' volume: in this case, the Release messages contained in each l-module present in such volume are displayed as shown below:

```
            Olivetti MOS  Configuration Manager  Rel. 5.2

              RELEASE MESSAGE OF SYS VOLUME COMPONENTS


                <l-module name> : <l-module message>

                <l-module name> : <l-module message>

                <l-module name> : <l-module message>

                <l-module name> : <l-module message>

                      ...                 ...

                                 Continue with CR:
```

OPTION S or s (DISPLAY/MODIFY SYSTEM SIZING CONFIGURATION)

This option displays the current values of the parameters relevant to system sizing (UNITs 1, 10, 50, 52 and 53).

Olivetti MOS Configuration Manager Rel. 5.2

SYSTEM SIZING

N. OF FAMILIES	30	BUFFER SIZE	512	MULTIPLE FILES	15
N. OF PROCESSES	30	HW Seg Num	2	PACKED POS FILES	2
N. OF PROGRAMS	60	DAM SIZE	300512	KEYED FILES	15
N. OF CHANNELS	30	PRIVATE BUFFER P	3	LOCKED RECORDS	15
N. OF FILES	230	NR OF PRIVATE BUF	3	LOCKED PHANTOM	15
N. OF SYSTEM I.D.	30	MEM VOLUME PAGES	256	SECONDARY INDICES	5
SYSTEM BUFFERS	30	FILES IN MEM VOLUME	100	MMU NUMBER	2

ALL VALUES CORRECT ? (Y/N) :

Help Messages

The user may, if necessary, modify the displayed values by replying **N** or **n** to the **ALL VALUES CORRECT ? (Y/N) :** prompt. In this case the cursor is positioned on the first value, an help message is displayed which explains the meaning of the parameter, and the user may enter the new value (or confirm the current one, entering /CR/). These steps are repeated for each parameter of the menu.

Note that a $CON file for a system equipped with one MMU can be made suitable for a system with two MMUs (and viceversa), just by modifying the "MMU number" value (besides any other parameter's values modification needed). Conversion of a $CON file from two MMUs to one MMU is only accepted if no more than seven segments are defined for the D.A.M. and the hardware pool.

OPTION M or m (DISPLAY/MODIFY MAGNETIC PERIPHERALS CONFIGURATION)

This option displays a menu which offers three options:

```
              Olivetti MOS  Configuration Manager  Rel. 5.2

                   - H -  HARD DISK CONFIGURATION

                   - B -  BACK_UP PERIPHERALS

                   - Q -  RETURN TO MAIN MENU

                                     SELECT:
```

The user can enter:

H or h to display information relevant to the hard disk configuration

B or b to display information relevant to the back up peripherals

Q or q to return to the main CONFMAN menu.

Information displayed when one of the first two options are selected are described below.

Option H or h

If the first option (**H** or **h**) is selected in the "DISPLAY/MODIFY MAGNETIC PERIPHERALS CONFIGURATION" menu, information relevant to the configured hard disk is displayed through a menu, as shown in the following example:

Olivetti MOS Configuration Manager Rel. 5.2

HARD DISK CONFIGURATION

Disk Type : aa..a

Physical Unit	Present	Logical Group
1	Y	x
2	Y	y
3	N	
4	Y	z

DO YOU CONFIRM? (Y/N) :

---

where:

Disk Type indicates the type of hard disk currently included in the configuration; possible types are SMD, SASI, ESDI, WREN, INTEGRATED or NONE (if no hard disk is present).

Present specifies whether the indicated hard disk is present (Y) or not (N).

Logical Group indicates the logical name of each physical unit: if some hard disks have been linked, they have the same logical name (see the "LINKDEV" command in Part V and the "LOGHD" parameter in Chapter 4).

The user can enter **Y** or **y** in response to the **DO YOU CONFIRM? (Y/N) :** prompt in order to confirm the displayed values, or **N** or **n** to modify them: in this case the cursor is placed at the first field of the menu and every value can be either modified or confirmed with **/CR/.**

Option B or b

If the second option (**B** or **b**) is selected in the "DISPLAY/MODIFY MAGNETIC PERIPHERALS CONFIGURATION" menu, information relevant to the configured back up peripherals is displayed through a menu, as shown in the following example:

```
              Olivetti MOS  Configuration Manager  Rel. 5.2

                     BACK_UP PERIPHERALS CONFIGURATION

FD present            (Y/N):        Magnetic Tape             (Y/N):
Physical Unit Number  (1/2):

MFD 1M present        (Y/N):        Streaming Cartridge Tape  (Y/N):
Physical Unit Number  (1/2):

MFD 320K present      (Y/N):
Physical Unit Number  (1/2):

                                     DO YOU CONFIRM? (Y/N) :
```

---

where:

Present specifies whether the indicated floppy disk unit (8" diskette unit, 1Mbyte 5 1/4" diskette unit, 320 Kbyte 5 1/4" diskette unit) is present (Y) or not (N).

Physical Unit Number is the number of the floppy disk physical unit (1 or 2).

Magnetic Tape specifies if the tape unit is present (Y) or not (N).

Streaming Cartridge Tape specifies if the SCT unit is present (Y) or not (N).

The user can enter **Y** or **y** in response to the **DO YOU CONFIRM? (Y/N) :** prompt in order to confirm the displayed values, or **N** or **n** to modify them: in this case the cursor is placed at the first field of the menu and every value can be either modified or confirmed with **/CR/.**

Option Q or q

If the third option (**Q** or **q**) is selected in the "HARD DISK CONFIGURATION" menu, the CONFMAN main menu is displayed again.

OPTION V or v (DISPLAY/MODIFY WORK STATION, SYSTEM PRINTERS AND RS232/CL DRIVER CONFIGURATION)

This option displays a menu which offers five options:

```
                Olivetti MOS  Configuration Manager  Rel. 5.2

     WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVER CONFIGURATION

          - H -  SHOW HARDWARE SUMMARY AND MODIFY CONFIGURATION

          - W -  SHOW WORK STATIONS AND WORK STATION DEVICES

          - S -  SHOW SYSTEM PRINTERS

          - G -  SHOW GENERAL DRIVERS

          - Q -  RETURN TO MAIN MENU

                                              SELECT:
```

The user can enter:

H or h to display information relevant to the hardware connections (that is via KDCs, TWIN boards, CPU RS232, MUXs) and, if necessary, to modify the current configuration

W or w to display information relevant to the work stations and their devices

S or s to display information relevant to the system printers

G or g to display information relevant to the RS232/CL driver

Q or q to return to the main CONFMAN menu.

Option H or h

If the first option (**H** or **h**) is selected in the "WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVERS CONFIGURATIONS", information relevant to the configured devices connected via KDCs, TWIN boards, CPU RS232 and MUXs is displayed through three menus, as shown in the examples below:

```
                Olivetti MOS  Configuration Manager  Rel. 5.2

   TWIN                          INTEGRATED OR VIA ELB CONNECTION

 Chn    device               ws   sc_kb  ELB1382  ChnA  ChnB   badge  pinpad

TW1 A   GD1          KDC1   WS1     Y       N      CA1   MC1    BG1    PP1
    B   ---
                     KDC2   WS2     Y       N      ---   ---    ---    ---
TW2 A   ---
    B   ---          KDC3   WS3     Y       Y      WP1   ---    ---   ---

TW3 A   ---          KDC4   ---     N       N      ---   ---    ---   ---
    B   ---
                     KDC5   ---     N       N      ---   ---    ---    ---
TW4 A   ---
    B   ---          KDC6   ---     N       N      ---   ---    ---    ---

TW5 A   ---          KDC7   ---     N       N      ---   ---    ---    ---
    B   ---
                     KDC8   ---     N       N      ---   ---    ---    ---
RSCPU   SP1

Paging :  <+> Next Page        <H> Help for ADD        <Q> to quit
Actions : ADD WS TO TWINnc/KDCnc/MUXnlT/MUXnlE/RSCPU
          DEL WSn
Command:
```

where:

- devices connected to TWIN boards or to the RS232 interface of the CPU board are indicated with a two-characters code and a number. Codes are as follows:

  WS = work station
  WP = work station printer
  SP = system printer
  GD = RS232/CL driver (general driver)
  CA = cash adapter
  MC = cheque reader
  BG = badge reader-reader/writer
  PP = PIN Pad

- the work station identifier (WSn), the screen-keyboard presence (sc_kb = Y) or absence (sc_kb = N), the ELB 1382 presence (ELB1382 = Y) or absence (ELB1382 = N), devices connected to channels A and B of the ELB 1382 (for example, CAn, WPn, ...), the badge reader or reader/writer presence (badge = BGn) or absence (badge = ---), and the PIN Pad presence (pinpad = PP1) or absence (pinpad = ---) are indicated for those devices which are connected to KDCs.

See the section "Update Commands" below for a description of how the current configuration can be modified.

Olivetti MOS Configuration Manager Rel. 5.2

WORK STATION : MUX CONTROLLERS AND PERIPHERALS (UNIT 05)

	Line	device	WSL1	sc_kb	Chn A	Chn B	badge	pinpad
MUX1	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX2	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX3	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX4	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---

```
Paging :  <+> Next Page       <-> Previous Page       <E> Exit
Actions : ADD WS/WP/CA/MC TO MUXnlT/MUXnlE
          DEL WSn/SPn/GDn
Command:
```

where, for each MUX line, the two-character code of the connected device (WS, SP or GD), the WSL1 presence (WSL1 = Y) or absence (WSL1 = N), the screen-keyboard presence (sc_kb = Y) or absence (sc_kb = N), devices connected to channels A and B of the ELB 3683/3684 (CA, MC), the badge reader or reader/writer presence (badge = BGn) or absence (badge = ---), and the PIN Pad presence (pinpad = PPn) or absence (pinpad = ---) are indicated.

Each menu displays information concerning four MUXs.

See the section "Update Commands" below for a description of how the current configuration can be modified.

Olivetti MOS Configuration Manager Rel. 5.2

WORK STATION : MUX CONTROLLERS AND PERIPHERALS

	Line	device	WSL1	sc_kb	Chn A	Chn B	badge	pinpad
MUX5	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX6	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX7	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---
MUX8	0	---	N	N	---	---	---	---
	1	---	N	N	---	---	---	---
	2	---	N	N	---	---	---	---
	3	---	N	N	---	---	---	---

```
Paging :  <+> Next Page       <-> Previous Page      <E> Exit
Actions : ADD WS/WP/CA/MC TO MUXnlT/MUXnlE
          DEL WSn/SPn/GDn
Command:
```

See the section "Update Commands" below for a description of how the current configuration can be modified.

If a "ADD WS TO ..." update command was entered during the current "H session" (that is, in one of the previous three menus), then the following four menu are also displayed.

```
          Olivetti MOS  Configuration Manager  Rel. 5.2

            CHARACTERISTICS OF WORK STATION (UNIT 05)

 ws    contr.    OlivettiT.   conn.    speed    screen    keyboard

WS1    KDC1E0        Y        RSANL     4800      AL      USAASCII_DP
WS2    KDC2E0        Y        RSANL     4800      AL      USAASCII_DP
WS3    KDC3E0        Y        RSANL     4800      AL      USAASCII_DP


TYPE:   <+>  to confirm this menu and continue
        <->  to display and/or modify the previous menu
        <&>  to modify this menu
        <q>  to quit
ACTION:
```

where:

ws	is the work station identifier
contr.	is the symbolic name of the controller to which the work station is connected
OlivettiT.	indicates whether the work station is an Olivetti (Y) or not (N)
conn.	is the type of connection (RSNNL = RS232 without automatic new line feature, RSANL = RS232 with automatic new line feature, CLNNL = Current Loop without automatic new line feature, CLANL = Current Loop with automatic new line feature)
speed	is the transmission/reception speed in baud
screen	is the screen type (AL = alphanumeric, GR = graphic black and white, GC = graphic colour, KA = Kanji)
keyboard	is the national keyboard table associated with the work station.

When modifying the existing configuration, if a new work station has been added, the last row of this menu presents it, with default values where possible. The values in the first two columns identify it, while the other values are displayed by CONFMAN according to values entered for the first fields.

In order to help the user in entering correct values, a brief description of the field, as well as the admitted range of values, is displayed at the bottom of the menu.

The default value displayed by CONFMAN is confirmed by entering **/CR/.**

An error made when entering a value is signalled by an acoustic warning, and another value can be entered.

The user can confirm each row of the menu (entering **+**), modify it (entering **&**), modify values in the previous menu (entering **-**), or terminate the modification session (entering **Q** or **q**).

Olivetti MOS Configuration Manager Rel. 5.2

WORK STATION PRINTERS

ws	contr.	HcpPRT	printer	light	conn.	mode	Hc.Dev	speed
WS1	KDC1E0	WP1	---	-	---	---	-	---
WS2	KDC2E0	WP1	---	-	---	---	-	---
WS3	KDC3E0	WP1	WP1	1	RS232	FREE	1	2400

```
TYPE:   <+>  to confirm this menu and continue
        <->  to display and/or modify the previous menu
        <&>  to modify this menu
        <q>  to quit
ACTION:
```

where:

ws — is the work station identifier

contr. — is the symbolic name of the controller to which the work station is connected

HcpPRT — indicates which hard copy printer is available for the work station (n = no hard copy printer available)

printer — is the work station printer

light — specifies the printer LED used to indicate that the printer is being used by the current work station (for shared printers only)

conn. — is the type of connection (RS232 = RS232 or CURR = Current Loop)

mode — indicates whether the printer is handled in Olivetti-controlled (CONT) or free running (FREE) mode

Hc.Dev — is the hard copy subdevice (0 = no hard copy, 1 = hard copy on default subdevice, 2 = hard copy on Automatic Front Feed, 3 = hard copy on Semi-automatic Front Feed, 4 = hard copy on Auto Sheet Feeder)

speed — is the transmission/reception speed in baud

Any required modification can be carried out in the same way as described for the "CHARACTERISTICS OF WORK STATION (UNIT 05)" menu.

```
             Olivetti MOS  Configuration Manager  Rel. 5.2

             MICRS, CADPS, BADGES AND PIN PADS INFORMATION

 ws   contr.  Micr Type speed  Cadp Type speed  Badge speed  PinP  speed

WS1   TW 1A0   MC1   0   4800   CA1   0  19200    Y   134.5   Y    1200
WS2   KDC1E0   ---   -   ---    CA1   0  19200    N     -     N    ---


TYPE:   <+>  to confirm this menu and continue
        <->  to display and/or modify the previous menu
        <&>  to modify this menu
        <q>  to quit
ACTION:
```

where:

ws	is the work station identifier
contr.	is the symbolic name of the controller to which the work station is connected
Micr	indicates the Cheque reader or Cheque reader/printer (MC1, MC2, ...)
Type	is the type of Cheque reader (0 = reader, 1 = reader/printer)
speed	is the transmission/reception speed in baud
Cadp	indicates the Cash Adapter (CA1, CA2, ...)
Type	is the type of Cash Adapter (0 = normal, 1 = teller)
speed	is the transmission/reception speed in baud
Badge	indicates whether the Badge reader is present (Y) or not (N)
speed	is the transmission/reception speed in baud
Pinpad	indicates whether the PIN Pad is present (Y) or not (N)
speed	is the transmission/reception speed in baud

Any required modification can be carried out in the same way as described for the "CHARACTERISTICS OF WORK STATION (UNIT 05)" menu.

```
             Olivetti MOS  Configuration Manager  Rel. 5.2

                          CHANNEL PARAMETERS


 ws contr. input output stop par ch XOFF XON tand tand break sign.  DTR RTS
                        bit                  sett mode sign. status
WS1 TW 1A0  128    256   1   EV  7   96  64   1    RT  MARK    0     1   1
WP1 TW 5A0  128    256   1   EV  7   96  64   1    RT  MARK    0     1   1



TYPE:   <+>  to confirm this line          <-> to modify the previous line
        <*>  to display and/or modify the previous menu
        <&>  to modify the current line
        <q>  to quit
ACTION:
```

where:

ws	is the work station identifier
contr.	is the symbolic name of the controller to which the work station is connected
input	is the size in bytes of the input buffer (it must be 16 for PIN Pads and 110 for Badge readers)
output	is the size in bytes of the output buffer (it must be 16 for PIN Pads and 110 for Badge readers)
stop bit	is the number of stop bits (0 = 1 bit, 1 = 1.5 bits, 2 = 2 bits; the value 0 is mandatory for PC connection)
par	is the parity check (NP = no parity check requested, OD = odd parity, EV = even parity)
ch	is the number of bits per character (0 = 5 bits, 1 = 6 bits, 2 = 7 bits, 3 = 8 bits; the value 3 is mandatory for PC connection)
XOFF	is the XOFF level (in number of bytes)
XON	is the XON level (in number of bytes)
tand sett	is the tandem mode setting (0 = reset, 1 = XON/XOFF, 2 = BREAK, 3 = CTS)

tand mode	specifies whether the tandem mode is supported in reception (R), in transmission (T) or in both (RT)
break sign.	is the state of the break signal (MARK, SPACE or NULL)
sign status	is the state of a general signal which depends on the peripheral (0 = logical 0, 1 = logical 1, 2 = no operation)
DTR	is the state of the Data Terminal Ready signal (0 = logical 0, 1 = logical 1, 2 = no operation)
RTS	is the state of the Request To Send signal (0 = logical 0, 1 = logical 1, 2 = no operation)

Any required modification can be carried out in the same way as described for the "CHARACTERISTICS OF WORK STATION (UNIT 05)" menu, but the accepted commands are:

+ to confirm each row of the menu

& to modify each row of the menu

- to modify values in the previous row

* to modify values in the previous menu ("MICRS, CADPS, BADGES, AND PIN PADS INFORMATIONS")

Q to terminate the modification session.

Note that confirmation of the last row of this menu terminates the addition phase for a new work station.

Option W or w

If the second option (**W** or **w**) is selected in the "WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVERS CONFIGURATION" menu, information relevant to work stations, work station printers, work station devices (such as cheque reader or reader/printer and cash adapter), and channels used for their connections is displayed through four menus, as shown in the examples below:

Olivetti MOS Configuration Manager Rel. 5.2

CHARACTERISTICS OF WORK STATION (UNIT 05)

ws	contr.	OlivettiT.	conn.	speed	screen	keyboard
WS1	KDC1E0	Y	RSANL	4800	AL	USAASCII_DP
WS2	KDC2E0	Y	RSANL	4800	AL	USAASCII_DP
WS3	KDC3E0	Y	RSANL	4800	AL	USAASCII_DP

TYPE: <+> to continue
<q> to quit
ACTION:

---

where:

ws — is the work station identifier

contr. — is the symbolic name of the controller to which the work station is connected

OlivettiT. — indicates whether the work station is an Olivetti (Y) or not (N)

conn. — is the type of connection (RSNNL = RS232 without automatic new line feature, RSANL = RS232 with automatic new line feature, CLNNL = Current Loop without automatic new line feature, CLANL = Current Loop with automatic new line feature)

speed — is the transmission/reception speed in baud

screen — is the screen type (AL = alphanumeric, GR = graphic black and white, GC = graphic colour, KA = Kanji)

keyboard — is the national keyboard table associated with the work station.

The user can continue in the display phase (entering **+**) or terminate it (entering **Q** or **q**).

Olivetti MOS Configuration Manager Rel. 5.2

WORK STATION PRINTERS

ws	contr.	HcpPRT	printer	light	conn.	mode	Hc.Dev	speed
WS1	KDC1E0	WP1	---	-	---	---	-	---
WS2	KDC2E0	WP1	---	-	---	---	-	---
WS3	KDC3E0	WP1	WP1	1	RS232	FREE	1	2400

TYPE: <+> to continue
<-> to display the previous menu
<q> to quit
ACTION:

where:

ws — is the work station identifier

contr. — is the symbolic name of the controller to which the work station is connected

HcpPRT — indicates which hard copy printer is available for the work station (- = no hard copy printer available)

printer — is the work station printer

light — specifies the printer LED used to indicate that the printer is being used by the current work station (for shared printers only)

conn. — is the type of connection (RS232 = RS232 or CURR = Current Loop)

mode — indicates whether the printer is handled in Olivetti-controlled (CONT) or free running (FREE) mode

Hc.Dev — is the hard copy subdevice (0 = no hard copy, 1 = hard copy on default subdevice, 2 = hard copy on Automatic Front Feed, 3 = hard copy on Semi-automatic Front Feed, 4 = hard copy on Auto Sheet Feeder)

speed — is the transmission/reception speed in baud

The user can continue in the display phase (entering **+**), display again the previous menu (entering **-**), or terminate the display phase (entering **Q** or **q**).

Olivetti MOS Configuration Manager Rel. 5.2

MICRS, CADPS, BADGES AND PIN PADS INFORMATION

ws	contr.	Micr	Type	speed	Cadp	Type	speed	Badge	speed	PinP	speed
WS1	TW 1A0	MC1	0	4800	CA1	0	19200	Y	134.5	Y	1200
WS2	KDC1E0	---	-	---	CA1	0	19200	N	-	N	---

TYPE: <+> to continue
<-> to display the previous menu
<q> to quit
ACTION:

where:

ws is the work station identifier

contr. is the symbolic name of the controller to which the work station is connected

Micr indicates the Cheque reader or Cheque reader/printer (MC1, MC2, ...)

Type is the type of Cheque reader (0 = reader, 1 = reader/printer)

speed is the transmission/reception speed in baud

Cadp indicates the Cash Adapter (CA1, CA2, ...)

Type is the type of Cash Adapter (0 = normal, 1 = teller)

speed is the transmission/reception speed in baud

Badge indicates whether the Badge reader is present (Y) or not (N)

speed is the transmission/reception speed in baud

Pinpad indicates whether the PIN Pad is present (Y) or not (N)

speed is the transmission/reception speed in baud

The user can continue in the display phase (entering **+**), display again the previous menu (entering **-**), or terminate the display phase (entering **Q** or **q**).

Olivetti MOS Configuration Manager Rel. 5.2

CHANNEL PARAMETERS

ws	contr.	input	output	stop bit	par	ch	XOFF	XON	tand sett	tand mode	break sign.	sign. status	DTR	RTS
WS1	TW 1A0	128	256	1	EV	7	96	64	1	RT	MARK	0	1	1
WP1	TW 5A0	128	256	1	EV	7	96	64	1	RT	MARK	0	1	1

TYPE: <+> to continue
<-> to display the previous menu
<q> to quit
ACTION:

---

where:

ws — is the work station identifier

contr. — is the symbolic name of the controller to which the work station is connected

input — is the size in bytes of the input buffer (it must be 16 for PIN Pads and 110 for Badge readers)

output — is the size in bytes of the output buffer (it must be 16 for PIN Pads and 110 for Badge readers)

stop bit — is the number of stop bits (0 = 1 bit, 1 = 1.5 bits, 2 = 2 bits; the value 0 is mandatory for PC connection)

par — is the parity check (NP = no parity check requested, OD = odd parity, EV = even parity)

ch — is the number of bits per character (0 = 5 bits, 1 = 6 bits, 2 = 7 bits, 3 = 8 bits; the value 3 is mandatory for PC connection)

XOFF — is the XOFF level (in number of bytes)

XON — is the XON level (in number of bytes)

tand sett — is the tandem mode setting (0 = reset, 1 = XON/XOFF, 2 = BREAK, 3 = CTS)

tand mode	specifies whether the tandem mode is supported in reception (R), in transmission (T) or in both (RT)
break sign.	is the state of the break signal (MARK, SPACE or NULL)
sign status	is the state of a general signal which depends on the peripheral (0 = logical 0, 1 = logical 1, - = no operation)
DTR	is the state of the Data Terminal Ready signal (0 = logical 0, 1 = logical 1, - = no operation)
RTS	is the state of the Request To Send signal (0 = logical 0, 1 = logical 1, - = no operation)

The user can continue in the display phase (entering **+**), display again the previous menu (entering **-**), or terminate the display phase (entering **Q** or **q**).

Option S or s

If the third option (S or s) is selected in the "WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVERS CONFIGURATION" menu, information relevant to the configured system printers and the channels used for their connections is displayed through two menus, as shown in the examples below:

Olivetti MOS Configuration Manager Rel. 5.2

SYSTEM PRINTERS

sysprt	Controller	connection	mode	speed
SP1	TW 2A0	RS232	FREE	4800
SP2	MUX4T1	RS232	FREE	9600

PAGING : <+> Next Page <E> Exit
COMMAND:

---

where:

sysprt — is the system printer identifier

Controller — is the symbolic name of the controller to which the system printer is connected

connection — indicates the type of connection (RS232 = RS232 or CURR = Current Loop)

mode — indicates whether the printer is handled in Olivetti-controlled (CONT) or free running (FREE) mode

speed — is the transmission/reception speed.

Olivetti MOS Configuration Manager Rel. 5.2

SYSTEM PRINTER CHANNEL PARAMETERS

name	contr.	input	output	stop bit	par	ch	XOFF	XON	tand sett	tand mode	break sign.	sign. status	DTR	RTS
SP1	TW 1A0	128	256	1	EV	7	96	64	1	R	MARK	0	1	1
SP2	TW 5A0	128	256	1	EV	7	96	64	1	R	MARK	0	1	1

PAGING : <-> Previous Page <E> Exit
COMMAND:

where:

name — is the system printer identifier

contr. — is the symbolic name of the controller to which the system printer is connected

input — is the size in bytes of the input buffer

output — is the size in bytes of the output buffer

stop bit — is the number of stop bits (0 = 1 bit, 1 = 1.5 bits, 2 = 2 bits)

par — is the parity check (NP = no parity check requested, OD = odd parity, EV = even parity)

ch — is the number of bits per character (0 = 5 bits, 1 = 6 bits, 2 = 7 bits, 3 = 8 bits)

XOFF — is the XOFF level (in number of bytes)

XON — is the XON level (in number of bytes)

tand sett — is the tandem mode setting (0 = reset, 1 = XON/XOFF, 2 = BREAK, 3 = CTS)

tand mode — specifies whether the tandem mode is supported in reception (R), in transmission (T) or in both (RT)

break sign. — is the state of the break signal (MARK, SPACE or NULL) sign status is the state of a general signal which depends on the peripheral (0 = logical 0, 1 = logical 1, - = no operation)

DTR is the state of the Data Terminal Ready signal (0 = logical 0, 1 = logical 1, - = no operation)

RTS is the state of the Request To Send signal (0 = logical 0, 1 = logical 1, - = no operation)

Option G or g

If the fourth option (**G** or **g**) is selected in the "WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVERS CONFIGURATION" menu, information relevant to the configured RS232/CL drivers and their channels is displayed through two menus, as shown in the examples below:

Olivetti MOS Configuration Manager Rel. 5.2

GENERAL PURPOSE DRIVERS

GD	Controller	Speed	DSR	CTS	DCD
GD1	MUX2B0	19200	Y	N	Y
GD2	TW 3B0	4800	Y	N	Y

PAGING : <+> Next Page <E> Exit
COMMAND:

---

where:

GD is the RS232/CL driver instance identifier

Controller is the symbolic name of the controller with which the RS232/CL driver instance is associated

Speed is the transmission/reception speed

DSR is the state of the Data Set Ready signal (Y = significant, N = non-significant)

CTS is the state of the Clear To Send signal (Y = significant, N = non-significant)

DCD is the state of the Data Carrier Detector signal (Y = significant, N = non-significant)

Olivetti MOS Configuration Manager Rel. 5.2

GENERAL DRIVER CHANNEL PARAMETERS

name	contr.	input	output	stop bit	par	ch	XOFF	XON	tand sett	tand mode	break sign.	sign. status	DTR	RTS
GD1	TW 1A0	128	256	1	EV	7	96	64	1	R	MARK	0	1	1
GD2	TW 5A0	128	256	1	EV	7	96	64	1	R	MARK	0	1	1

PAGING : <-> Previous Page <E> Exit
COMMAND:

where:

name — is the RS232/CL driver instance identifier

contr. — is the symbolic name of the controller with which the RS232/CL driver instance is associated

input — is the size in bytes of the input buffer

output — is the size in bytes of the output buffer

stop bit — is the number of stop bits (0 = 1 bit, 1 = 1.5 bits, 2 = 2 bits)

par — is the parity check (0 = no parity check requested, 1 = odd parity, 2 = even parity)

ch — is the number of bits per character (0 = 5 bits, 1 = 6 bits, 2 = 7 bits, 3 = 8 bits)

XOFF — is the XOFF level (in number of bytes)

XON — is the XON level (in number of bytes)

tand sett — is the tandem mode setting (0 = reset, 1 = XON/XOFF, 2 = BREAK, 3 = CTS)

tand mode — specifies whether the tandem mode is supported in reception (R), in transmission (T) or in both (RT)

break sign. — is the state of the break signal (MARK, SPACE or NULL)

sign status	is the state of a general signal which depends on the peripheral (0 = logical 0, 1 = logical 1, - = no operation)
DTR	is the state of the Data Terminal Ready signal (0 = logical 0, 1 = logical 1, - = no operation)
RTS	is the state of the Request To Send signal (0 = logical 0, 1 = logical 1, - = no operation)

Option Q or q

If the fifth option (**Q** or **q**) is selected in the "WORK STATIONS, SYSTEM PRINTERS AND GENERAL DRIVERS CONFIGURATION" menu, the CONFMAN main menu is displayed.

UPDATE COMMANDS

The commands which can be entered by the user (in the menu displayed by CONFMAN when the 'H' or 'h' option is selected in the "DISPLAY/MODIFY WORK STATION, SYSTEM PRINTERS AND RS232 DRIVER CONFIGURATION" menu) in order to modify the current configuration are described in this section.

Available update commands are listed in the bottom part of each menu.

A help menu is displayed if **H** or **h** is entered to the **COMMAND:** prompt. The help menu lists the commands that can be entered.

A different help menu is provided for each CONFMAN menu which allows the existing configuration to be modified, according to whether a work station has already been added to the existing configuration or not.

Entering **/CR/** causes the CONFMAN menu, displayed when help was been requested, to be displayed again, thus permitting the existing configuration to be modified (or confirmed).

All the possible update commands are listed below.

UPDATE COMMANDS	MEANING
+	The current menu (or row) is confirmed, and the next menu (or row) of information is displayed.
-	The previous menu (or row) is displayed.
*	The previous menu is displayed.
&	The current row is to be modified: the cursor is placed at the first field of the row and every value can be either modified or confirmed with /CR/.
Q (or q)	The CONFMAN main menu is displayed.
ADD nn TO mm	The "nn" device (WS, WP, CA, MC, BG, or PP) is added to the "mm" channel (for example, TWIN1A, TWIN2B, KDC3, RSCPU, ...). Details concerning this device will be asked by CONFMAN in the current and in subsequent menus.
DEL nn	The "nn" work station is deleted from the current configuration.
END	Terminates (or skips, if none has been specified) the addition of work station devices. The CONFMAN main menu is displayed.

Tab. 11-3 CONFMAN Update Commands

An error made when entering an ADD command is signalled by an acoustic warning, and the command must be entered again.

Symbolic names which identify devices and channels are as follows:

IDENTIFIER	OBJECT
WS	work station (See the Note 1)
WP	work station printer
BG	badge reader-reader/writer
CA	cash adapter
MC	cheque reader-reader/writer
PP	PIN Pad
SP	system printer
GD	RS232/CL driver instance
TWINnc (n = 1, 2, 3, ...; c = A or B)	channel A or B on the TWIN board 1, 2, 3,...
KDCnc (n = 1, 2, ..., 8; c = A or B)	channel A or B on the KDC 1, 2, 3, ... (See the Note 2)
MUXnlT (n = 1, 2, ..., 8; l = 0, 1, ..., 3)	line number 0, 1, 2 or 3 of the MUX number 1, 2, ..., which is transparent (T) (for work stations only)
MUXnlE (n = 1, 2, ..., 8; l = 0, 1, ..., 3)	line number 0, 1, 2 or 3 of the MUX number 1, 2, ..., which is channel 'E' of an ELB 3683/3684 (for work stations only)
MUXnlc (n = 1, 2, ..., 8; l = 0, 1, ..., 3) c = A, B)	line number 0, 1, 2 or 3 of the MUX number 1, 2, ..., which is the RS232 channel A or B (for devices only)
RSCPU	RS232 line of the CPU board

Tab. 11-4 Symbolic Names for Devices and Channels in CONFMAN

Notes:

1. The work station number must be specified when deleting it using the DEL update command (for example, "DEL WS3").

2. The KDC channel (c = A or B) must not be indicated when referring to a work station. For example, the command "ADD WS TO KDC4" is sufficient in order to add a work station to the KDC number four.

Remarks:

Any update to the existing configuration must be carried out bearing in mind that:

1. neither a badge reader-reader/writer nor a PIN Pad can be associated with a work station added via the "ADD WS TO ..." update command and which has been declared connected to TWIN or RSCPU, as they must be connected to the C or D channel of the MUX or KDC to which the work station is connected

2. no more than one device per type (that is, no more than one WP, BG, ...) can be connected to each work station added via the "ADD WS TO ..." update command

3. no device can be added to a work station if the work station itself has not already been added to the existing configuration (that is, in order to add a device to an existing work station, the work station must first be deleted via the "DEL WSn" update command, then added via the "ADD WS TO ...", and only then can a device be associated with it).

OPTION W or w (WRITE ON DISK THE MODIFICATIONS MADE)

This option writes on disk all modifications made during the current CONFMAN session, in the $CON file specified at CONFMAN activation time, or in the $CON file contained in the SYS volume whose path name was specified at CONFMAN activation time.

OPTION Q or q (CONFMAN TERMINATION)

By selecting this option the user terminates the execution of CONFMAN. Control is given to the application environment from which CONFMAN was activated (for example, Shell).

If, during the CONFMAN session which is being terminated, the **W** or **w** option has not been selected, then the existing configuration remains unchanged.

CONTROLLER BUSY!

A work station device cannot be added to the specified controller (in the Hardware Summary menu), because another device is already connected to it.
Enter /CR/ and choose another controller.

NAME NOT FOUND

The volume specified when activating CONFMAN does not contain a $CON file, or the path name specified does not exist.

PATHNAME IS NOT A VOLUME

The 'L' or 'l' option has been selected in the CONFMAN menu, but the path name specified when activating CONFMAN is not a volume.

```
==  WARNING ==
This work station hasn't any device.
Type <A> to abort (any modification in the configuration file)
     <&> to modify this menu              <-> to modify the previous menu
```

Neither devices nor screen-keyboard have been associated with the current work station. Enter /A/ to delete the WS insertion, or <&> or <-> to modify the current or the previous menu, respectevely.

YOU HAVE JUST ADDED A PP (or BG or WP or MC or CA) FOR THIS WORK STATION!

No more than one device per type can be added to each work station. Enter /CR/.

YOU MUST ADD WS BEFORE!

A work station must be added to the existing configuration before a device can be connected to it.

YOU CANNOT ADD A NEW CA!

32 cash adapters are present in the current configuration and no more are allowed.

YOU CANNOT ADD A NEW GD!

32 instances of the RS232/CL driver are present in the current configuration and no more are allowed.

YOU CANNOT ADD A NEW MC!

32 cheque readers-reader/writers are present in the current configuration and no more are allowed.

YOU CANNOT ADD A NEW SP!

8 system printers are present in the current configuration and no more are allowed.

YOU CANNOT ADD A NEW WP!

32 work station printers are present in the current configuration and no more are allowed.

YOU CANNOT ADD A NEW WS!

32 work stations are present in the current configuration and no more are allowed.

The INSTALLQM command copies all the structures (directories, data files, code files, configuration file, ...) required for either the spooling system or the batch processing system into a system volume, initialising the data structures. If both batch and spooling are required, INSTALLQM must be executed twice to carry out the two installations separately. It does not matter which system is installed first.

Note: In configurations containing a terminal emulator, the spooling system is also necessary.

INSTALLQM dirname1 dirname2 **TYPE** = { **BE** | **SP** } [**"R**]

---

where:

dirname1 is the directory containing the components to be copied

dirname2 is the system volume directory in which they are to be installed

TYPE is a code, either SP for "spooler" (but see "Configuration of the Spooler" below)
or BE for "batch"

"R is the "replace" option to specify that the entire tree structure of directories and files managed by INSTALLQM shall be replaced. The default if "R is not specified is "append" mode, in which the system currently being installed is added to the existing tree structure without overwriting any existing components.

The replace option is necessary when files may remain from a previous installation, for the first use of INSTALLQM. But if a second installation is carried out, that is, both batch and spooling are required, then "R should be omitted so that the first installation is not corrupted.

SPOOLER

Spooler Modules

For the spooler, a directory is created under the MOS directory /IPL with name "SP", that is the TYPE value entered by the user. The following components are copied into this directory:

SPGPA the "Grandpa" (control process) of the spooling system

UNSPOOL the standard print dispatcher program ("unspooler")

A directory $QM is also created under /IPL with two subdirectories: QUEUES and CODES. The queue manager, QUEMAN, is copied into the CODES directory. (This structure is also created for the batch system.)

As each module is copied, a message is printed to confirm the successful copy, that is:

```
SPGPA TRANSFERRED
```

The data files of the spooling system are set up ready for use. In particular, the printer configuration file (which dynamically links physical printers to spooling classes) is created empty, ready for the System Administrator to specify the required links via the SPOOL utility.

A warning is then printed:

```
WARNING: CONFIGURATION FILE (CONF) MUST BE INSTALLED
```

Configuration of the Spooler

After using INSTALLQM, the command SPCONF (described in this chapter) must be used to create the configuration file CONF, which defines the static links between printers and unspoolers. In fact, because of this requirement, the use of SPCONF to install the spooling system and its configuration file, without first using INSTALLQM, is recommended.

For the inclusion of the spooler in the system to be recognised at run time, the source configuration file for the user environment must contain the following two directives (see also Chapter 10):

```
CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
START:/IPL/SP/SPGPA,<>n<>m<>...;
```

An initial string to be passed to QUEMAN has been included as this is a requirement of the syntax, but in fact no parameters are required by QUEMAN. These two directives instruct the system process Grandpa to activate the monitor which will handle the spool queues (QUEMAN) and to start the spooler (see also Chapter 10 in this manual).

BATCH

Batch Modules

For batch processing facilities, a directory is created under the MOS directory /IPL with name "BE", that is the TYPE value entered by the user. The "Grandpa" (control process) for batch processing, BATCHGPA, is copied into this directory.

The queue manager is also installed, in the same way as for spooling (see above).

As each module is copied, a message is printed to confirm the successful copy, that is:

```
QUEMAN TRANSFERRED
```

Configuration of Batch Facilities

To use batch facilities at run time, the source configuration file for the user environment must contain the following two directives:

```
CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
START:/IPL/BE/BATCHGPA;
```

Note: The same monitor (QUEMAN) is used for the management of both the batch queue and the print queues; the directive relating to this monitor need only appear once in the configuration file when both batch and spooling systems are present, as in the following example:

```
CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
START:/IPL/SP/SPGPA!/IPL/BE/BATCHGPA;
```

When the QUEMAN monitor and the **LMS** package are requested, the source configuration file for the user environment must contain the following directive:

```
CALL:QUEMAN=/IPL/$QM/CODES/QUE_LMS,<LMS parameter list>;
```

where /IPL/$QM/CODES/QUE_LMS is the L-module containing both the QUEMAN and the LMS packages.

Note: The QUE_LMS file must be renamed as QUEMAN <u>before</u> using the INSTALLQM command. See the examples below.

Example 1: Installing the Spooling System

In order to install the spooling system, on a machine where neither the batch environment nor the LMS are required, the following steps must be carried out:

1. The directives:

   ```
   CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
   START:SPGPA=/IPL/SP/SPGPA;
   ```

   must be included in the Grandpa configuration file.

2. The SPCONF command (see below in this chapter) must be used to install and configure the Spooling System.

Example 2: Installing the Batch Environment

In order to install the batch environment, on a machine where neither the spooling system nor the LMS are required, the following steps must be carried out:

1. The INSTALLQM command must be used to install and initialise the batch environment, as described above.

2. The directives:

   ```
   CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>par;
   START:BATCHGPA=/IPL/BE/BATCHGPA;
   ```

   must be included in the Grandpa configuration file.

Example 3: Installing the Batch Environment and the LMS

In order to install the batch environment on a machine where the LMS is also required, the following steps must be carried out:

1. A back-up copy of the delivered QUEMAN l-module should be made, by entering:

   ```
   SETWDIR /IPL/DPC/SERVICE

   COPY QUEMAN QUEMAN_BAK
   ```

2. The l-module provided for both the queue manager and the LMS must be named as the INSTALLQM command requires, by entering:

   ```
   COPY QUE_LMS QUEMAN
   ```

3. The INSTALLQM command must be used to install and initialise the batch environment, as described above.

4. The directives:

   ```
   CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<LMS parameter list>;
   START:BATCHGPA=/IPL/BE/BATCHGPA;
   ```

   must be included in the Grandpa configuration file. The LMS parameter list is that described in the section "LMS" in chapter 10.

The KEYCONF command allows to create a new keyboard table or to modify an existing one.

It is a menu-driven interactive utility with option to:

- check current tables
- modify an existing keyboard table
- create a new keyboard table
- modify the screen table.

KEYCONF

This command is to be used, if necessary (that is, if the user needs a particular keyboard or screen table which is different from all those supplied), before running the SYSCONF program.

KEYCONF must be run in the directory containing the standard keyboard tables (that is, the command **SETWDIR /IPL/RELn.l/CONF_GEN/SC/KBTABLES** , where **n** is the release number and **l** is the release level, must be entered before running KEYCONF).

A new user created table must be named as:

KB_USER_1
or
KB_USER_2
or
KB_USER_3

and can subsequently be associated with a work station indicating its name when running SYSCONF.

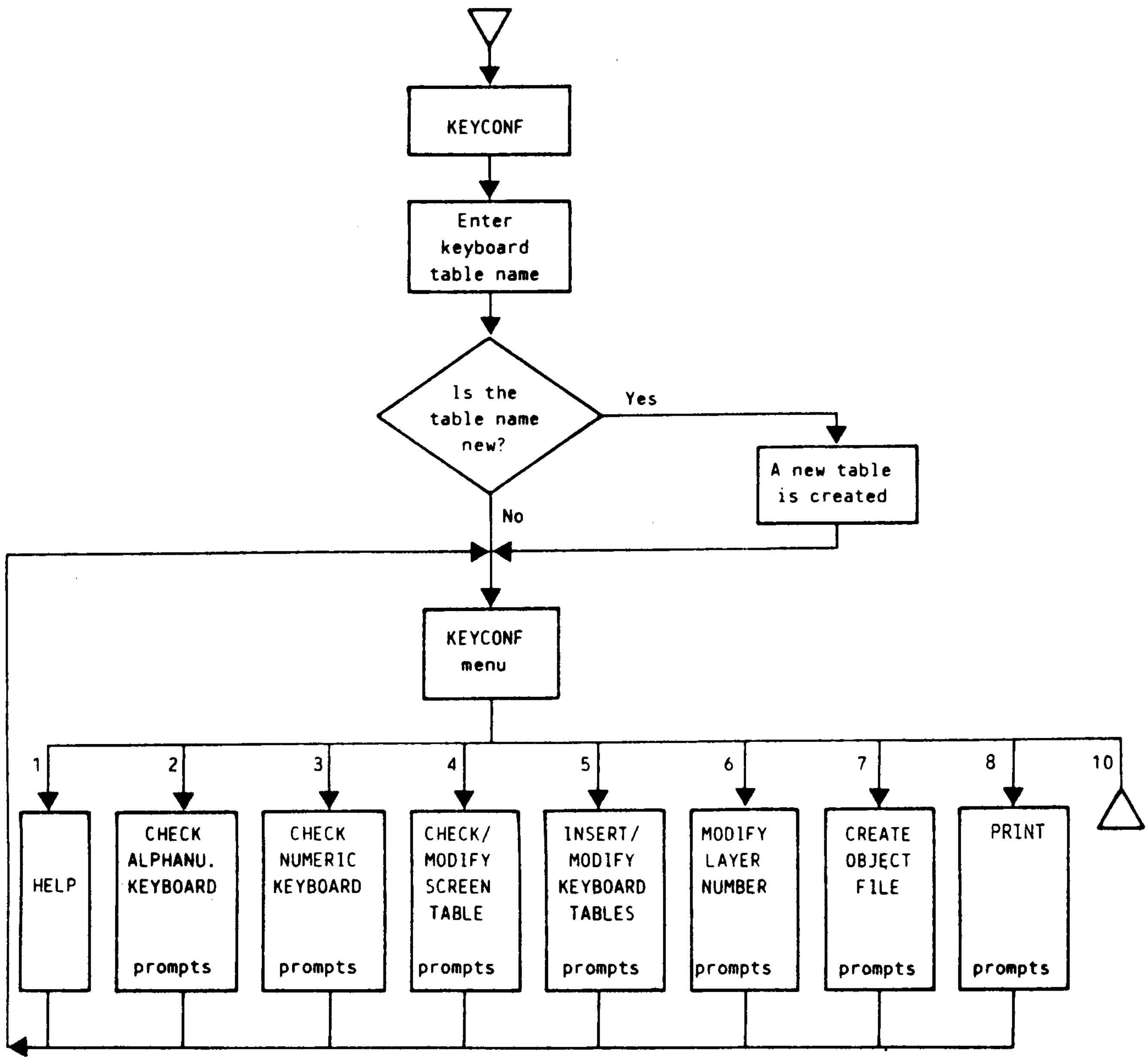


Fig. 11-5 General Logic of the KEYCONF Command

KEYCONF FUNCTIONING

Once activated, the KEYCONF utility prompts the user for a keyboard table name. This is to discern if an existing table is to be modified (the file name entered is that of one of the keyboard tables available in the current directory) or if a new table is to be created (the file name differs from all those contained in the current directory). In the latter case a new table with all its values initialised to 0 is created.

In any case, from now on the utility is working on a table, which will be referred to as "current table" in the following.

Choosen the table, a menu is displayed which offers eight options:

1 Displays a set of help pages, where useful information are given on how to go on with the utility.

2 The table relevant to the alphanumeric part of the keyboard is displayed.

3 The table relevant to the numeric part of the keyboard is displayed.

4 The screen table is displayed and can be modified.

5 A keyboard table can be created or modified.

6 A layer number can be modified.

7 Modification carried out can be made permanent (that is, written on disk).

8 A keyboard table can be printed.

10 Exit.

The general logic of the program is shown opposite. Choosing an option (except Exit) gives rise to a single display or sequence of displays, which either prompt for information or present stored information. The option **7** must always be selected, after having created or modified a table, in order to write on disk the resulting table.

Simple error messages appear if responses to prompts have invalid syntax or are incorrect, and the user must correct his input before he can continue. Each response accepted causes the next prompt to appear. The different displays are described below.

GENERAL CONCEPTS

Some concepts and terms are here briefly illustrated, as they often appear in the following and may be not familiar enough to the reader.

A table stored on disk is associated with each keyboard which permit to identify each key stroke with a code. Different types of tables exist to provide national sets of characters. Within a table, different sets of codes are grouped into layers according to the condition in which each key is pressed (that is, normal, with SHIFT, with LOCK, with CONTROL, with SHIFT and LOCK, and so on).

Another table is associated to the screen, which is used to determine which is the ASCII character to be displayed for each code (generated by the keyboard).

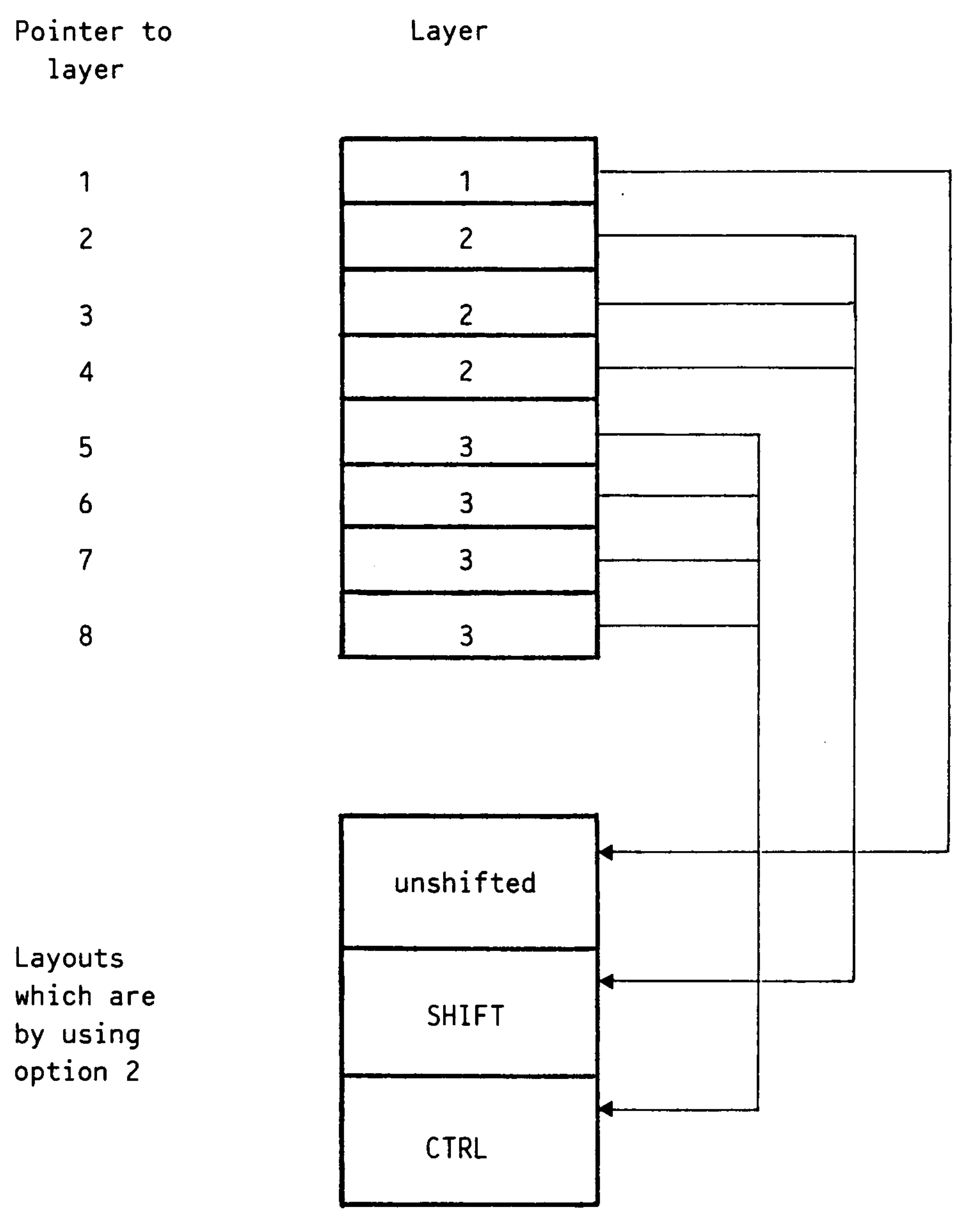


Fig. 11-6 Relationship between Pointers to Layers, Layers and Layouts

OPTION 2 (CHECK ALPHANUMERIC KEYBOARD) PROMPTS

```
CHECK ALPHA-KEYBOARD >
=====================

LAYERS OF THE VERSION:
x...xx.x
====================
n mmm kkkk ...

INPUT POINTER TO LAYER
```

n mmm kkkk ... are the layers of the current table (named x...xx.x). The highest number in the displayed list shows how many layers has the current table. An example of layers and relevant meanings follows:

```
                         1 222 3333
                         |  |||  ||||
                normal---┘  |||  ||||
                   SHIFT----┘||  ||||
                     LOCK----┘|  ||||
            LOCK + SHIFT------┘  ||||
                        CTRL-----┘|||
               CTRL + SHIFT-------┘||
                 CTRL + LOCK-------┘|
       CTRL + LOCK + SHIFT----------┘
```

nn The user must enter an integer number which specifies the pointer to the layer whose one or more meanings are to be modified. The pointer number is that of its position in the displayed list: for instance, in the example above the number to be entered to specify the pointer to the CTRL layer is 5 (as CTRL is the fifth item in the list).

Chosen the layer, a table is displayed, which shows, for each alphanumeric key, the positional code (top) and either the ASCII character code or the command identifier (for SHIFT, CTRL, ... keys).

OPTION 3 (CHECK NUMERIC KEYBOARD) PROMPTS

```
CHECK NUMERIC-KEYBOARD
======================

LAYERS OF THE VERSION:
x...xx.x

======================

n mmm kkkk ...

INPUT POINTER TO LAYER
```

nn Refer to details given before for option 2 in order to choose the layer.

The table displayed refers to the numeric part of the keyboard.

OPTION 4 (CHECK/MODIFY SCREEN TABLE) PROMPTS

When this option is chosen, the table containing the internal codes used for generating character to be displayed is shown.

DO YOU WANT TO MODIFY? (Y/N)

	Y	An affirmative response gives rise to the next prompt.
	N	The KEYCONF menu is displayed.

<CODE RANGE (00:FF) ? (*=BYE)

	xx	The ASCII code whose meaning is to be modified (that is, whose output on screen will be different).

<CHARACTER CODE ? (Actual, xx)

	yy	The address of the ROM which generates characters that will be used to generate the character corresponding to the CODE RANGE just entered. Each national keyboard has a different ROM.

The last couple of prompts is repeated until an asterisk is entered instead of a CODE RANGE.

OPTION 5 (INSERT/MODIFY KEYBOARD TABLES ...) PROMPTS

INSERT/MODIFY KEYBOARD TABLES
=============================

POINTER TO LAYER-NUMBER? (0 = EXIT)

nn Integer specifying the pointer to the layer whose one or more meanings are to be modified. This integer can be retrieved using the option 2.

<POSITIONAL CODE ? (* = BYE)

mm Positional code which identifies, in the layout displayed with option 2, the key whose meaning is to be changed.

<CHARACTER CODE ? (Actual, XXXX)

YYYY The ASCII code (or command identifier) which will be generated when the key identified by the given positional code will be pressed.

The last couple of prompts is repeated until an asterisk is entered instead of a POSITIONAL CODE.

OPTION 6 (MODIFY LAYER NUMBER) PROMPTS

```
    IDENTIFIER = xxxx
NEW IDENTIFIER =
```

xxxx is the hexadecimal code by which the current table is specified when configuring the Operating System (see chapter 3, UNIT 5).

yyyy Hexadecimal identifier which will identify the table.

The displayed value (xxxx) is assumed as default if only /CR/ is pressed.

```
NUMBER POINTERS TO LAYERS = n
ENTER NEW NUMBERS OF POINTERS
```

n is the total number of pointers to layers in the current table.

m Total number of pointers to layers in the table being created.

The list of pointers to layers is displayed, which consists of the first m pointers if m < n, and of the n previous pointers if m is greater than or equal to n.

```
INSERT NEW SEQUENCE LAYER:
```

1/CR/	The required new list of layers.
2/CR/	The same integer entered more than once
2/CR/	means creating more than one pointer to the
2/CR/	same layout (see "General Concepts" above).
3/CR/	
.	
.	
.	

OPTION 7 (CREATE FILE-OBJECT) PROMPTS

FILE OBJECT GENERATOR
=====================
KEYBOARD-NAME = xxxx.x

NEW FILE KEYBOARD

x..xx.x	The file name into which the newly created (nevertheless if it is new or just modified) table is to be saved. The current table file name entered after the first KEYCONF prompt is assumed as default if only /CR/ is pressed.

OPTION 8 (PRINT) PROMPTS

PRINT TABLE
=====================
KEYBOARD-NAME = xxxx.x

The current table file is printed on the system printer.

ERROR MESSAGES

>>>>>>> WRONG LEVEL - REPEAT <<<<<<<

An invalid pointer to layer has been entered. Enter another.

>>>>>>> WRONG POSITIONAL CODE - REPEAT <<<<<<<

A non existent positional code has been entered. Enter another.

This command links two or more physically separate hard disk devices into a single logical unit. Up to four hard disks may be linked. It can only be executed by the system administrator (ROOT user).

When linked, the hard disks appear to the system as a single logical unit with a single global name. The individual devices must exist in the device directory of the system (/DEV).

It is recommended that devices be linked, if required, before using the first of the disks which are to be linked. This is to avoid problems which could arise from file system structures allocated in non contiguous space.

The user is reminded that the MODCON Shell command must be used, after LINKDEV, in order to modify the $CON configuration file accordingly to the LINKDEV command entered (see the LOGHD parameter in Chapter 4).

LINKDEV sysname dev1 devn [devn] ...

---

where:

sysname is the name of the mounted system volume in which the linked devices are to be recognised. This name must be relative to local root (no initial slash character).

dev1 is the name of the hard disk unit which is to be recognised as the single logical unit.

devn are the names of the devices which are to be linked under the logical name dev1.

Example

LINKDEV IPL HD1 HD2 HD3

This links the hard disk devices HD1, HD2 and HD3 into a logical unit recognised as HD1 by the system resident in /IPL.

Note: Hard disks of the "WREN" type (connected to the ST506 controller) cannot be linked.

ERROR MESSAGES

DEVICE NUMBER TOO LARGE

An attempt has been made to link more than four devices.

ERROR IN WK_BIT MAP CREATION: NO ROOM AVAILABLE

The free space on the first disk is not enough to create a working bit map of the other specified disks.

INVALID PARAMETER(S): CAN LINK ONLY HARD DISK

It is not possible to link devices other than hard disks.

LINK DEVICE NOT ALLOWED

Only one device was specified; it is not possible to link a device to itself.

The MKBOOT command copies a switching procedure (of length 1.5 kb) and the system bootstrapper (of length 6 kb) from a source directory to a disk which will be used as a MOS system disk.

The appropriate procedure is selected automatically.

There are two bootstrappers: $BOOT_4 and $BOOT_6, for M3X/M4X and M5X/M6X/M7X respectively. When the MKBOOT command is activated, the user is prompted to enter the type of system in use (either M40 or M60).

MKBOOT sourcedir destdev

---

where:

sourcedir is the complete path name of the directory containing the files $BOOT_4, $BOOT_6 and $COMMFD (a switching procedure)

destdev is the name of the disk drive which contains the disk to be prepared (not mounted).

Example:

MKBOOT /IPL/SYS FL2
ENTER MACHINE TYPE (M40/M60): **M40**

Note: The files to be copied must be in the same directory, and the disk must have been prepared using MKVOL (hard disk) or MKENV followed by MKVOL (floppy disk). See the SHELL Commands, Reference Manual for information on these two commands.

The switching procedure (used to select either the diagnostic or the MOS environment and which is automatically stored on disk when this is formatted) is placed at the beginning of track 0 of the destination disk, without overwriting any sectors that are already reserved. On floppy disk it occupies sectors 1 to 4, 6, and 8 to 10 (sector length 128 bytes).

The precise address at which to locate the bootstrapper for a particular system is recorded in a descriptor (compiled by MKVOL) on the disk known to MKBOOT.

The MKLOGIN command creates the user login data base for a new system. After the execution of the command, and restart of the system, the login mechanism is effective for all users who connect to the system. For a system with MOS directory name /IPL, the command does the following:

- Creates the 'ETC' directory under the MOS directory (/IPL).
- Creates the files 'PASSWD', 'GROUP' and 'LASTLOG' in /IPL/ETC to contain permanent information about users.
- Creates the 'USR' directory under the MOS directory /IPL to contain user directories.
- Creates the first logical group 'SYSTEM' and the first user 'ROOT'.

The figure overleaf shows the directory structure of the system disk before and after MKLOGIN.

MKLOGIN dirname

---

where:

dirname identifies the directory where the login mechanism is to be created (a MOS directory).

Example: MKLOGIN /IPL

Note: If dirname is simply "/", that is, the memory volume, the login structure is written into main store and not onto disk. Therefore at logout time, the information will be lost.

Login Files

The 'PASSWD' and 'GROUP' files are updated by the USERMAN utility (see <u>SHELL Commands Reference Manual</u>). The 'LASTLOG' file is for system use only. The contents of the first two files are briefly described below.

PASSWD: The 'PASSWD' file contains the following information for each user: user name, password, user number, group number, information string, home directory, Shell program active at creation time. The last field is not currently used.

GROUP: The 'GROUP' file contains the following information for each user group: group name, group number, list of users in this group. All users must belong to a group. Users in the system group (group name "SYSTEM") have privileged status, and only these users can start up or shut down the system.

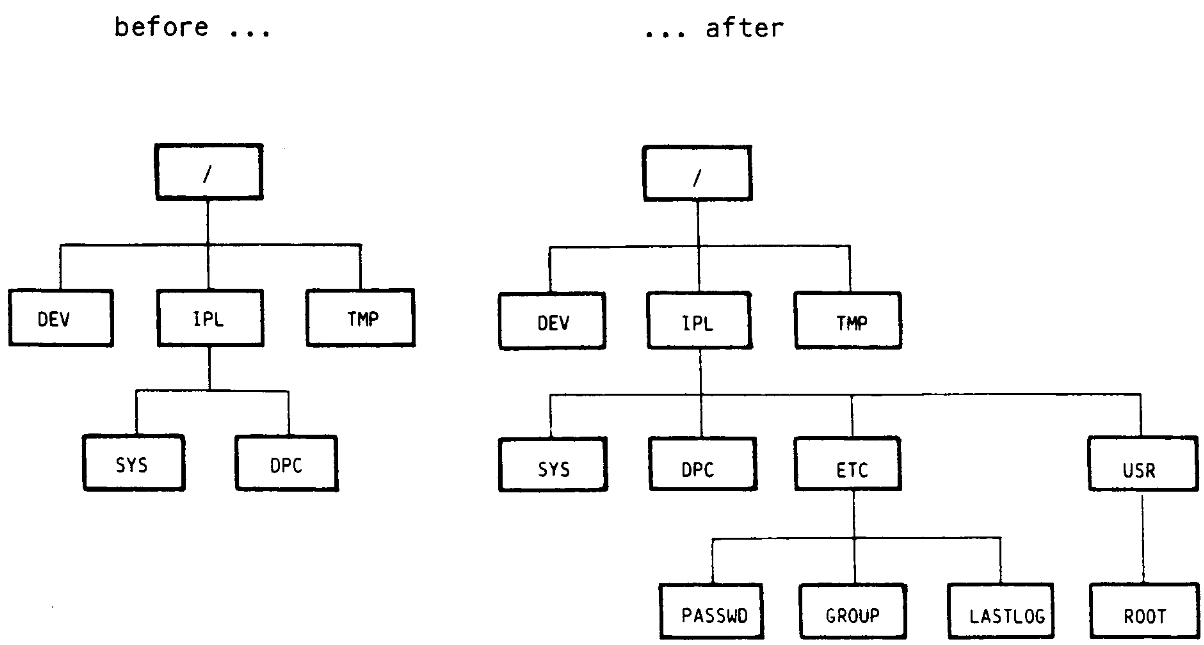


Fig. 11-7 The Effect of MKLOGIN

Note: For systems with only SCT as back up media, it is recommended that the USR directory is changed into a volume in order to facilitate back up operations. This is achieved, for example, by entering the Shell commands listed below:

```
RENAME /IPL/USR /IPL/USR1
MKVOL /IPL/USR
RENAME /IPL/USR1 /IPL/USR
```

The MKSYS command allows you to prepare a streaming cartridge tape or magnetic tape, then write onto it the contents of a configured customer system. The tape thus contains a system dump for use at initialisation time. The figure below shows the format of a tape after MKSYS has been used.

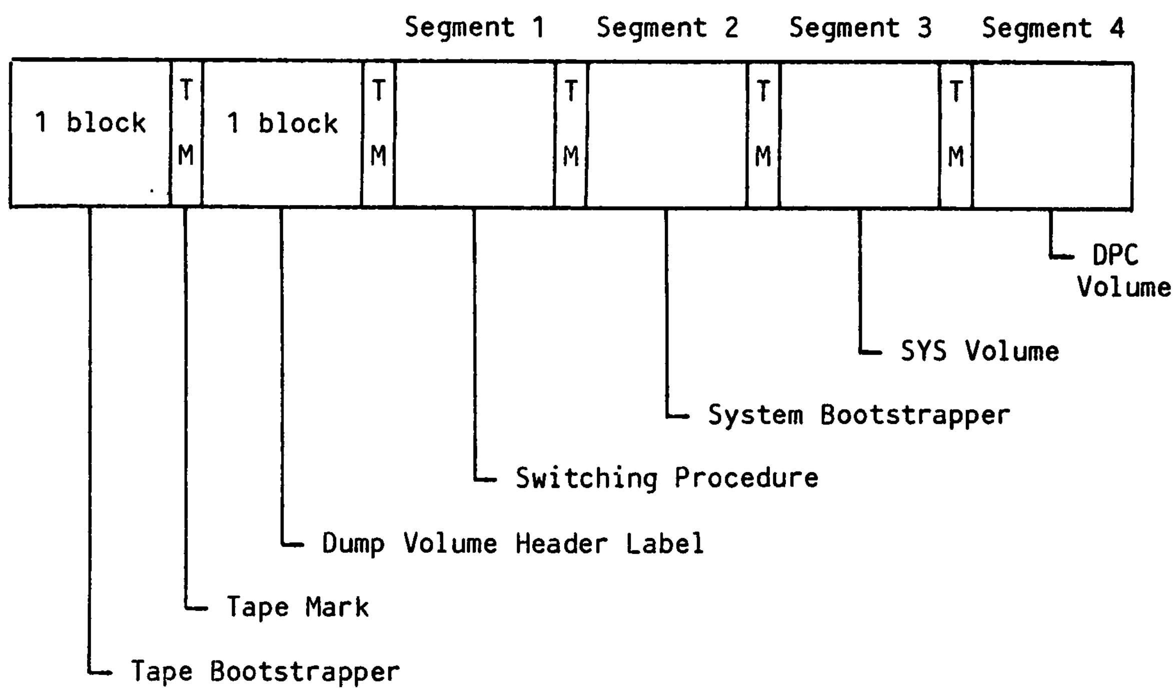


Fig. 11-8 Tape Format

The use of a tape prepared by MKSYS on a non-empty hard disk does not necessarily reinitialise the disk onto which it is loaded; you can specify at startup time that only the system volume SYS shall be overwritten, so that any remaining contents of the hard disk are still accessible.

$$\textbf{MKSYS} \begin{Bmatrix} \textbf{SC4} \\ \textbf{SC5} \\ \textbf{MT1} \\ \textbf{MT2} \end{Bmatrix} [\ \textbf{VERIFY}\ ]$$

where:

SC4 indicates that a tape is being prepared for a customer machine equipped with a SCT4 streaming cartridge tape.

SC5 indicates that a tape is being prepared for a customer machine equipped with a SCT5 streaming cartridge tape.

MT1, MT2 indicates that a tape is being prepared for a customer machine equipped with a MTU magnetic tape (unit 1 and unit 2).

VERIFY is an option to display the dump volume header label, skipping the requests for parameters and using instead the existing values if confirmed by the user. The display includes the lengths of the component in each tape segment.

FIELD NAME	DEFAULT VALUE	EXPLANATION
Label identifier	VOL	The name of the label
Label number	1	The number of the label
Volume identifier	spaces	The name of the volume
Accessibility indicator	space	Indicates restrictions on access to the volume (space = free access)
Owner identifier	spaces	The owner of the volume
Label type	D	The type of volume ("D" is for dump volume)
Dump identifier	N xxyy	The identifying code of the volume in the format "N xxyy", where "N" means native, "xx" is the release number and "yy" the release level
Multivolume indicator	space	Indicates whether the volume occupies more than one cartridge (space = 1)
Creation date	spaces	The date the system cartridge tape was created, in yymmdd format
Length of data	space	The number of blocks (each 256 bytes) occupied by the volume if volume is in single segment
Segment name	space	The name of the tape segment
Segment length	space	The length of the tape segment in blocks

(this last pair repeated for each segment)

Tab. 11-9 Dump Volume Header Label Fields

Following the bootstrapper and the tape label (each one block in length), the body of the tape is divided into intervals which may be of different lengths, known as segments. Each segment of the tape is separated from the next by a "tape mark" (TM). The first two segments contain routines used to bootstrap the system; the third and fourth segments contain the system to be installed. Each item on the tape is described briefly below.

tape bootstrapper: Allows you to load the remainder of the tape (minus the header label) onto hard disk at installation time. Module name $SCTBOOT (for SCT4 devices), $SCT5BOOT (for SCT5 devices) or $MTUBOOT (for MTU devices).

dump volume header label: Consists of a set of fields and values describing the tape's contents. These fields are explained in the table opposite.

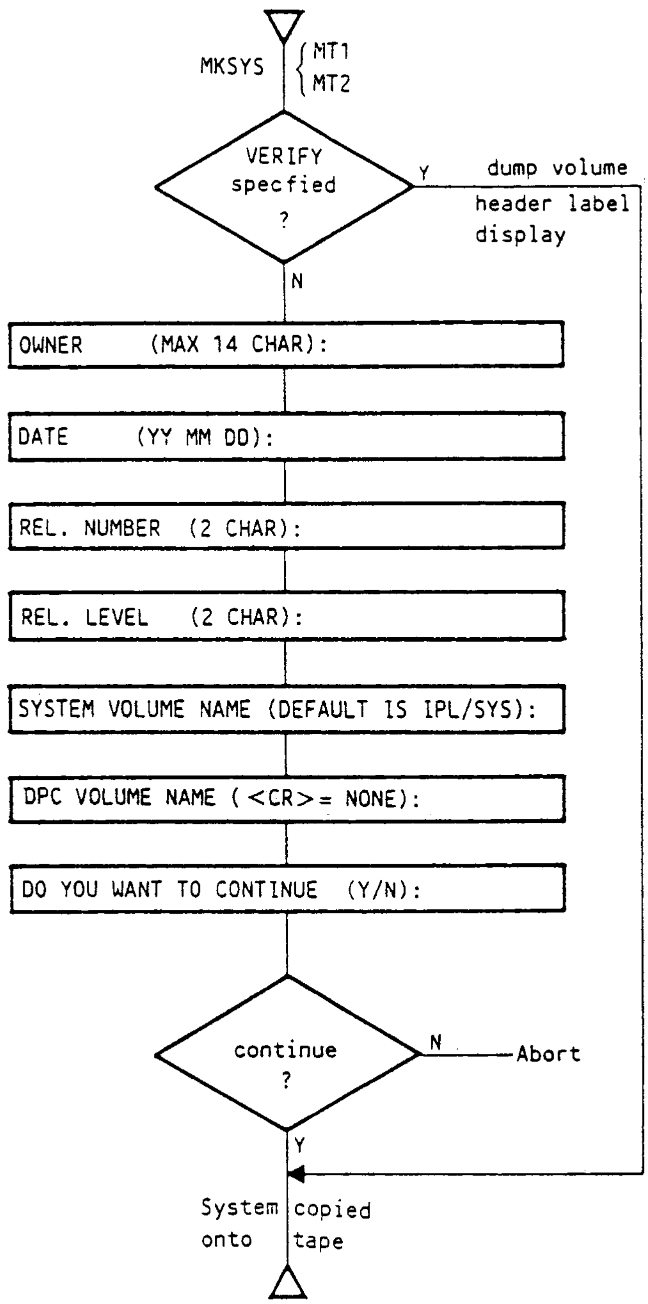
switching procedure: Loads the system bootstrapper code into the RAM at system start-up time.

system bootstrapper: Bootstraps the system from disk at system start-up time.

system volume dump: Contains l-modules of SYS volume of a configured system, with volume descriptor.

dependent components volume: Contains l-modules of DPC volume of a configured system, with volume descriptor.

Flow of Control for Magnetic Tape



MESSAGE DESCRIPTION

OWNER (MAX 14 CHAR):

Prompts for an owner name.

xx...x /CR/ Type a string of up to 14 characters.

DATE (YYMMDD):

Prompts for tape creation date.

yymmdd /CR/ Type six characters in year/month/day format.

REL. NUMBER (2 CHAR):

Prompts for the software release number.

xx /CR/ Type two characters.

REL. LEVEL (2 CHAR):

Prompts for the software release level.

xx /CR/ Type two characters.

SYSTEM VOLUME NAME (DEFAULT IS IPL/SYS):

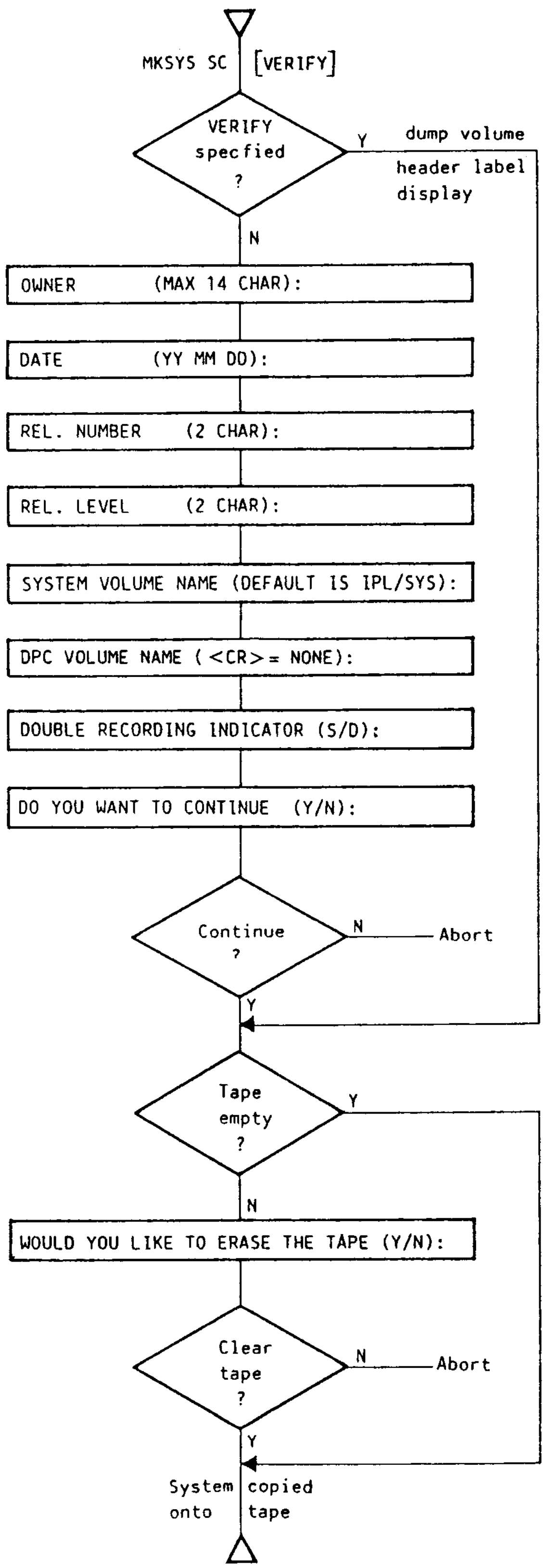
Prompts for the system volume that you wish to copy.

xx...x /CR/ Type the path name of the volume.

/CR/ Accepts the default of "IPL/SYS".

Note: The volume will be copied under the name "SYS" regardless of the name given in input.

Flow of Control for SCT



DPC VOLUME NAME (<CR> = NONE):

Prompts for the dependent components volume that you wish to copy, if any.

xx...x /CR/ Type the name of the volume.

/CR/ Specifies that a DPC volume is not required.

Note: When a DPC volume is specified, it is copied under the name "DPC" regardless of the name given in input.

DOUBLE RECORDING INDICATOR (S/D):

Prompts for single ("S") or double ("D") recording mode. Double recording provides a duplicate for each track, to minimise the possibility of subsequent hardware errors when reading from the tape.

S /CR/ To single-record.

D /CR/ To double-record.

This message is not displayed if preparing a SCT5 tape.

DO YOU WANT TO CONTINUE (Y/N):

Prompts you to confirm your parameter entries.

Y /CR/ To continue.

N /CR/ To abort the program and return to the Shell environment.

WOULD YOU LIKE TO ERASE THE TAPE (Y/N):

Prompts you to clear the previous contents of the streaming cartridge tape before writing onto it.

Y /CR/ Clears the tape and proceeds.

N /CR/ Aborts the program and returns to the Shell environment.

---

xxxx   IS NOT A VOLUME

The name specified as either the system volume or the dependent components volume is present but is not a volume.

INCOMPATIBLE HARDWARE. EXECUTION TERMINATED

An attempt to write in Olivetti mode (used for SCT4) on a SCT5 media. MKSYS must be run again, specifying the right tape.

PUT THE PERIPHERAL READY AND PRESS (C=CONTINUE/R=RETURN):

The device is not correctly set, or the tape has not been inserted.

C /CR/ Check the positions of all switches on the device, then check the tape insertion and continue ("C") if all now seems correct.

R /CR/ Abort the program and return ("R") to the Shell environment.

TAPE HARDWARE ERROR

A fault occurs in reading from the tape. The tape should be removed which causes the above prompt to appear, permitting recovery with the use of a different tape.

TAPE BOOTSTRAPPER NOT FOUND

The module $SCTBOOT or $MTUBOOT is not present in the system volume to be dumped. It will not be possible to transfer the system volume to hard disk.

HDU BOOTSTRAPPER NOT FOUND

No bootstrapper on the disk. It will not be possible to bootstrap the system.

HDU BOOTSTRAPPER READ ERROR

Disk hardware error while reading the bootstrapper.

ERROR IN WRITE OF TAPE or
TAPE BOOTSTRAPPER TRANSFER ERROR or
DUMP LABEL TRANSFER ERROR or
TRACK 0 TRANSFER ERROR or
OS TRANSFER ERROR

Disk hardware error. Non-recoverable faults at different stages of data transfer from disk to tape.

Note: All disk hardware errors are non-recoverable faults.

The MODCON command operates on the $CON file (the configuration file created using the SYSCONF command) and allows the user to:

- display the list of 1-modules and relevant UNITs
- display and modify the parameters of a UNIT
- display and modify the parameters of all the UNITs
- print the parameters of a UNIT
- print the parameters of all the UNIT
- display the hexadecimal dump of the $CON file
- print the hexadecimal dump of the $CON file
- display the hexadecimal dump of a section of the $CON concerning an 1-module
- print the hexadecimal dump of a section of the $CON concerning an 1-module.

MODCON filename

---

where:

filename is the name of the configuration file $CON and can be either its local name or its complete path name (that is, whose first character is "/").

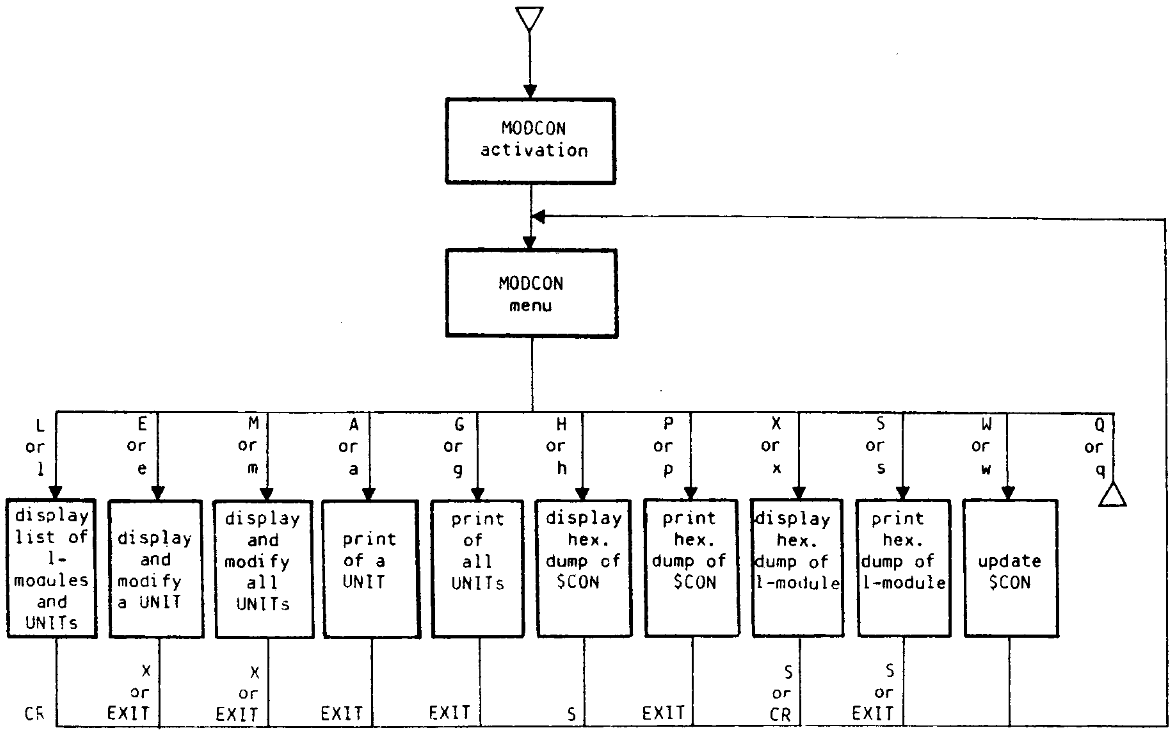


Fig. 11-10 General Logic of the MODCON Command

MODCON FUNCTIONING

Once activated, MODCON displays a menu which offers eleven options:

L or l	Display the list of 1-modules and relevant UNITs.
E or e	Display and modify the parameters of a UNIT.
M or m	Display and modify the parameters of all the UNITs.
A or a	Print the parameters of a UNIT.
G or g	Print the parameters of all the UNIT.
H or h	Display the hexadecimal dump of the $CON file.
P or p	Print the hexadecimal dump of the $CON file.
X or x	Display the hexadecimal dump of a section of the $CON concerning an 1-module.
S or s	Print the hexadecimal dump of a section of the $CON concerning an 1-module.
W or w	Write on disk the modifications made.
Q or q	Exit.

The general logic of the MODCON command is shown opposite. Choosing an option (except Exit) gives rise to a single display or sequence of displays which present the requested information and, in some cases, allows it to be modified.

The option **W** (or **w**) must always be selected, after having made any modification, in order to write them on disk.

The different options are described below.

OPTION L or 1 (DISPLAY L-MODULES LIST AND UNITS)

This option displays the list of l-modules contained in the $CON file, each followed by the relevant UNIT.

Depressing the carriage return (/CR/) exits this option, and the main menu is presented.

An example of list of l-modules and UNITs is the following:

```
FILE : /IPL/SYS/$CON
                                LIST OF UNITS
L_MODULE : $INFO
 UNIT :   0
L_MODULE : $IPL
 UNIT :   1
 UNIT :  33
 UNIT :   5
 UNIT :  14
 UNIT :  20
 UNIT :  18
 UNIT :  24
L_MODULE : $PMM
 UNIT :  10
L_MODULE : $BST
 UNIT :  50
L_MODULE : $AUX
 UNIT :  51
L_MODULE : $POS
 UNIT :  52
L_MODULE : $KEY
 UNIT :  53
                                CONTINUE WITH CR :
```

OPTION E or e (DISPLAY AND MODIFY THE PARAMETERS OF A UNIT)

This option displays the parameters of a UNIT. The user must specify the UNIT number answering to the prompt emitted when the 'E' option is selected. The user may, if necessary, modify the current values for those parameters which are modifiable.

All the parameters are displayed in screen pages (max. 60 per page). Each page has an header which gives the following information:

- Name of the configuration file
- Number of the selected UNIT
- Offset of the selected UNIT
- Length of the selected UNIT
- Number of parameters in the selected UNIT
- Number of pages (expressed in the format:

  Number of current page/Total number of pages)

All the parameters are displayed in the format:

parameter name = value

and each page consists of up to three columns of parameters (no more than 20 parameters per column).

Those parameters which are not modifiable appear without names, and their values cannot be altered. Each value may be an integer, alphanumeric or hexadecimal (whose first character is '%') value.

The user is allowed to modify the values of the parameters whose name is displayed. In order to do this (that is, to move the cursor, to modify values and to exit from this option) the user may use the keys whose meaning is as follows:

KEY	MEANING
→	Moves cursor one position right
←	Moves cursor one position left
^	Moves cursor one position up
v	Moves cursor one position down
→|	Moves cursor to the right most column
|←	Moves cursor to the left most column
+LINE	Moves cursor one line down
-LINE	Moves cursor one line up
CR	Moves cursor to the next line
+PAGE	Displays the following parameters (60 at the most)
-PAGE	Displays the preceding 60 parameters
HOME	Moves the cursor to the upper left corner of the current page
CLEAR	Deletes the current value of the parameter
BS	Deletes the latest character entered
IC	Allows a new character insertion
DC	Deletes a character
EXIT	Exits from the 'E' option of MODCON
X	Exits from the 'E' option of MODCON

Tab. 11-11 Meaning of Some Keys in the 'E' Option of MODCON

Modifications are not written directly to the $CON file, but to a copy residing in memory (created when MODCON 'E' option is selected). Modifications are made permanent when (after exiting from the 'E' option) the user selects the 'W' MODCON option from the main menu (see below).

OPTION M or m (DISPLAY AND MODIFY THE PARAMETERS OF ALL THE UNITS)

This option displays the parameters of all the UNITs present in the $CON file.

The user may, if necessary, modify the current values for those parameters which are modifiable. The way to do it is that described above for the 'E' option. The meaning of the various keys is the same, too, except that 'X', instead of terminating the session, displays the next UNIT's parameters. When the last UNIT has been displayed, the /X/ key terminates the session and exits from the 'M' option.

The 'W' option must be selected if any modification made is to be written to the actual $CON file on disk.

OPTION A or a (PRINT THE PARAMETERS OF A UNIT)

Prints on the specified device all the parameters of a UNIT.

The UNIT number and the device to be used must be entered by the user in reply to prompts.

If the UNIT whose number has been specified is not included in the $CON file, then the message "UNIT NOT AVAILABLE!" is displayed, and the user may enter a new number (for another UNIT) or depress the /EXIT/ key (to exit the 'A' MODCON option).

If the required UNIT is present in the $CON file, MODCON displays the available print devices, asking which is to be used. If the specified device is not ready, then a message is emitted ("ERROR BY CONNECT OF ...", "ERROR BY OPENSEQ OF ..." or "ERROR BY OPENPR OF ..."), and the user may enter a new device identifier (to retry the print phase) or depress the /EXIT/ key (to exit the 'A' MODCON option).

The printout presents all the parameters of the UNIT in the format:

parameter name = value

OPTION G or g (PRINT THE PARAMETERS OF ALL THE UNITS)

Print on the specified device all the parameters of all UNITs contained in the $CON file.

The operation mode is exactly the same as that described above for the 'A' option, except that no UNIT number is specified: only the printing device is selected.

In the printout each UNIT is preceded by a heading line which identifies the UNIT itself.

OPTION H or h (DISPLAY THE HEXADECIMAL DUMP OF THE $CON FILE)

This option displays the hexadecimal dump of the $CON file.

The $CON file is a byte stream file, and it is seen by the system as structured in logical records ("configuration records"). Each l-module is associated with one or more configuration records.

Each configuration record consists of a first field (2 bytes) which contains the record length, followed by two or more fields (always preceded by their length). Information contained in these fields is:

- l-module name
- status word (defined if the l-module is executed in privileged mode or not)
- system identification code
- initialisation values.

The hexadecimal dump is displayed as a series of pages. The display phase may be interrupted by entering the /S/ key, which exits the 'H' option.

The display phase terminates when the value -1 (FF in hexadecimal) is detected instead of a meaningful length for a configuration record.

The format in which the hexadecimal dump is displayed is as follows:

```
FILE : /IPL/SYS/$CON
                           COMPLETE HEXDUMP OF $CON
0000:  00 84 00 2A 24 49 4E 46 4F 20 20 20 20 20 20 20   "...*$INFO          "
0010:  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   "                   "
0020:  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   "                   "
0030:  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20   "                   "

  ...    ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
```

---

where the left most column indicates the number of the displayed line (0, 1, 2, 3, ...), the central column is the hexadecimal dump requested and the right most column is its ASCII representation.

OPTION P or p (PRINT THE HEXADECIMAL DUMP OF THE $CON FILE)

Prints the hexadecimal dump of the $CON file.

The operation mode is exactly the same as that described above for the 'G' option.

OPTION X or x (DISPLAY THE HEXADECIMAL DUMP OF A SECTION OF THE $CON FILE CONCERNING AN L-MODULE)

Displays the hexadecimal dump of a section of the $CON concerning an l-module.

The user must specify the l-module name answering to the prompt shown when the 'X' option is selected.

If the l-module whose name has been specified is not present in the $CON file, then the message "L_MODULE NOT FOUND!" is displayed, and the user may enter a new name (for another l-module) or depress the /S/ key (to exit the 'X' MODCON option).

If the specified l-module is present in the $CON file, the hexadecimal dump of the section of $CON concerning it is displayed: the user may list the various screen pages depressing /CR/ or exit from the 'X' option depressing the /S/ key. Both the /S/ and the /CR/ keys have the same effect (exit) if the hexadecimal dump is not longer than one page.

OPTION S or s (PRINT THE HEXADECIMAL DUMP OF A SECTION OF THE $CON FILE CONCERNING AN L-MODULE)

Prints the hexadecimal dump of a section of the $CON concerning an l-module.

The user must specify the l-module name answering to the prompt shown when the 'S' option is selected.

If the l-module whose name has been specified is not included in the $CON file, then the message "L_MODULE NOT FOUND!" is displayed, and the user may enter a new name (for another l-module) or depress the /S/ key (to exit the 'S' MODCON option).

If the specified l-module is present in the $CON file, MODCON displays the available print devices, asking which is to be used. If the specified device is not ready, then a message is emitted ("ERROR BY CONNECT OF ...", "ERROR BY OPENSEQ OF ..." or "ERROR BY OPENPR OF ..."), and the user may enter a new device identifier (to try again the print phase) or depress the /EXIT/ key (for exiting the 'S' MODCON option).

If the device is ready, the hexadecimal dump of the section of $CON concerning the selected l-module is printed.

OPTION W or w (WRITE ON DISK THE MODIFICATIONS MADE)

This option writes to $CON (on disk) all modifications made to the $CON copy held in memory (during the current MODCON activity, in 'E' or 'M' sessions).

OPTION Q or q (MODCON TERMINATION)

By selecting this option the user terminates the MODCON execution. Control is given to the application environment from which MODCON was activated (for example, Shell).

ERROR MESSAGES

INPUT : MODCON FILENAME	No file name has been specified when activating MODCON.
INVALID OPERATION	The file name specified when activating MODCON is not a $CON configuration file.
NAME NOT FOUND	The file name specified when activating MODCON does not exist.

The OLICONF command configures the **OLIEMU** emulator.

It is a menu-driven interactive utility with options to:

- define the type of connection between the PC and the L1 MOS system, and the relevant line parameters
- define the work station printer's parameters, if any
- define the other work station devices' parameters, if any, such as badge reader (or reader/writer) or PIN Pad
- select the version and type of keyboard table
- define further parameters, such as type of PC, type of emulated screen, etc.

OLICONF

OLICONF must be run on the PC to be used as an L1 work station, and it must reside in the MS-DOS directory containing the L1WSE emulator program.

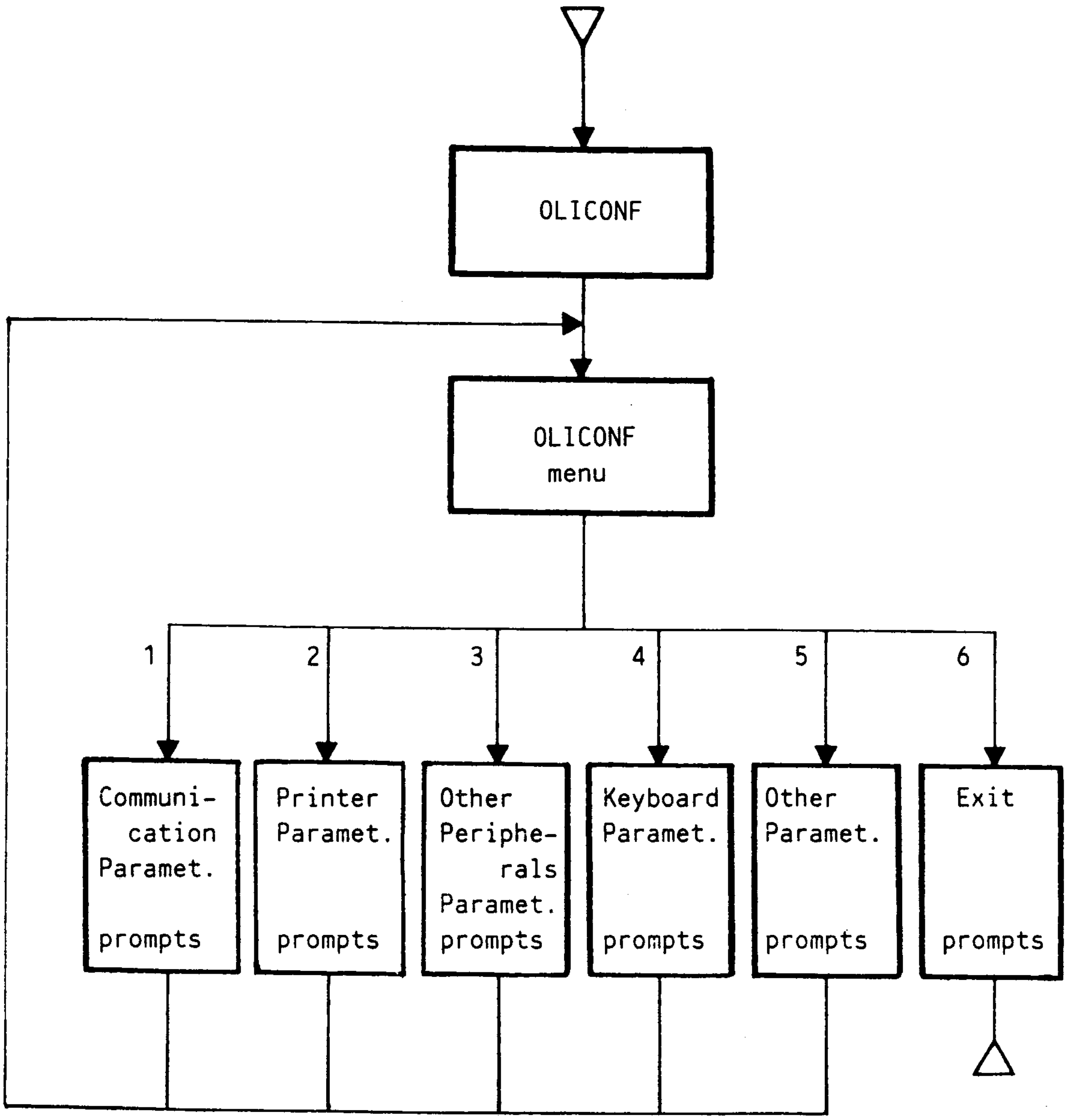


Fig. 11-12 General Logic of the OLICONF Command

OLICONF FUNCTIONING

The OLICONF menu offers six options:

1. to define the type of connection between the PC and the L1 MOS system and the relevant line parameters

2. to specify whether a work station printer is present and to define its interface type

3. to specify whether other work station devices are present and to define the relevant parameters

4. to specify the version and type of the keyboard table to be used

5. to specify the type of screen to be emulated (alphanumeric or graphic), the time out in receive mode, and whether the PC is an M19

6. to end the OLICONF run and return to MS-DOS.

The general logic of the program is shown opposite. Choosing an option (except **6**) gives rise to a single display or sequence of displays, which either prompt for information or present stored information.

Simple error messages appear if responses to prompts have invalid syntax or are incorrect, and the user must correct his input before he can continue. Each response accepted causes the next prompt to appear. The different displays are described below.

Note: Each menu is presented with default values. The last row of the menu itself asks if one or more of the displayed values is to be modified, as follows:

set up parameters (Yes-No) =

The user may:

- enter **Y** in reply to the set up prompt and return to the first question of the menu, where the default values can either be confirmed (entering **/CR/**) or modified as required

- enter **N** in reply to the set up prompt, thus confirming all the displayed values and giving rise to the next OLICONF menu.

OPTION 1 (COMMUNICATION PARAMETERS) PROMPTS

```
   **********  HOST LINE (M40) CONFIGURATION PARAMETERS  **********

async. comm. port ----------------------(1=motherb, 2/3=extens)  =  1
interrupt ----------------------------------------------------  =  4
address I/O port ---------------------------------------------  =  3F8
speed ------- (50-75-110-150-300-600-1200-2400-4800-9600-19200)  =  9600
current loop ------------------------------------------ (Yes-No)  =  N
ignore DSR -------------------------------------------- (Yes-No)  =  N
ignore CTS -------------------------------------------- (Yes-No)  =  N
parity ------------------------------------------(None-Odd-Even)  =  0
stop bit ------------------------------------------------(1-2)  =  1
character length ------------------------------------(5-6-7-8)  =  8
Transmit Control ----------------------(None-Xon-Break-DSR-CTS)  =  N
Receive  Control ----------------------(None-Xon-Break-DTR-RTS)  =  X
```

Current values for the RS232 line used to connect the PC to the L1 MOS system are displayed. These values can be either confirmed or modified, bearing in mind that:

- values entered are only accepted if belonging to the admitted ranges (which are displayed in brackets). Agreement between them and those given when configuring the RS232 line in question on the L1 MOS system is the the user's responsibility
- values given for the interrupt and address I/O parameters must be coherent with the hardware configuration present
- the recommended value for speed is 9600 bps (19200 bps can only be used if CONTSW will not be used)
- if the Current Loop connection is used, the DSR and CTS signals are ignored
- the character length must be 8 bits (as required on the L1 MOS system)
- the Xon(/Xoff) control cannot be specified in transmission.

```
first  aux. RS232 line ---------------------------------(Y/N) = Y
second aux. RS232 line ---------------------------------(Y/N) = Y
```

Y If the auxiliary serial line is present (which is used to connect work station devices such as printer, MICR, ...).

N If the auxiliary serial line is not present.

In case of affirmative answer, the preceding menu is is displayed again, thus allowing to check and, if necessary, to modify parameters' values relevant to each auxiliary line. All the comments made for the menu above apply to these menus, too.

The proposed default values are as follows:

```
First  auxiliary line:  2,3,2F8,9600,N,Y,Y,E,1,7,N,X
Second auxiliary line:  3,3,1F8,2400,N,N,N,N,1,8,N,X
```

It must be remembered that all the values given in this phase can be modified by values given when configuring (on the L1 MOS system) the MUX channel used to connect the PC (except the interrupt and the address I/O values).

OPTION 2 (PRINTER PARAMETERS) PROMPTS

PRINTER PARAMETERS

```
work station printer -----------------------------------(Yes/No) = N
printer connection ----------------------------(Serial/Parallel) = S
aux. RS232 line ----------------------------------(First-Second) = F
ESC 0 required on Hard Copy ----------------------------(Yes/No) = N
```

The type of printer can be parallel (that is, a printer of /B type - IBM compatible, connected to the PC via the parallel interface) or serial (that is, a printer of /A type - Olivetti compatible, connected to a serial interface). Only serial printers (of the /A type) can be handled by OLIEMU.

It must be specified to which auxiliary line the printer is connected, and whether the printer requires an ESC 0 before the execution of an hard copy or not.

OPTION 3 (OTHER PERIPHERALS PARAMETERS) PROMPTS

```
badge reader --------------------------------------------(Yes/No) = Y
PIN Pad -------------------------------------------------(Yes/No) = Y
```

It must be specified if a badge reader (or reader/writer) and/or a PIN Pad are connected to the PC or not.

OPTION 4 (KEYBOARD SELECTION PARAMETERS) PROMPTS

```
National keyboard -(DENMARK-FINLAND-FRANCE-GERMANY-HEBREW-ITALY-NORWAY
                    PORTUGAL-SPAIN-SPAIN2-SWEDEN-SWITZERLAND-UK-USA-) = USA

Keyboard layout ---------------------------(0 = PC   1 = PB 105 keys) = 0
```

aa...a/CR/ The name of the national keyboard used on the PC must be entered.

n The type of keyboard used on the PC must be selected (entering 0 if it is a standard PC keyboard, or 1 if it is the PB keyboard with 105 keys).

OPTION 5 (OTHER PARAMETERS) PROMPTS

```
alpha/graphics terminal -------------------------------------(A-G) = A

timeout in receive (0 = infinite wait) --------------------------- = -1

M19 personal computer ------------------------------------(Yes/NO) = N
```

a The character identifying the type of emulated terminal: either A (if no graphics feature are required) or G (if graphics are used).

n/CR/ The number of polling cycles that the emulator must execute, when it is waiting for a character needed for a file transfer or a command, before signalling an error in reception. Accepted values are in the range from -1 to 65536, where -1 (which is the suggested value) means 65536 and 0 means infinite wait.

a Either Y, if the PC used is an M19, or N, if it is another PC.

OPTION 6 (EXIT)

Choosing this option ends the OLICONF run and returns to MS-DOS.

PARSER is a non-interactive utility that creates a file in object format for use in configuring the MOS user environment. Input to PARSER is a source file containing configuration directives previously created using the Editor (as described in the Chapter "Preparation of the MOS User Environment" above).

The object file is read by the system process Grandpa at startup time. It supplies information which enables the initialisation and activation of:

- interactive programs which will provide working environments at the work stations
- non-interactive programs in three different priority classes
- user function packages which will provide libraries of functions for general use
- initial and final programs run at startup and shutdown time respectively.

There is also information on which work stations are recognised and therefore usable, which have "master" status, and the facilities provided by each work station.

For a complete discussion of the user environment, the creation of source files for PARSER, and the functions of Grandpa, see Chapter 10.

PARSER filename1 filename2 [**"C**]

---

where:

filename1 is the name or path name of a byte-stream source file previously created using the Editor.

filename2 is the object file created by PARSER.

"C provides optional check output.

The file name filename1 must already exist. The file name filename2 is created by PARSER as the object file. If this name already exists, it will first be removed by PARSER. The object file comprises a set of configuration records, each pertaining to a single l-module (program or package) or to a single work station. At system startup time this file must be in the system volume /IPL/SYS with the name $CONFGP, as this is where Grandpa expects to find it.

The check option "C should be specified when a source file is first converted to object format, as neither PARSER nor Grandpa carry out error diagnosis. An invalid source file produces an invalid object file which, if subsequently used, will cause the system either not to function at all, or not to function as intended.

The "C option displays the configuration file as interpreted by PARSER, that is, the object file reconverted to source format, so that a comparison can be made with the original source file. If the output is likely to be too long to fit on the screen, output can be redirected to a file by adding "> filename3" to the command string.

The REBUILD command operates in the system configuration environment created by SYSCONF (see Chapter 6). The REBUILD command permits the regeneration of some modules originally created by SYSCONF or LINECONF. This could be achieved using the update option of SYSCONF/LINECONF, but using REBUILD avoids repeating the interactive phase of SYSCONF/LINECONF.

To run REBUILD, the working directory must be set to the FSUPROG directory (that is, **SETWDIR /IPL/RELn.1/CONF_GEN/FSUGEN/FSUPROG** , where n is the release number and 1 is the release level).

```
         ┌ C lineconf_user_dir                               ┐
         │                                                   │
REBUILD  ┤ I sysconf_user_dir                                ├
         │                        ┌ N                  ┐     │
         └ S sysconf_user_dir     ┤                    ├     ┘
                                  └ C lineconf_user_dir┘
```

---

where:

C lineconf_user_dir requests regeneration of the $CHM 1-module in the directory defined when it was created during a LINECONF session

I sysconf_user_dir requests regeneration of the $IPL 1-module in the directory defined when it was created during a SYSCONF session

S sysconf_user_dir N requests regeneration of the SYS volume for a stand alone system (that is, without communication components) in the directory defined when it was created during a SYSCONF session

S sysconf_user_dir C lineconf_user_dir requests regeneration of the SYS volume for an on-line system in the directory defined when it was created during a SYSCONF session, including in that volume the $CHM 1-module contained in the directory defined when it was created during a LINECONF session.

Regeneration of the SYS volume can only be made if its creation had been requested at SYSCONF time. This does not apply to regeneration of the $IPL and $CHM 1-modules, whose regeneration using REBUILD can be made also if they had not been created at SYSCONF/LINECONF time.

The regeneration phase is executed in batch. A message is displayed when the batch execution starts and, when the batch phase is ended, the notification arrives in the form of a "wrtuser": "There is a message" is displayed on the system line, and information concerning the batch phase can be found in the .MESG file in the HOME directory of the user who ran REBUILD.

ERROR MESSAGES

ERROR IN PARAMETER

An invalid parameter has been given when activating REBUILD.

```
ERROR IN PARAMETERS NUMBER
USAGE IS:
 REBUILD I <sysconf-user-dir>    (TO PRODUCE $IPL)
 REBUILD C <lineconf-user-dir>   (TO PRODUCE $CHM)
 REBUILD S <sysconf-user-dir> N  (TO PRODUCE SYSVOLUME WITHOUT CSS)
 REBUILD S <sysconf-user-dir> C <lineconf-user-dir>  (TO PRODUCE
                                                   SYSVOLUME WITH CSS)
```

The wrong number of parameters has been given when activating REBUILD.

The SPCONF command installs and configures the spooling system. It is a menu-driven interactive utility with options to:

- install all spooling system components
- build a spooling configuration file
- delete a spooling configuration file
- verify a spooling configuration file.

SPCONF

The spooling system can handle different types of system printers, allowing different printing policies depending on the physical printers (laser, high quality, high speed, ...). Each printer can be assigned a different print dispatcher program ("unspooler") by means of the configuration file. To install the spooling files, the user must specify a source directory containing the modules:

SPGPA = spooling system control program

QUEMAN = Queue Manager

He must also specify the path names of all unspoolers. Standard unspooler programs are supplied with the following path names:

/IPL/DPC/SERVICE/UNSPQLY for high quality printers

/IPL/DPC/SERVICE/UNSPOOL for not high quality printers

/IPL/DPC/SERVICE/WSUNSPQ for work station printers

If a spooling configuration file does not already exist, the second option (build a configuration file) is automatically appended to the first (install a system).

When the Spooling System and the LMS are required, the QUE_LMS 1-module must be loaded by Grandpa via CALL or PCALL (see the section "The Queue Manager" in chapter 10). The QUE_LMS file must be renamed as QUEMAN <u>before</u> using the SPCONF command (see the INSTALLQM command above in this chapter).

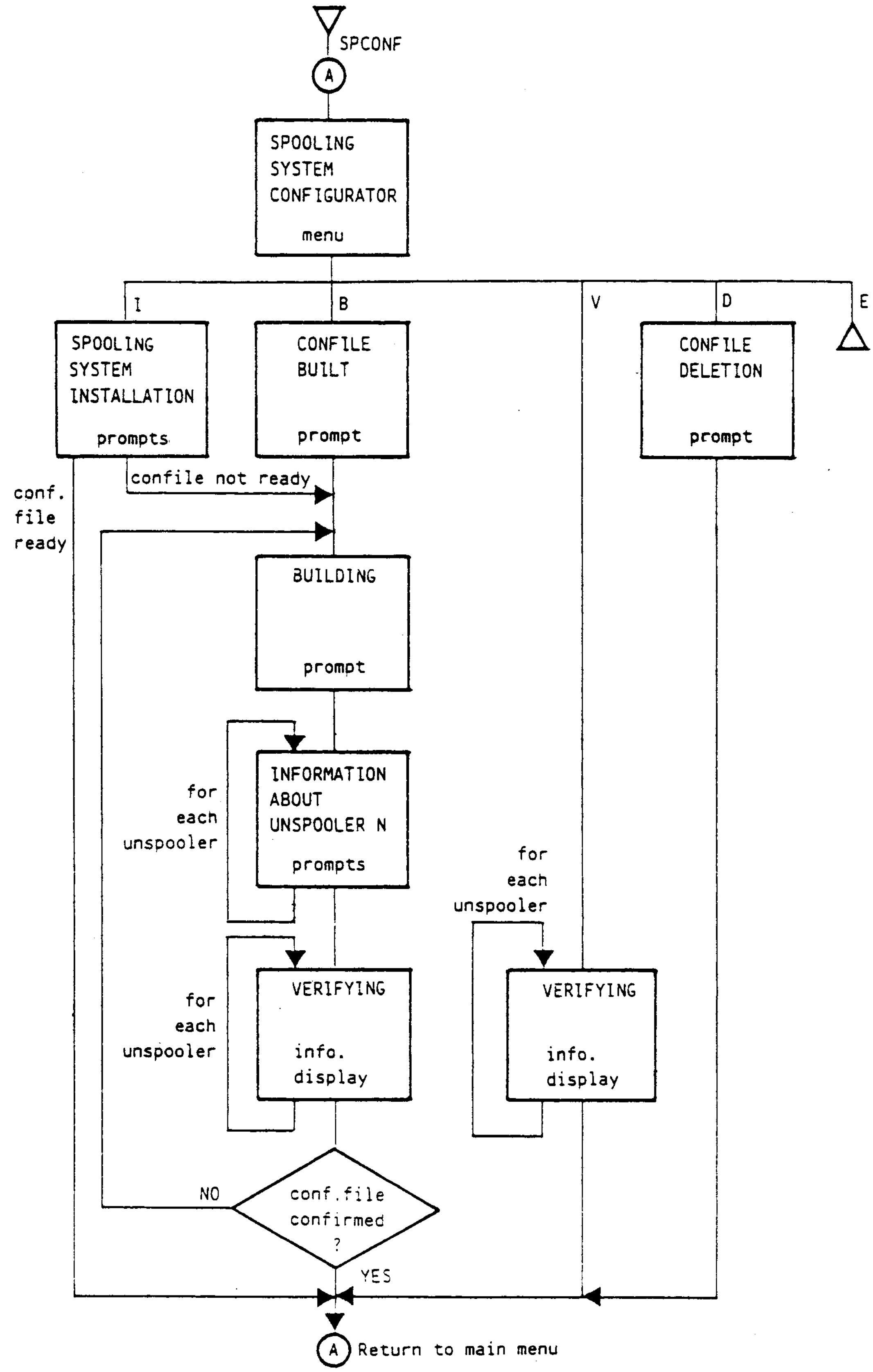


Fig. 11-13 General Logic of the SPCONF Command

SPCONF FUNCTIONING

The SPCONF menu offers five options:

I   Install the spooling system

B   Build a spooling configuration file

V   Verify a spooling configuration

D   Delete an existing spooling configuration file

E   End the SPCONF run and return to Shell

The general logic of the program is shown opposite. Choosing an option (except End) gives rise to a single display or sequence of displays, which either prompt for information or present stored information. The install, build and verify functions are interrelated:

- If a spooling system is installed with no configuration file prepared, the build function is enforced.

- After building a configuration file, the verify function is automatic, and the contents of the file must be confirmed.

Simple error messages appear if responses to prompts have invalid syntax or are incorrect, and the user must correct his input before he can continue. Each response accepted causes the next prompt to appear. The different displays are described below.

SPOOLING SYSTEM INSTALLATION

SERVICE LIBRARY PATH_NAME >

usually /IPL/DPC/SERVICE — The path name of the directory containing SPGPA and QUEMAN should be entered.

DESTINATION PATH_NAME >

usually /IPL — The path name of the MOS directory where the spooling system is to be installed should be entered.

ARE THERE ALREADY EXISTING QUEMAN STRUCTURES ? (Y/N) >

Y An affirmative response gives rise to the next prompt.

N The next prompt is skipped.

DO YOU WANT TO REMOVE ? (Y/N) >

Y All files in directories SP and $QM are removed, and then the directories themselves.

N The files SPGPA (in SP) and QUEMAN (in $QM) are overwritten; all other files and directories are left untouched.

TRANSFERRING ....

IS THE CONFILE READY ? (Y/N) >

Y An affirmative response takes you to the first prompt of the build option.

N A negative response takes you to the second prompt of the build option (first prompt skipped).

BUILD OPTION PROMPTS

CONFILE BUILT — The header only appears in the build option.

CONFILE PATH_NAME >

xx..x/xx..x — The path name of an already existing configuration file (install option with existing configuration file, after which control returns to the main menu), or the path name to be given to the configuration file that will be built (build option).

Note: This question is not asked if the install option has been executed with no existing configuration file, as the file built is automatically put under the path name /IPL/SP/CONF in this case.

BUILDING ....

UNSPOOLERS NUMBER (MAX 9) >

You must give the total number of different printing programs ("unspoolers") that will be used. This number is the same as the number of printers in the hardware configuration which will be handled by the spooling system.

INFORMATION ABOUT THE UNSPOOLER 1 (or N)

UNSPOOLER PATH_NAME >

The path name of the first (nth) unspooler must be specified. The response must be a complete local path name.

UNSPOOLER NUM (UNSPOOL<NUM> NUM=1..9)>

(Keyed parameter NUM) 1..9 — The sequence number of the unspooler must be given as an integer in the range 1 to 9.

PRINTER NUMBER (? FOR HELP) >

1..8 : SYSTEM PRINTER
A..Z : WORK STATION PRINTER
0 : ANY OTHER PRINTER

(Keyed parameter PRT)	0..8 A..Z	The second two lines are the HELP display which disappears when a valid response is entered. The letter in the range from A to Z identifies the work station to which the printer is connected. It is the responsibility of the user to guarantee that the printer specified here (where 0 is for a laser printer) is suitable for the unspooler selected.

DESCRIPTION FOR THE USER (MAX 40 CHAR)>

(Keyed parameter DES)	xx..	You can record any information here for recall by the verify function.

MORE PARAMETERS ?(Y/N) >

Y	You can specify further keyed and then positional parameters by means of the prompts shown below before entering the automatic verify phase. See below the section "Standard Unspoolers Parameters".
N	The verify phase is entered immediately (no user-defined parameters).

KEY NAME (MAX 6 CHAR) (HIT <CR> TO EXIT)>

xx..	For each keyed parameter required, a name must be specified (up to 6 characters). The next prompt then appears.
<CR>	No (more) keyed parameters. The next prompt is skipped.

PARAMETER VALUE (MAX 60 CHAR) >

xx..	For each keyed parameter, its value must be entered as a string of up to 60 characters. The previous prompt is then displayed again for the next parameter to be specified if any.

```
POSITIONAL PARAMETER (MAX 60 CHAR)
(HIT <CR> TO EXIT)                        >
```

xx.. The value of each positional parameter must be specified, and this prompt is repeated until the user presses carriage return to end the parameter entry phase for the current unspooler. No positional parameter is required by the standard unspoolers.

The unspooler parameter entry prompts (INFORMATION ABOUT THE UNSPOOLER N display) are repeated for each unspooler in turn, and then the verify displays start automatically.

VERIFY DISPLAYS

```
        VERIFYING ....
        --------------

THE CONFIGURATION FOR THE UNSPOOLER 3:

UNSPOOLER PATH_NAME      = /IPL/DPC/MYUNSPOOL

KEY PARAMETER PRT        = 2

KEY PARAMETER NUM        = 2

KEY PARAMETER DESCR      = VERSION 4

KEY PARAMETER BAN        = L
```

This display is available for each unspooler in turn. After each display the user can decide whether to continue or not.

```
DO YOU WANT TO CONTINUE  ? (Y/N)       >
```

Y The set of information for the next unspooler is displayed.

N No further verify displays. If a new file is being verified, confirmation is requested. If an existing file was being examined (verify option), the exit prompt appears instead.

DO YOU CONFIRM THIS CONFILE ? (Y/N) >

Y The new file is stored under the name specified at the beginning of the build option or /IPL/SP/CONF if the install option has been executed, and control returns to the main menu.

N Control returns to the start of the parameter entry sequence in the build option (BUILDING... heading) to give the user another chance to compile a configuration file.

HIT <CR> TO EXIT >

As soon as carriage return is pressed, control returns to the main menu.

DELETE OPTION PROMPT

CONFILE DELETION

CONFILE PATH_NAME >

/xx.. The response must be the path name of a file that exists. The file is deleted and control returns to the main menu. If the specified file does not exist, an error message is displayed.

INVALID PATHNAME
ERROR. HIT <CR> >

As soon as carriage return is pressed, control returns to the main menu.

STANDARD UNSPOOLERS PARAMETERS

The following parameters are required for the standard unspoolers:

- printer number (<>0)
- unspooler number
- description (<40 characters)
- keyed parameter BAN for type of banner. Possible values are:
  . L for long banner (132 characters)
  . S for short banner (80 characters)
  . N for no banner
- keyed parameter RES for release condition. Possible values are:
  . Y to release the printer when its queue is empty
  . N to release the printer when the unlink operation (Deassign) is performed
- keyed parameter SETFF for form feed option. Possible values are:
  . Y to insert a form feed command after each printed job
  . N for no form feed after each printed job
- keyed parameter MWS for interaction with the local master work station. Possible value is:
  . LOC to force interaction between the spooling system and the local master work station. If the MWS parameter is not specified, then the spooling system interacts with the global master work station.

The default values for the keyed parameters are:

- BAN = 'L'
- RES = 'N'
- SETFF = 'N'
- MWS = global master work station.

BAN = 'N' **Note:** If files are printed singly, that is, one per job, "BAN='N'" also omits the "new-page" command at the end of each file. A succession of one-file jobs will therefore commence each file-print on the page where the previous print ended.

LPR (Shell Command) Note: If, using this command, more than one file is to be printed in the same job, a "new-page" command is inserted at the end of each file-print, so that each file begins on a new page. If, using LPR, "no-copy" is specified, the "new-page" command is omitted and the files are printed as though they were a single file.

CONFIGURATION ERRORS

Should any problem arise when Grandpa starts the spooling system, the following error messages can be found in the file used as standard output by the spooling system itself. As the spooling system is a non-interactive program, the file in question is named /DEV/BKGn, where "n" is the number of the family executing the SPGPA program. It can normally be found by using the SHFAM Shell command. However, if the activation of the spooling system has failed, "n" may be deduced from the entries in the $CONFGP file, where the SPGPA program is the "n"th non-interactive program.

PRINTER x NOT AVAILABLE

The indicated printer (x = 1, 2, ..., 8), which was been declared to the spooler (Build option), is not available.

PRINTER OF WORKSTATION aaaa NOT AVAILABLE

The printer connected to the indicated work station (aaaa = TTYA, TTYB, ...), which was been declared to the spooler (Build option), is not available.

UNSPOOLER y NOT STARTED

The indicated unspooler (y = 1, 2, ..., 8), which was been declared to the spooler (Build option), is not available.

The WSECONF command configures the **WSELAN** emulator.

It is a menu-driven interactive utility with options to:

- define the type of connection between the PC and the L1 MOS system, and the relevant line parameters

- define the work station printer's parameters, if any

- define the other work station devices' parameters, if any

- select the version and type of keyboard table

- define further parameters, such as type of PC, type of emulated screen, etc.

WSECONF

WSECONF must be run on the PC to be used as an L1 work station, and it must reside in the MS-DOS directory containing the WSELAN emulator program.

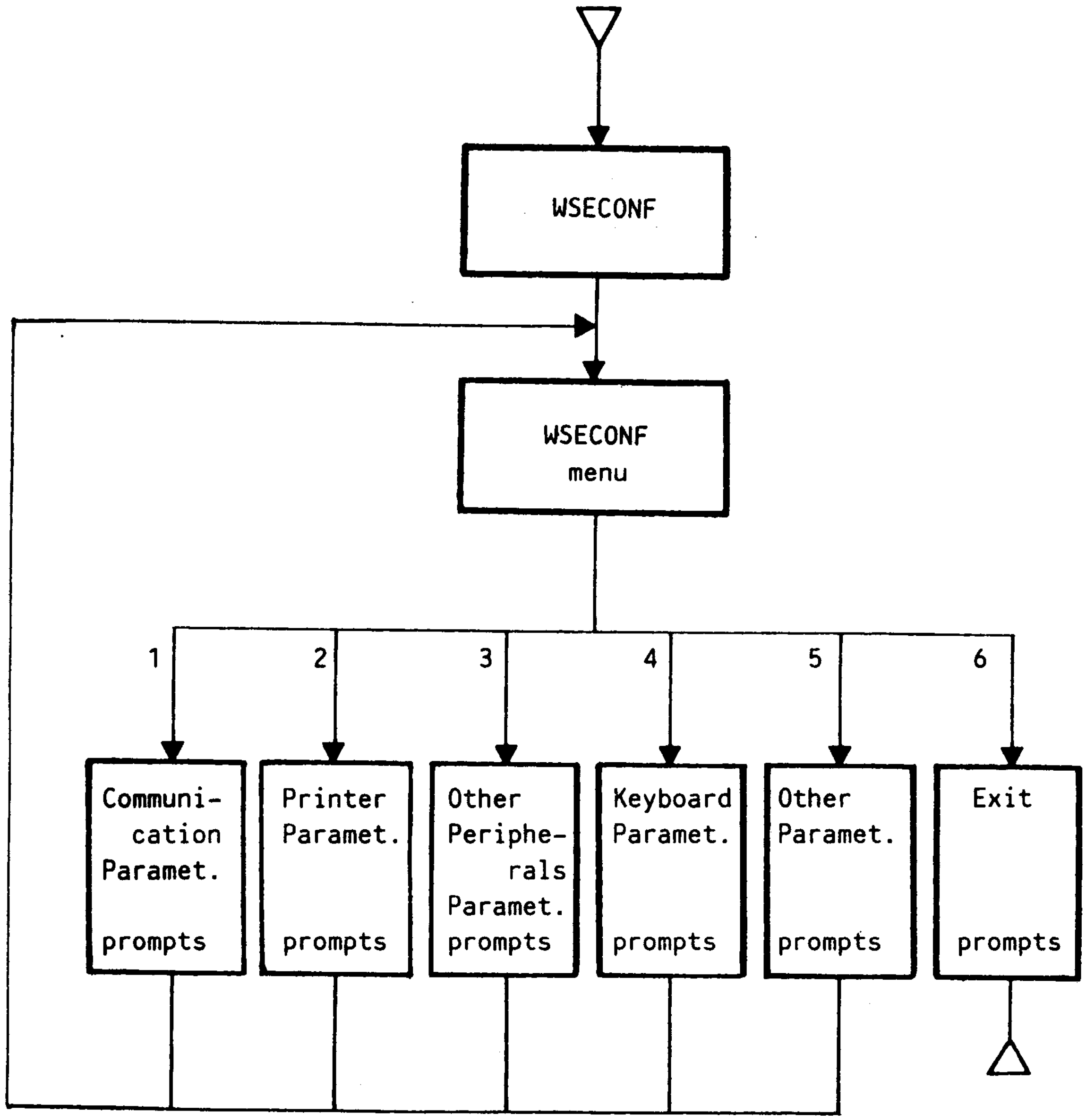


Fig. 11-14 General Logic of the WSECONF Command

WSECONF FUNCTIONING

The WSECONF menu offers six options:

1. to define the type of connection between the PC and the L1 MOS system and the relevant line parameters
2. to specify whether a work station printer is present and to define its interface type
3. to specify whether other work station devices are present and to define the relevant parameters
4. to specify the version and type of the keyboard table to be used
5. to specify the type of screen to be emulated (alphanumeric or graphic), the time out in receive mode, and whether the PC is an M19
6. to end the WSECONF run and return to MS-DOS.

The general logic of the program is shown opposite. Choosing an option (except **6**) gives rise to a single display or sequence of displays, which either prompt for information or present stored information.

Simple error messages appear if responses to prompts have invalid syntax or are incorrect, and the user must correct his input before he can continue. Each response accepted causes the next prompt to appear. The different displays are described below.

Note: Each menu is presented with default values. The last row of the menu itself asks if one or more of the displayed values is to be modified, as follows:

set up parameters (Yes-No) =

The user may:

- enter **Y** in reply to the set up prompt and return to the first question of the menu, where the default values can either be confirmed (entering **/CR/**) or modified as required
- enter **N** in reply to the set up prompt, thus confirming all the displayed values and giving rise to the next WSECONF menu.

OPTION 1 (COMMUNICATION PARAMETERS) PROMPTS

Physical interconnection ---------------(Ethernet-Omninet-Rs232) = O

- E If the Ethernet connection is used.
- O If the Omninet connection is used.
- R If the RS232 connection is used.

Regardless of the connection type selected, no further details concerning the transmissive media are requested, as they are given to the required driver when it is installed on the PC. This must be done before running the WSELAN emulator program.

OPTION 2 (PRINTER PARAMETERS) PROMPTS

PRINTER PARAMETERS

```
work station printer -----------------------------------(Yes/No) = N
printer connection ----------------------------(Serial/Parallel) = P
```

Only parallel printers (that is, printers of /B type - IBM compatible, connected to the PC via the parallel interface) can be configured as work station printer.

OPTION 3 (OTHER PERIPHERALS PARAMETERS) PROMPTS

No other peripherals are currently supported. An acoustic warning is emitted if this option is selected, and another option can be chosen.

OPTION 4 (KEYBOARD SELECTION PARAMETERS) PROMPTS

```
National keyboard -(DENMARK-FINLAND-FRANCE-GERMANY-HEBREW-ITALY-NORWAY
                    PORTUGAL-SPAIN-SPAIN2-SWEDEN-SWITZERLAND-UK-USA-) = USA

Keyboard layout ---------------------------(0 = PC   1 = PB 105 keys) = 0
```

aa...a/CR/ The name of the national keyboard used on the PC must be entered.

n The type of keyboard used on the PC must be selected (entering 0 if it is a standard PC keyboard, or 1 if it is the PB keyboard with 105 keys).

OPTION 5 (OTHER PARAMETERS) PROMPTS

---

alpha/graphics terminal ------------------------------------(A-G) = A

timeout in receive (0 = infinite wait) -------------------------- = -1

M19 personal computer ------------------------------------(Yes/NO) = N

a	The character identifying the type of emulated terminal: either A (if no graphics feature are required) or G (if graphics are used).
n/CR/	The number of polling cycles that the emulator must execute, when it is waiting for a character needed for a file transfer or a command, before signalling an error in reception. Accepted values are in the range from -1 to 65536, where -1 (which is the suggested value) means 65536 and 0 means infinite wait.
a	Either Y, if the PC used is an M19, or N, if it is another PC.

OPTION 6 (EXIT)

Choosing this option ends the WSECONF run and returns to MS-DOS.

6

PART VI : SYSTEM GENERATION AND CONFIGURATION USING SYSCONFR

INTRODUCTION TO PART VI

The use of the SYSCONFR procedure is described in this Part.

SYSCONF permits to generate and configure customer systems with predefined characteristics.

12. THE SYSCONFR PROCEDURE

The SYSCONFR Shell procedure is described in this Chapter. It is a tool **alternative** to SYSCONF (see Chapter 6) provided for configuring and generating stand alone systems (mainly in the BC area) belonging to a limited range of software and hardware characteristics.

The whole process is started by running the procedure which:

- asks the necessary information for configuring the system
- generates the system configuration file ($CON)
- generates the Grandpa configuration file ($CONFGP)
- generates a file containing information relevant to the configured system (FLINFO)
- generates the SYS volume
- generates the DPC volume
- generates the installation support
- generates a procedure which sets up the system and the application environments on the user hard disk (GENHD).

Remarks

The SYSCONFR procedure does not allow generation of the $IPL module. It assumes that the required $IPL module for the system being generated is already available in an archive directory (/IPL/RELn.1/ARCH_IPL_CHM/IPL_SCR, where n is the release number and 1 is the release level).

A specific procedure (SCRIPLBUILD, see below) is provided which produces the $IPL modules required by SYSCONFR and places them in the aforementioned directory.

SYSCONFR FILES

The disk file system structure in which SYSCONFR must operate is the following:

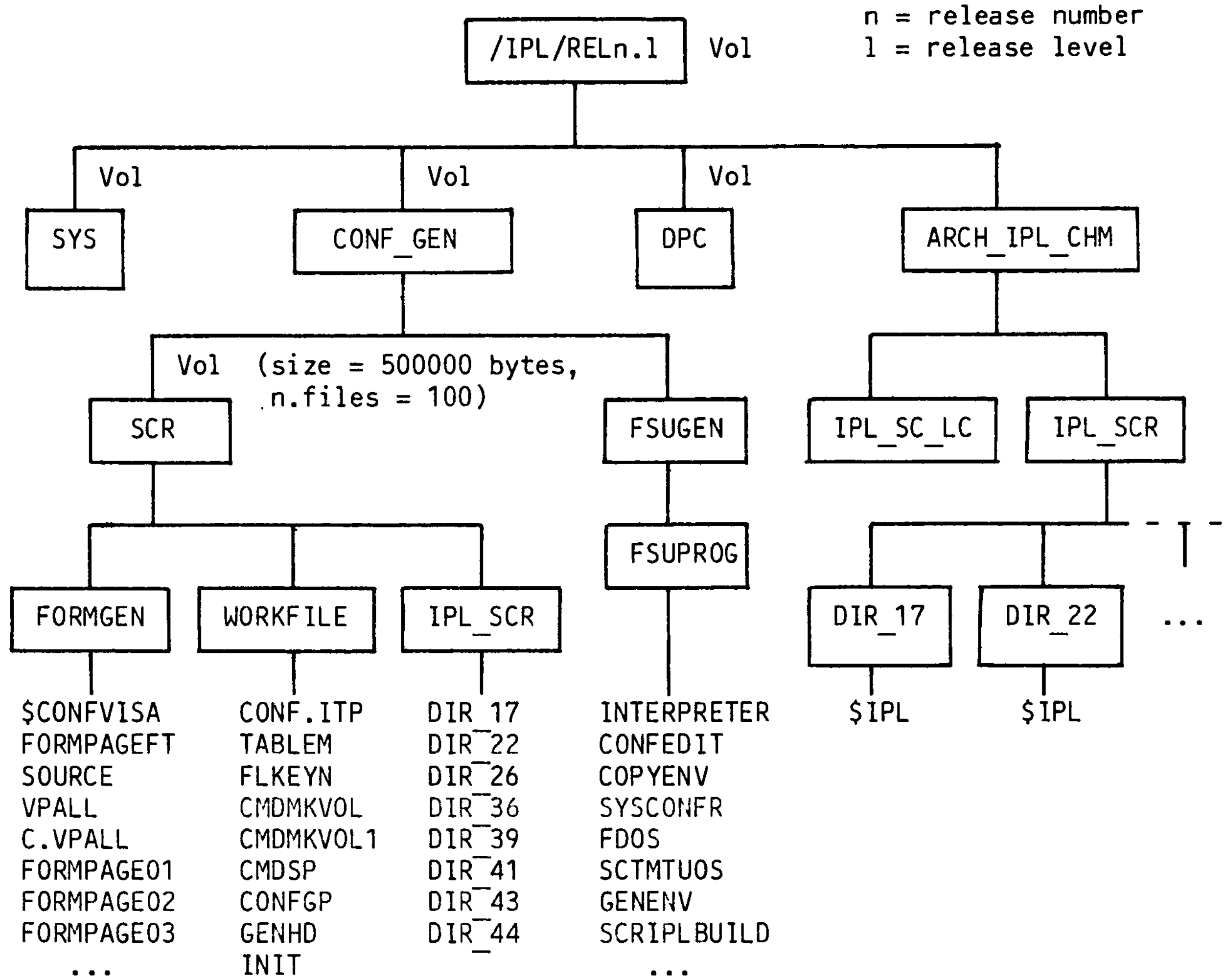


Fig. 12-1 SYSCONFR Disk File System

The /IPL/RELn.l/CONF_GEN/SCR/IPL_SCR directory contains the command files necessary to regenerate the $IPL l-modules required by SYSCONFR. This phase is carried out by the SCRIPLBUILD procedure, which must be activated by the user once before using SYSCONFR and which generates and places in the /IPL/RELn.l/ARCH_IPL_CHM/IPL_SCR directory all the $IPL l-modules expected by SYSCONFR.

Relationships between the various $IPL l-modules and the configurations which can be chosen in the menu three of SYSCONFR (see below) are listed in the following table. Any $IPL l-module contains, however, the drivers supporting monochromatic graphics, MUX, KDC and RS232 of CPU.

CONFIGURATION NUMBER	$IPL	NOTES
1	DIR_22/$IPL	For HD 27 MB.
2	DIR_22/$IPL	For HD 27 or 65 MB.
3	DIR_43/$IPL	For HD 65 MB.
3	DIR_44/$IPL	For HD 120 MB.
4	DIR_39/$IPL	For HD 65 MB.
4	DIR_17/$IPL	For HD 120 MB.
5	DIR_41/$IPL	For HD 65 MB.
5	DIR_26/$IPL	For HD 120 MB. + SCT
5	DIR_36/$IPL	For HD 120 MB. + MTU
6	DIR_36/$IPL	For HD 120 MB. + MTU

Fig. 12-2 $IPL L-modules Required by SYSCONFR

SCRIPLBUILD

The SCRIPLBUILD command must be used before activating the SYSCONFR procedure. The SCRIPLBUILD command generates one or more of the $IPL modules required by SYSCONFR, placing them where SYSCONFR expects them to be.

SCRIPLBUILD [**17**] [**22**] [**26**] [**36**] [**39**] [**41**] [**43**] [**44**]

---

where each numeric parameter identifies the directory which will contain a $IPL module (see above figure "SYSCONFR Disk File System").

Any combination of parameters is allowed. If none of them is given, SCRIPLBUILD creates all the eight $IPL modules which are required by SYSCONFR in order to produce any possible configured system.

Warning

Each $IPL module creation is carried out in batch mode: that is, a job is submitted to the batch monitor for each $IPL creation requested. Progress of the job(s) can be seen by the user using the normal Shell commands (BMQ, BATCH). Only when all the jobs are terminated can SYSCONFR be activated.

SCRIPLBUILD starts its execution by emitting the message:

..... PLEASE WAIT .....

followed by a message of the form:

***** $IPL_nn GENERATING *****

for each $IPL creation requested (where nn identifies the directory which will contain the $IPL module).

The general logic of the SYSCONFR is shown in the following two pages.

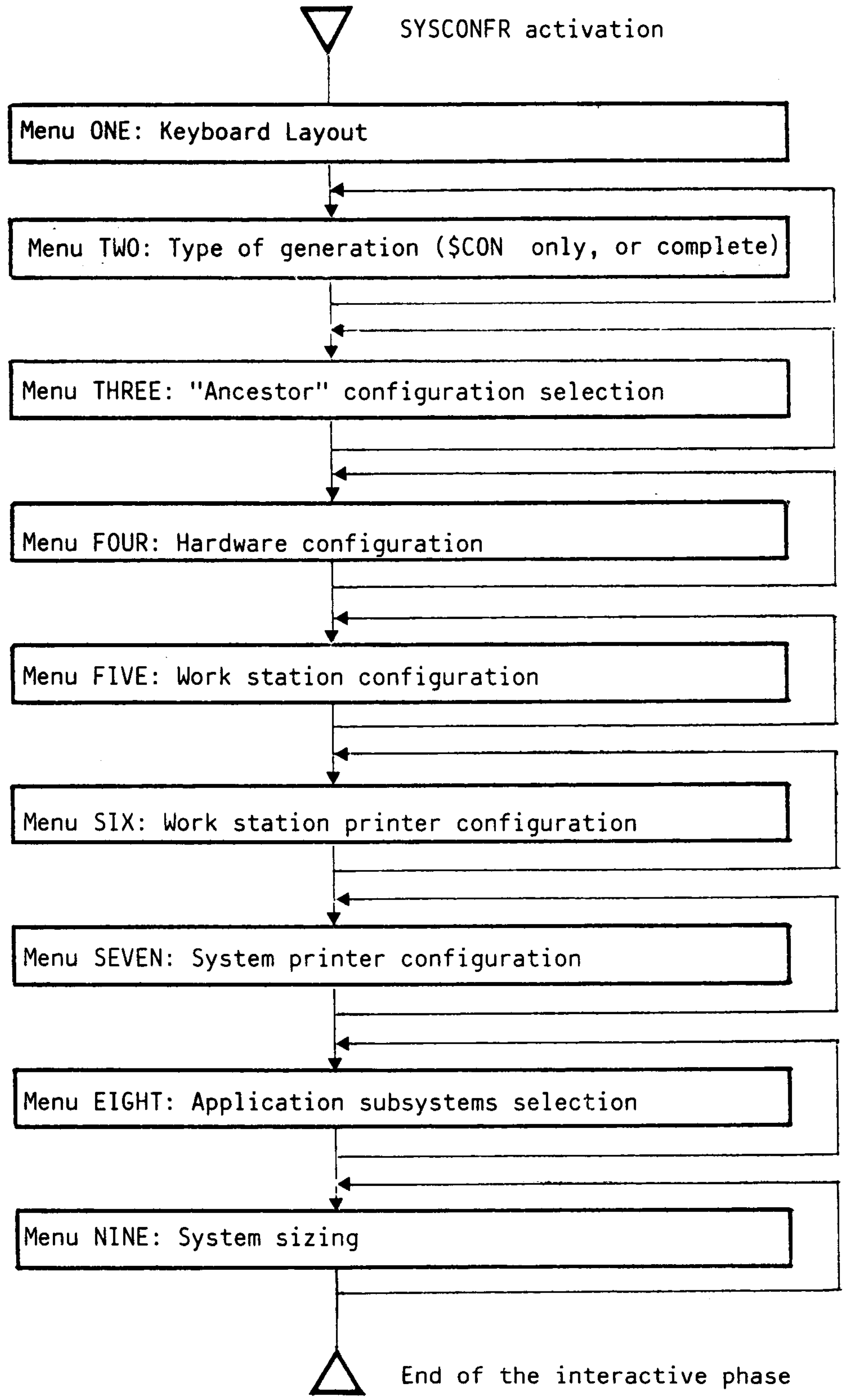


Fig. 12-3 SYSCONFR Logic Flow (cont.)

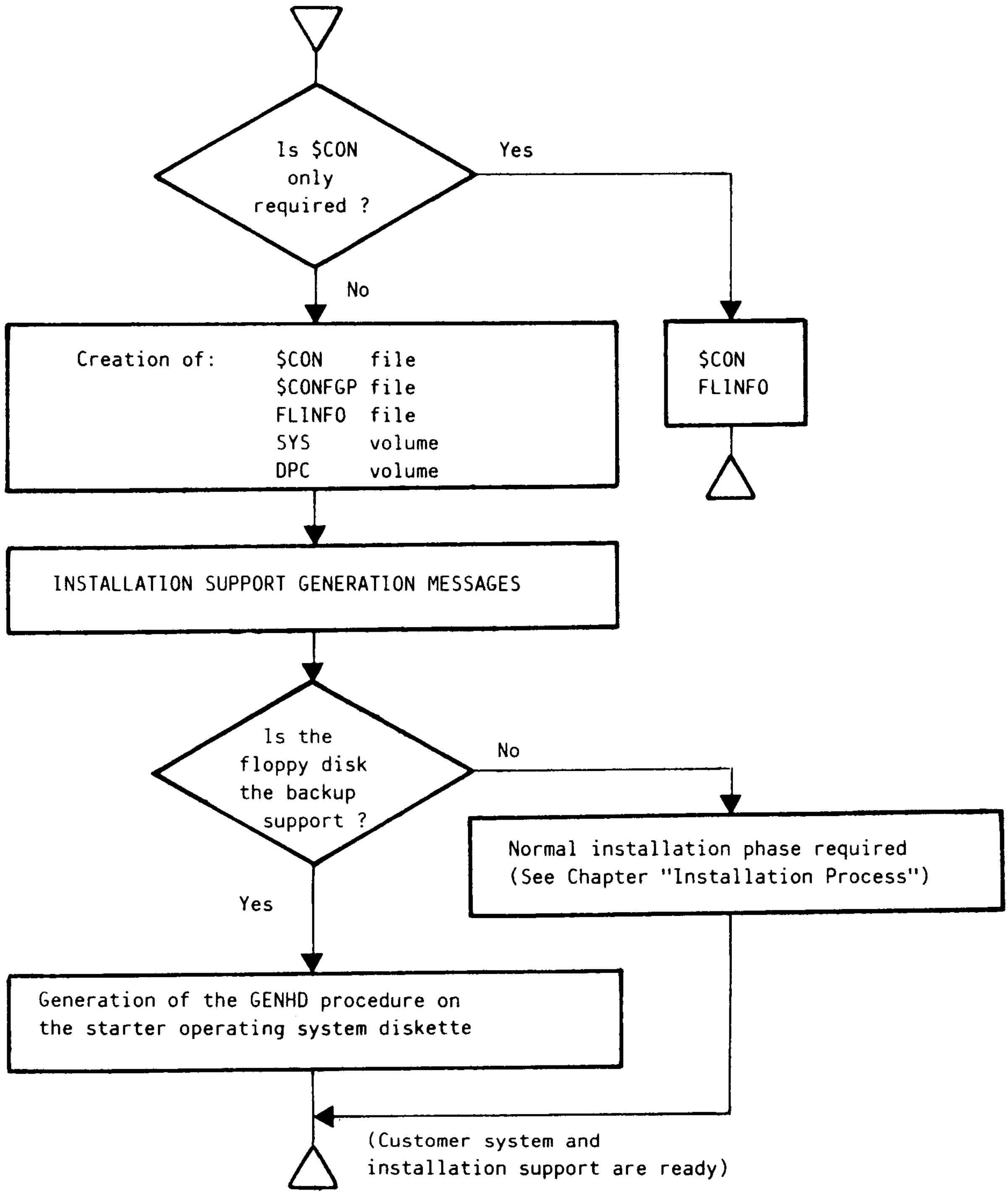


Fig. 12-3 SYSCONFR Logic Flow

EXECUTION

To run SYSCONFR, the working directory is set to the FSUPROG directory (that is, **SETWDIR /IPL/RELn.1/CONF_GEN/FSUGEN/FSUPROG** , where **n** is the release number and 1 is the release level) and the procedure SYSCONFR is run (that is, **SYSCONFR**).

An interactive session then begins, allowing the user to configure the system to be generated by selection from a set of menus (described below).

Warning: The SYSCONFR procedure must be run by ROOT, because it invokes some reserved Shell commands, and cannot be interrupted. This warning does not apply if only the $CON file is to generated.

MENU ONE

```
              MOS CONFIGURATOR FOR BUSINESS COMPUTER
                                                    KEYBOARD LAYOUT

                    | F1  | F2 | F3 | F4 | F5 | F6 | F7 | F8  | EXIT |
                    | CLEAR |                    | E  |    | |<-- |      |
                                                 | N  |    |      |      |
                                                 | T  |    |      |      |
                                                 | E  |    |      |      |
                                                 | R  |    |      |      |
 <CR> = next field        <--- = one character       | <--- | ---> |
 |<-- = back field               left                |      | HOME |
 F4   = next name on      ---> = one character
        the same field           right
CLEAR = clear field       HOME = first field
ENTER = last field   CTRL EXIT = abort by
                                 operator
                                        HIT <CR> TO CONTINUE :
```

"Menu one" is displayed in order to show SYSCONFR's functional keys and their meaning.

Functional keys appropriate to each menu will be shown in their help messages.

'Functional Keys' Function Details

FUNCTIONAL KEY	ACTION PERFORMED
CR	Confirms displayed value for current field and moves cursor to the next field.
|←	Moves cursor to the previous field.
F4	Enabled when the answer must be selected from a predefined set. Possible answers are proposed each time this key is pressed. Select one entering /CR/.
CLEAR	Deletes characters entered in the current field.
ENTER	Moves the cursor to the last field of the current menu.
←	Moves the cursor one character left.
→	Moves the cursor one character right.
HOME	Moves the cursor to the first field of the current menu. Values already entered in other fields are still valid.
CTRL EXIT	Aborts the SYSCONFR procedure, exiting to Shell.

Tab. 12-4 Functional Keys Meaning in SYSCONFR

From now on, when possible, SYSCONFR proposes default values for required answers: they can either be confirmed entering /CR/ or modified by the user.

Note:

Y entered in response to the **CONFIRM (Y/N):** prompt, makes SYSCONFR accept values already entered.

N deletes values already entered for the present menu, and places the cursor at the first field of the menu again.

MENU TWO

```
MOS CONFIGURATOR FOR BUSINESS COMPUTER

$CON GENERATION ONLY (Y / N) :

ENTER PATH_NAME OF SYS AND DPC VOLUMES :



WARNING :

THE VOLUME PATH_NAME INCLUDING SYS AND DPC VOLUMES IS THE FOLLOWING:

(user_path_name)/FREE


                                                    CONFIRM (Y/N) :

H E L P   M E S S A G E
```

An affirmative reply (**Y**) to the first question makes the second question unnecessary, and SYSCONFR goes on to the menu three. In this case, when the SYSCONFR activity has ended (that is, when all menus have been used), the user will be told where the $CON and the FLINFO files have been placed as follows:

```
**********************************************************************
*   $CON AND FLINFO FILES ARE RECORDED INTO THE FOLLOWING DIRECTORY:  *
*   /IPL/FILEGEN                                                      *
**********************************************************************
```

If the first answer is negative (**N**), then the user must respond to the second question entering the complete path name (starting with "/") of the volume where the FREE volume must be created (the SYS and DPC volumes generated later will be placed in this FREE volume).

MENU THREE

MOS CONFIGURATOR FOR BUSINESS COMPUTER BASIC MACHINE SELECTION						
CODE	BASIC MACHINE	NUMBER OF WORK STATION	MAGNETIC SUPPORT	BACKUP MAGNETIC SUPPORT	NUMBER OF WORK STATION PRINTER	NUMBER OF SYSTEM PRINTER
1	M40	1 - 4	HD 27 MB.	FD / SCT	0	1
2	M44	2 - 6	HD 27 MB. HD 65 MB.	FD SCT	0	1 - 2
3	M64	4 - 8	HD 65 MB. HD 120 MB.	FD / SCT MTU	1 - 2	1 - 3
4	M60	4 - 8	HD 65 MB. HD 120 MB.	FD / SCT MTU	1 - 2	1 - 3
5	M60/2	6 - 12	HD 65 MB. HD 120 MB.	SCT MTU	1 - 3	1 - 3
6	M60/3	10 - 16	HD 120 MB.	MTU	1 - 6	1 - 4
ENTER CONFIGURATION CODE (1 - 6): CONFIRM (Y/N) :						
H E L P M E S S A G E						

The user selects, from the possible configurations listed, the set up which most closely matches his requirements, by entering a number in the range from 1 to 6.

The maximum number of hard disks allowed in each configuration is as follows:

CONFIGURATION NUMBER	HARD DISK TYPE	NUMBER OF UNIT(S) ALLOWED
1	HD 27 MB.	1 - 2
2	HD 27 MB.	1 - 2
	or	
	HD 65 MB.	1 - 2
3	HD 65 MB.	1 - 2
	or	
	HD 120 MB.	1 - 2
4	HD 65 MB.	1 - 2
	or	
	HD 120 MB.	1 - 2
5	HD 65 MB.	1 - 2
	or	
	HD 120 MB.	1 - 4
6	HD 120 MB.	1 - 4

MENU FOUR

```
                MOS CONFIGURATOR FOR BUSINESS COMPUTER
                                           HARDWARE CONFIGURATION

   WORK STATION NUMBER ( X1 - X2 )                :

   WORK STATION PRINTER NUMBER ( Y1 - Y2 )        :

   SYSTEM PRINTER NUMBER ( Z1 - Z2 )              :

   MAGNETIC SUPPORT :
      Hard Disk  27 Mega Bytes Number ( 0 - 2 ) :
      Hard Disk  65 Mega Bytes Number ( 0 - 2 ) :
      Hard Disk 120 Mega Bytes Number ( 0 - U1) :

   BACK UP SUPPORT
      Floppy Disk ( Y / N )                     :
      Streaming Tape ( Y / N )                  :
      Magnetic Tape ( Y / N )                   :

                                               CONFIRM (Y/N) :

                        H E L P   M E S S A G E
```

Values here presented with X1, X2, Y1, Y2, Z1, Z2, and U1 indicate admitted ranges, and vary according to the selected configuration.

Only one type of hard disk may be selected (among those permitted for the selected configuration).

Besides the floppy disk, another back up support may be selected (among those permitted for the selected configuration). MTU and SCT are, therefore, mutually exclusive.

MENU FIVE

MOS CONFIGURATOR FOR BUSINESS COMPUTER WORK STATION CONFIGURATION				
W.S. NUMBER	W.S. TYPE	CONNECTION MODE	LINE TYPE	NATIONALITY
01	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
02	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
03	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
04	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
05	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
06	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
07	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
08	1 (Alphanum.)	1 (KDC)	2 (RS232)	1 (Italy)
09	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
10	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
11	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
12	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
13	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
14	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
15	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
16	1 (Alphanum.)	2 (MUX)	1 (C. L.)	1 (Italy)
			CONFIRM (Y/N) :	
		HELP MESSAGE		

The total number of configurable work stations varies according to the choices made from the preceding menu.

The keyboard nationality is determined by entering a code among those proposed in the "NATIONALITY" column when /F4/ is depressed, and refers to all the work stations of the configuration.

Information appropriate to the previously specified number of work station is requested: the list presented in the menu above refers to the maximum configuration allowed.

Values in brackets () are not selected by the user: they are displayed according to the code which has been selected. Default values are initially displayed in the menu.

Acceptable values for the current field are displayed when the /F4/ key is depressed. They are:

- national keyboards (see "The FLKEYN File" section below)

- work station types (Accepted values are: 1 for alphanumeric, 2 for graphic, 3 for PC)

- connection types (Accepted values are: 1 for KDC, 2 for MUX)

- connection modes (Accepted values are: 1 for Current Loop, 2 for RS232).

Notes

If the selected work station is a PC, then it is assumed to be connected to MUX via RS232.

If the selected work station is a graphic work station, then it is assumed to be a KDC.

If the selected work station is an alphanumeric work station, then it is assumed to be connected in one of the following modes:

- Integrated KDC

- Via MUX (RS232 or Current Loop)

MENU SIX

MOS CONFIGURATOR FOR BUSINESS COMPUTER WORK STATION PRINTER CONFIGURATION			
PRINTER NUMBER	PRINTER NAME	CONNECTION MODE	W.S. NUMBER CONNECTED
1	1 (PR1450)	2 (Oliv. Cont.)	
2	1 (PR1450)	2 (Oliv. Cont.)	
3	1 (PR1450)	2 (Oliv. Cont.)	
4	1 (PR1450)	2 (Oliv. Cont.)	
5	1 (PR1450)	2 (Oliv. Cont.)	
6	1 (PR1450)	2 (Oliv. Cont.)	
			CONFIRM (Y/N) :
H E L P M E S S A G E			

The total number of work station printers varies according to the choices made answering to the menu four.

Information appropriate to the previously specified number of work station printers is requested: the list presented in the menu above refers to the maximum configuration allowed.

In the right most column the user specifies the work station number to which the printer is connected. This number cannot be greater than the number of work stations specified in the menu three. SYSCONFR cannot produce configurations which include shared printers.

Values in brackets () are not entered by the user: they are displayed according to the code which has been selected.

Configurations 1 and 2 do not permit work station printers: therefore, if one of these configuration has been selected, the present menu will not be displayed.

Printers referred to work stations connected via MUX are assumed to be connected to the same controller ELB 3683 via MUX, as well.

Printers referred to work stations connected via KDC are assumed to be connected to the MUX, via RS232 line.

Acceptable values for the current field are displayed when the /F4/ key is keyed in. They are:

- selectable printers (see "The FLPRTN File" section below)

- protocol types (Accepted values are: 1 for Free Running mode, 2 for Olivetti Controlled mode).

MENU SEVEN

MOS CONFIGURATOR FOR BUSINESS COMPUTER SYSTEM PRINTER CONFIGURATION		
PRINTER NUMBER	PRINTER NAME	CONNECTION MODE
1	1 (PR1450)	1 (Free Running)
2	1 (PR1450)	1 (Free Running)
3	1 (PR1450)	1 (Free Running)
4	1 (PR1450)	1 (Free Running)
		CONFIRM (Y/N) :
H E L P M E S S A G E		

The total number of configurable system printers varies according to the choices made from menu three.

Information appropriate to the previously specified number of system printers is requested: the list presented in the menu above refers to the maximum configuration allowed.

Values in brackets () are not entered by the user: they are displayed according to the code which has been selected.

For configurations 1, 2 and 3 (M40, M44 and M64) the first system printer is always assumed to be connected to the RS232 line on the CPU board. Other system printers, if any, are assumed to be connected to MUX via RS232 line.

For configurations 4, 5 and 6 (M60, M60/2 and M60/3) all the system printers are always assumed to be connected to MUX via RS232 line.

Acceptable values for the current field are displayed when the /F4/ key is keyed in. They are:

- selectable printers (see "The FLPRTN File" section below)
- protocol types (Accepted values are: 1 for Free Running mode, 2 for Olivetti Controlled mode).

MENU EIGHT

```
MOS CONFIGURATOR FOR BUSINESS COMPUTER
                                   APPLICATION SUBSYSTEM SELECTION
--------------------------------------------------------------------

  COMPILED BASIC RUN ONLY (Y/N)................. :
  BASIC (Y/N)................................... :
  BASIC GRAPHIC (Y/N)........................... :
  COBOL RUN ONLY WITH VISA S6000 COMPATIBLE (Y/N) :
  COBOL RUN ONLY WITH VISA (Y/N)................ :
  COBOL ICE RUN ONLY (Y/N)...................... :
  BEAM RUN ONLY (Y/N)........................... :



                                         CONFIRM (Y/N) :
--------------------------------------------------------------------
                      H E L P   M E S S A G E
```

The user selects one or more application subsystems. SYSCONFR will include in the system being generated all the l-modules necessary to make up the selected application subsystems. Furthermore, the Shell application environment, the Batch Facilities and the Spooling System are always included in any configuration, by default.

As information about the application environment is needed by SYSCONFR in order to correctly size some system tables, this menu is also displayed when only the $CON file is being generated.

MENU NINE

```
+----------------------------------------------------------------+
|            MOS CONFIGURATOR FOR BUSINESS COMPUTER               |
|                                                  SYSTEM SIZING  |
+----------------------------------------------------------------+
|                                                                |
|   MEMORY VOLUME SIZE (K BYTES) ........... :                   |
|                                                                |
|   APPLICATION FILES NUMBER ............... :                   |
|                                                                |
|   PRIVATE BUFFERS POOL NUMBER ............ :                   |
|                                                                |
|                                                                |
|                                                                |
|                                            CONFIRM (Y/N) :     |
+----------------------------------------------------------------+
|                     H E L P   M E S S A G E                     |
+----------------------------------------------------------------+
```

The user enters the size for the memory volume (in kbytes), the total number of application files to be simultaneously handled and the total number of private buffer pools required. More detail about these information is given in the UNIT 1 description in Chapter 4.

SYSCONFR OUTPUT

When the interactive session is ended, SYSCONFR processes the information entered by the user and produces:

- the $CON file
- the $CONFGP file
- the FLINFO file (see below the "The FLINFO File" section).
- the SYS volume
- the DPC volume

SYSCONFR displays messages about the current activity:

```
..... TABLE CREATING .....
..... FLINFO CREATING .....
..... VOLUME /user_path_name/FREE CREATING .....
..... $CON FILE CREATING .....
..... SYSTEM VOLUME CREATING .....
..... DEPENDENT COMPONENT VOLUME CREATING .....
..... CORRECT TERMINATION .....
```

'SYS' AND 'DPC' VOLUMES GENERATION

The SYS and DPC volumes are created by SYSCONFR according to selections from menus.

The necessary l-modules are copied from the release archive into the SYS volume and, when they have been generated, the $CON, the $IPL and the $CONFGP files are copied into the SYS volume, too.

Similarly, the l-modules required by the configuration choices made in the menu eight are copied from the release archive into the DPC volume.

INSTALLATION SUPPORT GENERATION

When both the SYS and DPC volumes have been generated, they are saved on the back up support selected in menu four. If more than one support has been indicated, the tape based one is used (SCT or MTU).

On the same back up support where the SYS and DPC volumes are saved, a **starter operating system** is also generated. This will enable the initialisation of the system where the SYS and DPC volumes are to be installed starting from the back up support where they reside. Therefore, SYSCONFR invokes the MKSYS command (see Part V) if a tape based peripheral has been specified as back up support, or a set of other

Shell commands (MKENV, MKVOL, MKBOOT, MKLOGIN, ...) if the floppy disk is used. In this case, two (or more) diskettes are generated: the starter operating system and the SYS volumes reside on the first, the DPC volume on the other(s).

The user is asked to set up the appropriate peripheral by means of the messages listed below.

MESSAGES FOR FLOPPY DISK SUPPORT

```
**************************************************
*                                                *
*   ..... INSTALLATION SUPPORT GENERATION .....  *
*                                                *
**************************************************

    ..... STARTER OS GENERATING .....
 >> ..... INSERT FLOPPY DISK IN FL1 <<
          HIT <CR> TO CONTINUE :
```

The user must insert a floppy disk and then key in /CR/.

```
    ..... START MKENV .....
    ..... START MKVOL .....
    ..... START MKBOOT .....
    ..... START MKVOL .....
    ..... START MKLOGIN .....
    ..... START MKDIR .....
    ..... END OF STARTER OS GENERATING .....
    ..... SAVE DEPENDENT COMPONENTS IN FL1 .....
 >> ..... CHANGE FLOPPY DISK IN FL1 <<
          HIT <CR> TO CONTINUE :
```

The user must insert another floppy disk and then key in /CR/.

If more than one floppy disk is required to contain the DPC volume, the FLDUMP messages guide the user in changing the floppy disks. These messages are described in the Shell Commands, Reference Manual.

MESSAGES FOR STREAMING CARTRIDGE SUPPORT

```
**************************************************
*                                                *
*   ..... INSTALLATION SUPPORT GENERATION .....  *
*                                                *
**************************************************

 >> ..... INSERT CARTRIDGE IN STREAMING TAPE UNIT <<
          HIT <CR> TO CONTINUE :
```

The user must insert a cartridge tape and then key in /CR/.

```
   ..... START MKSYS .....
```

MESSAGES FOR MAGNETIC TAPE SUPPORT

```
**************************************************
*                                                *
*   ..... INSTALLATION SUPPORT GENERATION .....  *
*                                                *
**************************************************

 >> ..... INSERT TAPE IN MAGNETIC TAPE UNIT <<
          HIT <CR> TO CONTINUE :
```

The user must insert a magnetic tape and then key in /CR/.

```
   ..... START MKSYS .....
```

USER HARD DISK GENERATION

Once the installation support has been produced, the configured operating system can installed on the hard disk of the user machine.

This phase is described in the Chapter "Installation Process" above as far as SCT or MTU support are involved.

Instead, if the floppy disk has been chosen as back up support, a specific Shell procedure (**GENHD**) is prepared by SYSCONFR on the diskette containing the starter operating system. This procedure can be run by the user immediately after the initialisation phase: it sets up on the user hard disk the system and application environments selected through SYSCONFR. Only the creation of the volume which will contain the system and application environments (step number 4 in the "Installation

From Floppy Disk" section in the "Installation Process" Chapter) must be performed by the user before running the GENHD procedure. All the messages displayed by the GENHD procedure are listed below:

```
***************************************************
*                                                 *
*      HARD DISK USER ENVIRONMENT GENERATING      *
*                                                 *
***************************************************

>> ENTER HARD DISK NUMBER TO GENERATE USER ENVIRONMENT :
```

The user keys in the hard disk number where the OS must be installed.

```
..... START MKBOOT ON HDn .....
..... START MKLOGIN ON HDn .....
..... SYS VOLUME GENERATING ON HDn .....
..... DPC VOLUME GENERATING ON HDn .....
..... INSERT THE FIRST FLOPPY DISK OF DPC VOLUME IN FL1 .....
               >> HIT <CR> TO CONTINUE :
```

The user must insert the first floppy disk where the DPC volume resides and then key in /CR/.

If the DPC volume to be installed resides on more than one floppy disk, the FLDUMP messages guide the user in changing the floppy disks. These messages are described in the Shell Commands, Reference Manual.

The correct termination of the configured system installation on the user hard disk is notified by the message:

```
*************************************************************************
*                                                                       *
*                USER HARD DISK HAS BEEN GENERATED CORRECTLY            *
*                                                                       *
*************************************************************************
```

Finally, the user is requested to insert the floppy disk containing the starter operating system by the message:

```
INSERT STARTER OS IN FL1
HIT <CR> TO CONTINUE :
```

The user must insert the floppy disk containing the starter operating system.

THE FLINFO FILE

This file is created by SYSCONFR and contains all the answers entered by the user during the interactive part of SYSCONFR. It also contains a description of the hardware configuration compatible with the software configuration thus generated. Contents of this file can be useful to the subsidiary personnel and to Central Assistance technicians. This file is placed by SYSCONFR in the /user_path_name/SYS volume: it is therefore present on the hard disk of the machine where SYSCONFR is run, on the installation support generated, and on the hard disk of the user machine.

The detailed contents of the FLINFO file are shown in the following figure, where possible values are listed for each field.

HARDWARE CONFIGURATION

.GENERAL CONFIGURATION

Machine Type	: M40 M44 M64 M60 M60/2 M60/3
Work Station Number	: 1 - 16
Work Station Printer Number	: 1 - 6
System Printer Number	: 1 - 4
Hard Disk Type	: 1 - 2 HD 27 M. Bytes 1 - 2 HD 65 M. Bytes 1 - 4 HD 120 M. Bytes
Back_up Device	: Floppy Disk Streaming Cartridge Tape Magnetic Tape

. WORK STATION CONFIGURATION

1) WORK STATION N. 1

- SCREEN / KEYBOARD

Type	: M24 Graphic Alphanumeric
Nationality	: ....
Connected To	: MUX Controlled via ELB 3683 MUX Controlled Transp. Line KDC Integrated
Connection Mode	: Free Running Olivetti Controlled
Connection Type	: Current Loop RS232
MUX Line Number	: 1 / 2 / 3 / 4
Relative MUX / KDC Position	: A - R

Fig. 12-5 FLINFO File Structure (cont.)

- PRINTER

Name	: PR1450 / PR1460 / ......
Connected To	: MUX Controlled via ELB 3683 MUX Controlled Transp. Line
Connection Mode	: Free Running / Olivetti Controlled
Connection Type	: RS232
MUX Line Number	: 1 / 2 / 3 / 4
Relative MUX / KDC Position	: A - R
Number Of Stop Bits	: 1
Parity Type	: Even
Transmission And Reception Character Length	: 7 Bits
Speed	: 2400 / 4800
Tandem Mode	: XON - XOFF

1) WORK STATION N. 1

.
.
.

.
.
.

16) WORK STATION N. 16

.
.
.

Fig. 12-5 FLINFO File Structure (cont.)

. SYSTEM PRINTER CONFIGURATION

1) PRINTER N. 1

Name	: PR3300 / PR1460 / ......
Connected To	: CPU MUX Controller Transp. Line
Connection Mode	: Free Running / Olivetti Controlled
Connection Type	: RS232
MUX Line Number	: 1 / 2 / 3 / 4
Relative MUX / KDC Position	: A - R
Number Of Stop Bits	: 1
Parity Type	: Even
Transmission And Reception Character Length	: 7 Bits
Speed	: 2400 / 4800
Tandem Mode	: XON - XOFF

2) PRINTER N. 2

.
.
.

.
.
.

4) PRINTER N. 4

Fig. 12-5 FLINFO File Structure (cont.)

SOFTWARE CONFIGURATION

. APPLICATION SUBSYSTEM

Compiled BASIC Run Only	: Y / N
Basic	: Y / N
Basic Graphic	: Y / N
Cobol Run Only With Visa S6000 Compatible	: Y / N
Cobol Run Only With Visa	: Y / N
Cobol Ice	: Y / N
Beam Run Only	: Y / N

. SYSTEM SIZING

Families Number	: (Run Time Value)
Processes Number	: (Run Time Value)
Programs Number	: (Run Time Value)
Channels Number	: (Run Time Value)
Files Number	: (Run Time Value)
System I.D.S. Number	: (Run Time Value)
System Buffers Number	: (Run Time Value)
Buffer Size	: (Run Time Value)
Private Buffer Pools	: (Run Time Value)
Private Buffers Number	: (Run Time Value)
Memory Volume Pages	: (Run Time Value)
Memory Volume Files Number	: (Run Time Value)
Multiple Files	: (Run Time Value)
Locked Records	: (Run Time Value)
Locked Phantoms	: (Run Time Value)
Secondary Indices	: (Run Time Value)

Fig. 12-5 FLINFO File Structure

CONFIGURABLE SYSCONFR FILES

Some of the files listed in Fig. 12-1 above can be modified by the user. These files are described below together with the existing limitation in their manipulation.

Those files which are not described below must not be modified by the user, because they contain the required information in the format expected by SYSCONFR.

The FLKEYN File

This file contains the names of all national keyboards which can be selected from the menu five ("NATIONALITY" question).

The names originally contained in this file are:

```
DENMARK_DP
DENMARK_BAS
ENGLAND_DP
ENGLAND_BAS
FRANCE_DP
FRANCE_BAS
GERMANY_DP
GERMANY_BAS
GERMDIN_DP
GERMDIN_BAS
GREACE_DP
KANA_DP
KANA_BAS
ITALY_DP
ITALY_BAS
INTER_DP
ISRAEL_DP
LEUMI_DP
NORWAY_DP
NORWAY_BAS
PORTUGAL_DP
SPAIN_DP
SPAIN_BAS
SWFL_DP
SWFL_BAS
SWISG_DP
SWISG_BAS
SWISR_DP
SWISR_BAS
USAASCI_DP
USAASCI_BAS
USAOCA_DP
YUGOS_DP
USSR_DP
CCCP_BAS
NOKYO_DP
NOKYO_BAS
DENMARK_M
ENGLAND_M
FRANCE_M
GERMDIN_M
GREACE_M
ITALY_M
NORWAY_M
PORTUGAL_M
SPAIN_M
SWFL_M
SWISG_M
SWISR_M
USAASCI_M
YUGOS_M
SPAINZ_M
```

The user can modify this file, using the EDITOR, bearing in mind that:

1. the maximum number of names contained in it is 80

2. each name must be 11 characters long (fill shorter names with blanks)

3. each name must be followed by a carriage return (/CR/).

The FLPRTN File

This file contains the names of all printers which can be selected from menu six and seven ("PRINTER NAME" question).

The names originally contained in this file are:

```
PR_1450
PR_1470
PR_15
PR_17
PR_19
PR_3300
PR_3600
PR_340
PR_340/B
PR_11
PR_13
PR_38
PR_17/B
PR_15/B
```

The user can modify this file, using the EDITOR, bearing in mind that:

1. the maximum number of names contained in it is 60

2. each name must be 8 characters long (fill shorter names with blanks)

3. each name must be followed by a carriage return (/CR/).

The CONFGP File

This file is given by the SYSCONFR procedure to the PARSER command in order to generate the $CONFGP configuration file.

The original contents of this file are:

```
TTYA!TTYB!TTYC!TTYD!TTYE!TTYF!TTYG!TTYH!TTYI!TTYJ!TTYK!TTYL!TTYM!TTYN!TTYO!
TTYP:MCL=/IPL/SYS/$VSH;
TTYA:MASTER!DATE!SHUTDOWN;
CALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>PAR;
START:/IPL/BE/BATCHGPA;
START:/IPL/SP/SPGPA;
```

Furthermore, if the "BEAM run-only" application environment has been selected in menu eight, the following directives are also inserted by SYSCONFR in the CONFGP file:

```
PCALL:BEAMMON=/IPL/DPC/BEAM/BEAMMON,<BEAM>/IPL/DPC/BEAM/MAIN
                                    <BEAMDB>/IPL/DPC/BEAMDD;
ALLT:BEAM=/IPL/DPC/BEAM/MAIN;
```

The user is referred to the Chapter "Preparation of the User Environment" for any modification which can be made on this file.

SYSCONFR ERROR MESSAGES

..... TOO MANY PARAMETERS .....

..... USAGE : SCRIPLBUILD [17] [22] [26] [36] [39] [41] [43] [44] .....

More than eight parameters have been given in activating the SCRIPLBUILD procedure. No more than eight are allowed. No $IPL module has been been created.

..... INVALID PARAMETER(S) .....

..... USAGE : SCRIPLBUILD [17] [22] [26] [36] [39] [41] [43] [44] .....

One or more mistakes have been made in entering the SCRIPLBUILD procedure parameters. No $IPL module has been been created.

..... /IPL/RELn.1/ARCH_IPL_CHM/IPL_SCR/DIR_nn/$IPL : FILE NOT FOUND .....

..... USE "SCRIPLBUILD" PROCEDURE TO GENERATE IT .....

..... SYSCONFR PROCEDURE IS ABORTED .....

The $IPL module necessary for the selected configuration is not present in the directory where SYSCONFR expects it to be. The missing $IPL is that whose number nn is indicated as directory name (DIR_nn). The SCRIPLBUILD procedure must be used in order to create it, before running SYSCONFR again.

..... ABORT BY OPERATOR .....
..... GENENV PROGRAM IS ABORTED .....
..... SYSCONFR PROCEDURE IS ABORTED .....

The key combination /CTRL/ and /EXIT/ has been keyed in to interrupt SYSCONFR.
No output has been produced. Run SYSCONFR again.

```
..... VISA ERROR .....
            or
..... FS ERROR .....
..... GENENV PROGRAM IS ABORTED .....
..... SYSCONFR PROCEDURE IS ABORTED .....
```

An error detected either during the interactive phase of SYSCONFR or during an I/O attempt caused SYSCONFR to end abnormally.
No output has been produced. Run SYSCONFR again.

```
..... INTERPRETER STATUS ERROR .....
..... SYSCONFR PROCEDURE IS ABORTED .....
```

The INTERPRETER program (invoked by SYSCONFR to generate the $CON file) has not ended normally.
The SYSCONFR procedure must be run again.
The FLINFO file relevant to the aborted session can be found in the /IPL/FILEGEN directory.

```
>>> /user_path_name/FREE VOLUME ALREADY EXIST.  OVERWRITE (Y/N)?
```

The /user_path_name/FREE volume, where the newly generated SYS and DPC volumes are to be placed, already exists.

Y The existing /user_path_name/FREE volume is overwritten.

N The existing /user_path_name/FREE volume is not modified.
The FLINFO file relevant to the session can be found in the /IPL/FILEGEN directory.

```
..... SYSGEN PROCEDURE IS ABORTED .....
..... SYSCONFR PROCEDURE IS ABORTED .....
```

The SYSGEN procedure invoked by SYSCONFR to create the SYS volume has not been ended normally. The SYSCONFR procedure must be run again. The FLINFO file relevant to the aborted session can be found in the /IPL/FILEGEN directory.

```
..... DPCGEN PROCEDURE IS ABORTED .....
..... SYSCONFR PROCEDURE IS ABORTED .....
```

The DPCGEN procedure invoked by SYSCONFR to create the DPC volume has not been ended normally. The SYSCONFR procedure must be run again. The FLINFO file relevant to the aborted session can be found in the /IPL/FILEGEN directory.

```
..... INSTALLATION SUPPORT FAILED .....
..... SYSCONFR PROCEDURE IS ABORTED .....

*******************************************************
*   TO RETRY INSTALLATION SUPPORT CREATION, ENTER:    *
*                                                     *
*   FDOS   or    SCTMTUOS SC     or     SCTMTUOS MT1  *
*******************************************************
```

The installation support generation phase has not been terminated correctly.
The configured operating system (consisting of the SYS and DPC volumes, and of the $CON file, as well as the FLINFO file) has been generated.
The user can retry to save it on an installation support by entering one of the proposed commands (which are for floppy disk, streaming cartridge tape and magnetic tape respectively). The choice of the installation support must be done as is appropriate with the back up support available.
The chosen command must be entered when the working directory is the "user path_name" keyed in in menu two: this is the actual working directory when this error message is emitted by SYSCONFR.
Remember that SYSCONFR must be run by "ROOT" and that the peripheral involved in the installation support creation must be available (that is, not being used by another user).

```
*************************************************************************
*                                                                       *
*                  USER HARD DISK GENERATING FAILED                     *
*                                                                       *
*************************************************************************
```

User environment generation on hard disk starting from installation support has not ended normally. The GENHD procedure must be run again.
The configured operating system (consisting of the SYS and DPC volumes, and of the $CON file, as well as the FLINFO file) has been generated and it has been successfully transferred on the installation support, together with the starter operating system.

Besides the above messages, other error conditions detected during execution of a Shell command invoked by SYSCONFR (for example, COPY) may cause a Shell error message to be displayed. See the <u>SHELL Commands, Reference Manual</u> or the <u>Message Book</u> for their meaning.

7

PART VII : APPENDICES

INTRODUCTION TO PART VII

Appendix A contains information concerning the memory allocation.

Appendix B describes the correspondence between the user files and their internal mapping.

Some examples of system configuration are given in Appendix C.

A. MEMORY ALLOCATION

Information given in this Appendix can be useful when evaluating some parameters values.

UNIT 1

USRFAMILY

Allocation:

- Family table = 1 pointer (4 bytes) for each family descriptor.
- Family descriptor = 130 bytes (%82) monoprocessor;
  = 132 bytes (%84) multiprocessor;
- Automaton descriptor = 58 bytes (%3A) monoprocessor;
  = 66 bytes (%42) multiprocessor;
- Event interrupt = 22 bytes (%16) monoprocessor;
  = 24 bytes (%18) multiprocessor;
- Segment table = 180 bytes (%84) [30 entries (6 bytes) per family].
- Total: 394 bytes (%18A) monoprocessor;
  406 bytes (%196) multiprocessor.

NUMPROCESS

Allocation:

- Process table = 1 pointer (4 bytes) for every process descriptor.
- Process descriptor = 66 bytes (%42) monoprocessor;
  = 74 bytes (%4A) multiprocessor;
- Automaton descriptor = 160 bytes (%A0) monoprocessor;
  = 174 bytes (%AE) multiprocessor;
- Suspension event = 22 bytes (%16) monoprocessor;
  = 24 bytes (%18) multiprocessor;
- Segment table = 12 bytes (%C) [2 entries (6 byte) per process].
- Total: 264 bytes (%108) monoprocessor,
  288 bytes (%120) multiprocessor.

UNIT 10

NUMCHANNEL

Allocation:

- 1 channel = 174 bytes (%AE).

NUMPROGRAM

Allocation:

- Program table = 1 pointer (4 bytes) for each program descriptor.
- Program descriptor = 302 bytes (%12E).
- Total: 306 bytes (%132)

UNIT 50

FILENUMBER

Allocation:

-	1 AD entry	= 64 bytes (%40)
-	1 ADD entry	= 86 bytes (%56)
-	1 sid	= 40 bytes (%28)
-	1 File Table Entry (for MULTFILES)	= 2 bytes (%2)
-	1 Multiple File Table Entry	= 26 bytes (%1A)
-	1 Secondary Index Table Entry	= 12 bytes (%C)

NUMSID

Allocation:

- 1 sid = 40 bytes (%28).

NUMSYSBUFF

Allocation:

- Buffer header = 38 bytes (%26).

NUMPRIVBPOOL

Allocation:

- Pool header = 26 bytes (%1A).

NUMPRIVBUFF

Allocation:

- Buffer header = 38 bytes (%26).

FILFORMEMVOL

Allocation:

- 1 PDD for each file = 100 bytes (%64).

UNIT 52

MULTPLFILES

Allocation:

- 1 File Table Entry (for MULTPLFILES) = 2 bytes (%2)
- 1 Multiple File Table Entry = 26 bytes (%1A)

PACKPOSFILES

Allocation:

- 1 descriptor for every positional file = 28 bytes (%1C).

KEYEDFILES

Allocation:

- 1 descriptor for every keyed file = 28 bytes (%1C).

LOCKEDREC

Allocation:

- 1 descriptor for every locked record = 24 bytes (%18).

LOCKEDPHAN

Allocation:

- 1 phantom record descriptor = 100 bytes (%64).

UNIT 53

SECONDARYIND

Allocation:

- 1 descriptor for every Secondary Index = 12 bytes (%C).

B. USER FILES TYPES AND CORRESPONDING INTERNAL FILES

Correspondences between the user defined file types and their internal structures are listed below. For further datails see the manual MOS Structure and Functioning.

TYPE OF USER FILE	CORRESPONDING INTERNAL FILES
Byte-stream	- data file
Positional no deletion	- data file - info file
Positional yes deletion	- data file - info file - bit map file
Positional packed	- data file - info file - numeric index file
Keyed	- data file - info file - primary index file - secondary index files

FILE CONNECTED	CORRESPONDING INTERNAL OPERATIONS
Byte-stream	assignment of: - 1 ADD Table Entry - 1 AD Table Entry - 1 sid (Connect-Id) for the data file
Positional no deletion	assignment of: - 1 ADD Table Entry - 1 AD Table Entry - 1 sid (Connect-ID) for data files and info files assignment of: 1 sid (Open-Id) for info file assignment of: 1 Multiple File Table Entry A connection to the same filename from another user causes the assignment of 1 sid (Connect-Id) for the data file (valid up to the relative Disconnect) and 1 sid (Connect-Id) for the info file (temporary). The last disconnect causes the release of everything previously assigned.
Positional yes deletion	assignment of: - 1 ADD Table Entry - 1 AD Table Entry - 1 sid (Connect-Id) for data files, info files and bitmap files assignment of: 1 sid (Open-Id) for info files and bitmap files assignment of: 1 Multiple File Table Entry A connection to the same filename by another user causes the assignment of 3 sid (Connect-Id) for the data file (valid up to the relative Disconnect), the bitmap file and the info file (temporary). The last Disconnect causes the release of everything assigned previously.

FILE CONNECTED	CORRESPONDING INTERNAL OPERATIONS
Positional packed	assignment of: - 1 ADD Table Entry - 1 AD Table Entry - 1 sid (Connect-Id) for data files, info files and numeric index files. assignment of: 1 sid (Open-Id) for info files and numeric index files. assignment of: 1 Multiple File Table Entry. The info file is always open. A connection to the same filename by another user causes the assignment of 3 sid (Connect-Id) for the data file (valid up to the relative Disconnect), the numeric indix file and the info file (temporary). The last Disconnect causes the release of everyting assigned previously.
Keyed	assignment of: - 1 ADD Table Entry - 1 AD Table Entry - 1 sid (Connect-Id) for data files, info files, primary index and secondary indices, if any. assignment of: 1 sid (Open-Id) for info files, primary index and every secondary index, if present. assignment of: 1 Multiple File Tabel Entry for every secondary index, if present. The info file is always open. A connection to the same filename by another user causes the assignment of 1 sid (Connect-Id) for - respectively - the data file (valid up to the relative Disconnect), primary index file, info file and index files (all temporary). The last Disconnect causes the release of everything assigned previously.

C. CONFIGURATION PARAMETERS EXAMPLES

Some examples of configuration parameters are given in this Appendix.

EXAMPLE NUMBER 1: STAND ALONE ST SYSTEM

This example refers to a stand-alone system. It is a ST system supporting:

- the BASIC application environment with the possibility to use graphics and VISA

- the preparation and execution of programs written in FORTRAN.

Information is now given on:

- hardware configuration

- software configuration, and in particular:

  . system configuration ($CON)

  . Grandpa configuration file ($CONFGP).

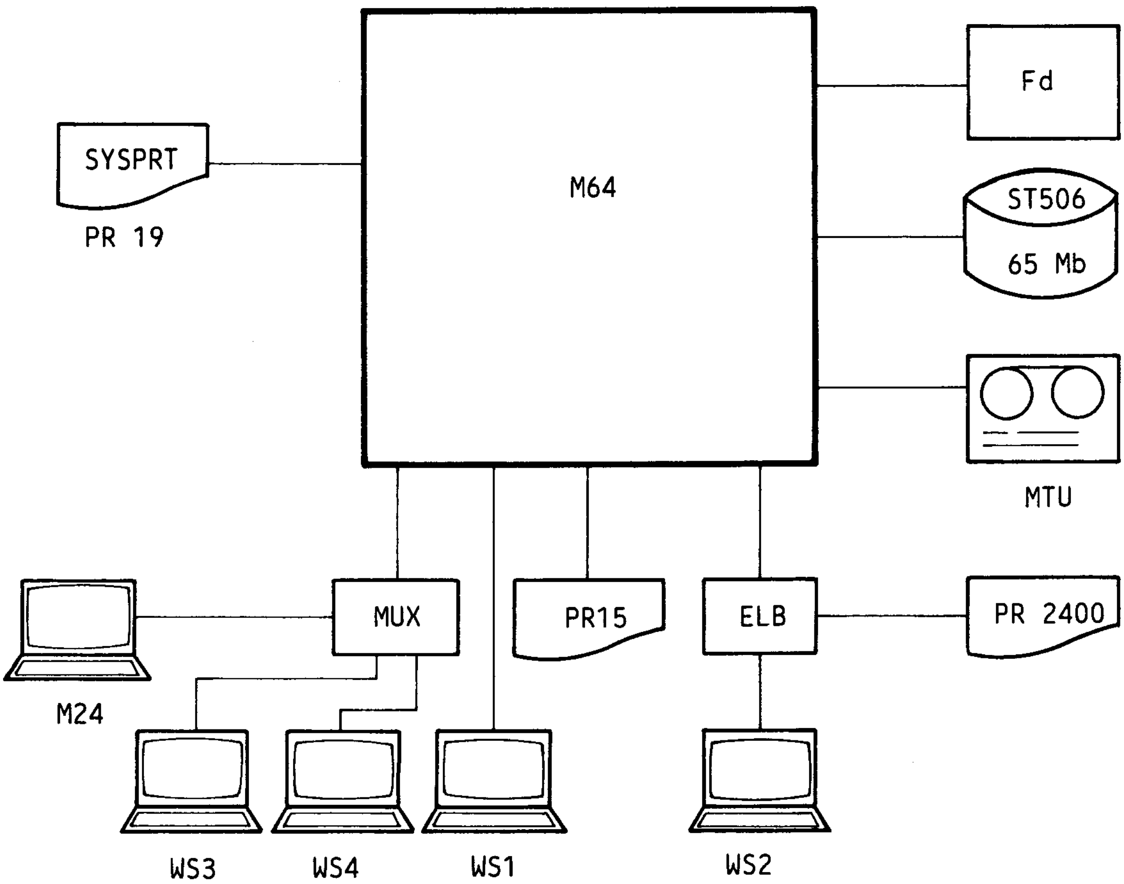


Fig. C-1 Stand-Alone ST System Hardware Configuration

STAND-ALONE $CON

UNIT 1

```
USRFAMILY       =  20
NUMPROCESS      =  40
MACHNAMEFLAG    =  1
NETWORKNAME     =  0
HOSTNAME        =  128
NUMDAMSEG       =  12
SEG_NUMBER      =  %0040
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0041
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0042
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0043
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0044
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0045
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0046
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0047
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0048
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0049
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004A
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004B
SEG_SIZE        =  %FFFE
NUMHWDSEG       =  8
SEGNUMBER       =  %0050
SEGNUMBER       =  %0051
SEGNUMBER       =  %0052
SEGNUMBER       =  %0053
SEGNUMBER       =  %0054
SEGNUMBER       =  %0055
SEGNUMBER       =  %0056
SEGNUMBER       =  %0057
```

UNIT 33

```
VDKMEMPAGES     =  256
VDKKEY          =  VDK0
```

UNIT 31

This unit has no parameters

UNIT 5

```
              =  %7773
WSNUMBER      =  5
WSID          =  1                    Work station 1 definition
SKREQUEST     =  1
OSKARCHKEY    =  EEA0
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  TAA0
PRREQUEST     =  1
PRT_CHKEY     =  TAA0
PRT_PRMODEF   =  %0001
PRT_HARDCP    =  1
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
BADGEREQUE    =  0
BADGECHKEY    =  EAA0
MICRREQUE     =  0
MICRCHKEY     =  EAA0
MICRMLTYPE    =  0
CADPREQUEST   =  0
CADPCHKEY     =  EAA0
CADPCATYPE    =  0
CADPSHARING   =  0
WSID          =  2                    Work station 2 definition
SKREQUEST     =  1
OSKARCHKEY    =  EEB0
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  EAB0
PRREQUEST     =  1
PRT_CHKEY     =  EAB0
PRT_PRMODEF   =  %0000
PRT_HARDCP    =  1
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
BADGEREQUE    =  0
BADGECHKEY    =  EAA0
MICRREQUE     =  0
MICRCHKEY     =  EAA0
MICRMLTYPE    =  0
CADPREQUEST   =  0
CADPCHKEY     =  EAA0
CADPCATYPE    =  0
CADPSHARING   =  0
WSID          =  3                    Work station 3 definition
SKREQUEST     =  1
OSKARCHKEY    =  MEA1
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  ****
PRREQUEST     =  0
PRT_CHKEY     =  EAA0
PRT_PRMODEF   =  %0000
PRT_HARDCP    =  0
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
```

```
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  4                  Work station 4 definition
SKREQUEST      =  1
OSKARCHKEY     =  MEA3
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  5                  Work station 5 definition (PC)
SKREQUEST      =  1
OSKARCHKEY     =  MTA0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
               =  %7370
SYSPRTNUMBE    =  1                  System printer definition
SYSPRTCONF     =  1
SYSPRTCHKEY    =  RAA0
SYSPRTMODE     =  %0001
```

```
                =  %7264
RSDRINSTNUM     =  1              RS232 driver definition
RSDRCHKEY       =  MAA3
RSDROFFCTL1     =  %0001
RSDROFFCTL2     =  %0000
                =  %636C
CHNUMBER        =  9              Channels definition
CHITEMCHKEY     =  EEA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F4
CHITEMCHKEY     =  TAA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0A0A
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  EEB0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F3
CHITEMCHKEY     =  EAB0
STOPBPAR        =  %0002
RXTXCHLK        =  %0303
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  64
INITXOFFLEV     =  96
```

```
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  MEA1
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  MEA3
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  MTA0
STOPBPAR        =  %0000
RXTXCHLK        =  %0303
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  2000
INITTXBUFF      =  2000
INITXOFFLEV     =  1600
INITXONLEV      =  600
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  RAA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0A0A
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  MAA3
```

```
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  2000
INITTXBUFF      =  2000
INITXOFFLEV     =  1600
INITXONLEV      =  600
KBTABLEID       =  %0000
WSTYPEID        =  %0000
                =  %6C6C
LNNUMBER        =  5
LNITEMCHKEY     =  EEA0
LNITEMRXTX      =  %0B0B
LNITEMCHKEY     =  EEB0
LNITEMRXTX      =  %0B0B
LNITEMCHKEY     =  MEA1
LNITEMRXTX      =  %0C0C
LNITEMCHKEY     =  MEA3
LNITEMRXTX      =  %0C0C
LNITEMCHKEY     =  MTA0
LNITEMRXTX      =  %0C0C
                =  %6E6D
NIMWSNUMB       =  %0000
                =  %6D33
M31M40LINEN     =  0
                =  %7462
TRANSCODTBN     =  1
include         =  ITALY_DP
                =  %7FFF
```

UNIT 14

```
FDUPRIORITY     =  1
FDUNUMOPERAT    =  42
FDUUNITSNUM     =  2
```

UNIT 15

```
HDCPRIORITY     =  1
HDCNUMOPERAT    =  42
HDCUNITSNUM     =  1
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
```

UNIT 24

This unit has no parameters

UNIT 54

```
TERMNUM        =  %0001
KEYCODE        =  %0042
```

UNIT 10

```
NUMCHANNEL     =  20
NUMPROGRAM     =  40
FILLER4        =  0
FILLER5        =  0
FILLER6        =  0
FILLER7        =  0
FILLER8        =  0
```

UNIT 50

```
FILENUMBER     =  100
NUMSID         =  250
NUMSYSBUFF     =  8
SYSBUFFSIZE    =  512
NUMPRIVBPOOL   =  3
NUMPRIVBUFF    =  5
FILFORMEMVOL   =  50
O_T_SIZE       =  50
LOGHD          =  1
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  15
PACKPOSFILES   =  2
KEYEDFILES     =  13
LOCKEDREC      =  5
LOCKEDPHAN     =  3
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 30

This unit has no parameters

GRANDPA CONFIGURATION FILE FOR THE STAND-ALONE ST SYSTEM

```
ALLT:
DATE!GMASTER!MCL=/IPL/SYS/$VSH!
SHUTDOWN=SHUTDOWN;
```

EXAMPLE NUMBER 2: LAN (LION 200 HDLC)

This example refers to two systems belonging to a LAN based on the LION 200 HDLC support. They are DP systems supporting a transactional application written in COBOL and using the MTS application environment.

Information is now given on:

- hardware configuration

- software configuration, and in particular:

  - system configuration ($CON of each system)

  - Grandpa configuration file ($CONFGP of each system).

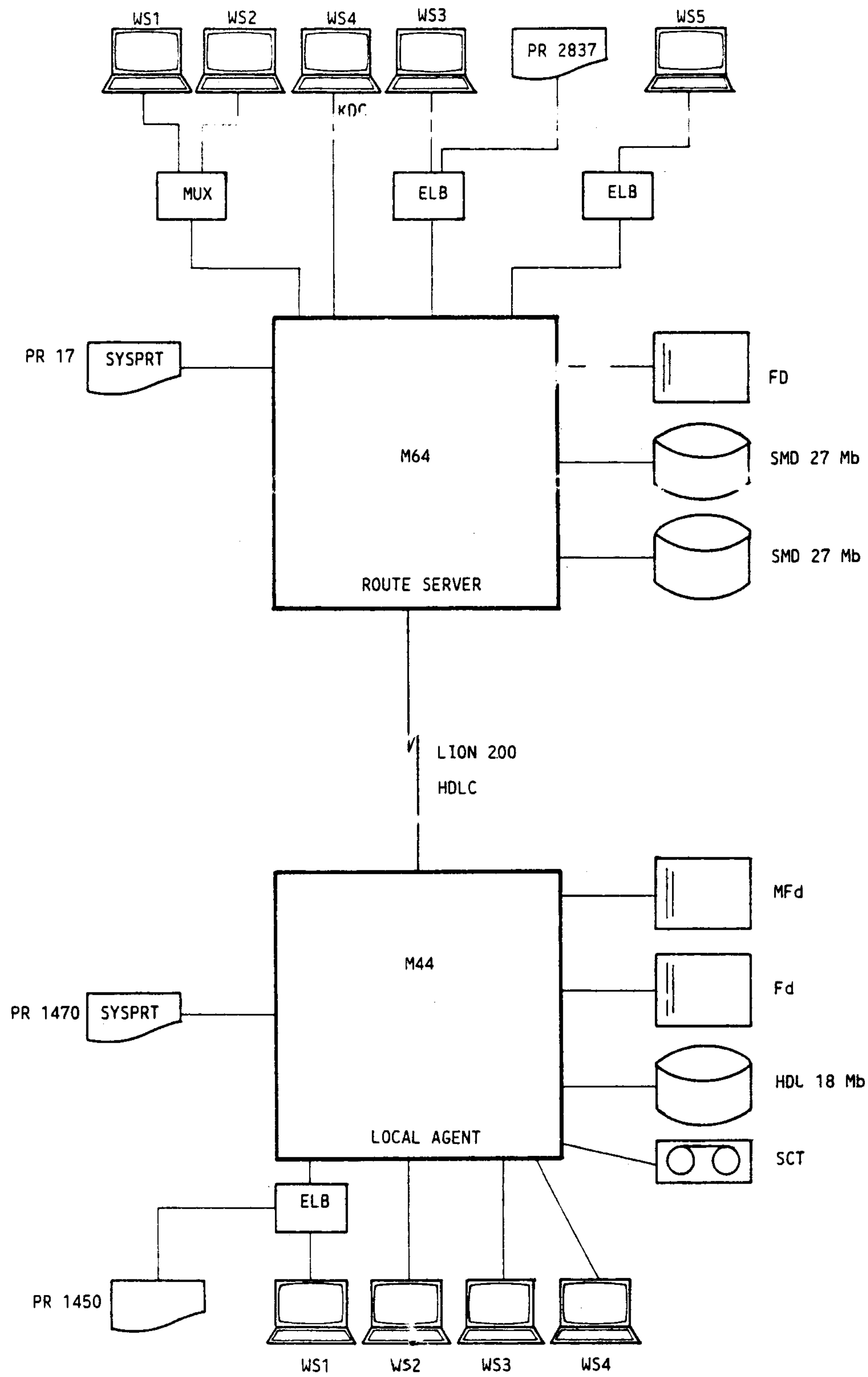


Fig. C-2 LAN Hardware Configuration

LAN ROUTE SERVER $CON

UNIT 1

```
USRFAMILY       =  40
NUMPROCESS      =  60
MACHNAMEFLAG    =  1
NETWORKNAME     =  0
HOSTNAME        =  128
NUMDAMSEG       =  12
SEG_NUMBER      =  %0040
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0041
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0042
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0043
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0044
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0045
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0046
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0047
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0048
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0049
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004A
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004B
SEG_SIZE        =  %FFFE
NUM_HWDSEG      =  8
SEGNUMBER       =  %0050
SEGNUMBER       =  %0051
SEGNUMBER       =  %0052
SEGNUMBER       =  %0053
SEGNUMBER       =  %0054
SEGNUMBER       =  %0055
SEGNUMBER       =  %0056
SEGNUMBER       =  %0057
```

UNIT 33

```
VDKMEMPAGES     =  256
VDKKEY          =  VDK0
```

UNIT 31

This unit has no parameters

UNIT 5

```
                    =  %7773
WSNUMBER            =  5
WSID                =  1                     Work station 1 definition
SKREQUEST           =  1
OSKARCHKEY          =  MEA1
WSTYPEOLTER         =  %0101
OSKARHCPRID         =  ****
PRREQUEST           =  0
PRT_CHKEY           =  EAA0
PRT_PRMODEF         =  %0000
PRTHARDCP           =  0
SHARING             =  %0001
PINPADREQUE         =  0
PINPADCHKEY         =  EAA0
BADGEREQUE          =  0
BADGECHKEY          =  EAA0
MICRREQUE           =  0
MICRCHKEY           =  EAA0
MICRMLTYPE          =  0
CADPREQUEST         =  0
CADPCHKEY           =  EAA0
CADPCATYPE          =  0
CADPSHARING         =  0
WSID                =  2                     Work station 2 definition
SKREQUEST           =  1
OSKARCHKEY          =  MEA3
WSTYPEOLTER         =  %0101
OSKARHCPRID         =  ****
PRREQUEST           =  0
PRT_CHKEY           =  EAA0
PRT_PRMODEF         =  %0000
PRTHARDCP           =  0
SHARING             =  %0001
PINPADREQUE         =  0
PINPADCHKEY         =  EAA0
BADGEREQUE          =  0
BADGECHKEY          =  EAA0
MICRREQUE           =  0
MICRCHKEY           =  EAA0
MICRMLTYPE          =  0
CADPREQUEST         =  0
CADPCHKEY           =  EAA0
CADPCATYPE          =  0
CADPSHARING         =  0
WSID                =  3                     Work station 3 definition
SKREQUEST           =  1
OSKARCHKEY          =  EEA0
WSTYPEOLTER         =  %0101
OSKARHCPRID         =  EAA0
PRREQUEST           =  1                     Work station printer
PRT_CHKEY           =  EAA0
PRT_PRMODEF         =  %0000
PRTHARDCP           =  1
SHARING             =  %0001
PINPADREQUE         =  0
PINPADCHKEY         =  EAA0
```

```
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  4                        Work station 4 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEB0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRTHARDCP       =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  5                        Work station 5 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEC0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRTHARDCP       =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
                =  %7370
SYSPRTNUMBE     =  1                        System printer definition
SYSPRTCONF      =  1
SYSPRTCHKEY     =  RAA0
SYSPRTMODE      =  %0001
```

```
                =  %7264
RSDRINSTNUM     =  0                       RS232 driver definition
                =  %636C
CHNUMBER        =  7                       Channels definition
CHITEMCHKEY     =  MEA1
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  MEA3
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0C0C
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
CHITEMCHKEY     =  EEA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EAA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0A0A
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
```

```
CHITEMCHKEY    =  EEB0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0B0B
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %FBF1
WSTYPEID       =  %00F1
CHITEMCHKEY    =  EEC0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0B0B
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %FBEA
WSTYPEID       =  %00F1
CHITEMCHKEY    =  RAA0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0A0A
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
               =  %6C6C
LNNUMBER       =  5                        Line definition
LNITEMCHKEY    =  MEA1
LNITEMRXTX     =  %0C0C
LNITEMCHKEY    =  MEA3
LNITEMRXTX     =  %0C0C
LNITEMCHKEY    =  EEA0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEB0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEC0
LNITEMRXTX     =  %0B0B
               =  %6E6D
NIMWSNUMB      =  %0000
               =  %6D33
M31M40LINEN    =  0
               =  %7462
```

```
TRANSCODTBN     =  3
include         =  ITALY_DP
include         =  USAASCI_DP
include         =  ITALY_DP
                =  %7FFF
```

UNIT 14

```
FDUPRIORITY     =  1
FDUNUMOPERAT    =  42
FDUUNITSNUM     =  2
```

UNIT 15

```
HDCPRIORITY     =  1
HDCNUMOPERAT    =  42
HDCUNITSNUM     =  1
HDCUNITSNUM     =  1
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
HDCUNITSNUM     =  0
```

UNIT 18

This unit has no parameters

UNIT 54

```
TERMNUM         =  %0000
KEYCODE         =  %0042
```

UNIT 10

```
NUMCHANNEL      =  40
NUMPROGRAM      =  60
FILLER4         =  0
FILLER5         =  0
FILLER6         =  0
FILLER7         =  0
FILLER8         =  0
```

UNIT 50

```
FILENUMBER      =  200
NUMSID          =  400
NUMSYSBUFF      =  8
SYSBUFFSIZE     =  512
NUMPRIVBPOOL    =  3
NUMPRIVBUFF     =  5
FILFORMEMVOL    =  100
O_T_SIZE        =  50
LOGHD           =  1
LOGHD           =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  2
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 32

```
NSERVERTYPE    =  2
GLOBNSNET      =  0
GLOBNSHOST     =  0
FSERVERINST    =  2
GSERVERINST    =  1
RFA_TIMEOUT    =  90
RESERVED       =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  15
PACKPOSFILES   =  2
KEYEDFILES     =  13
LOCKEDREC      =  5
LOCKEDPHAN     =  3
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 28

```
MAXUSRNUM      =  10
```

UNIT 66

This unit has no parameters

UNIT 38

```
LINES1         =  1
SLOT1          =  %0027
LINES2         =  0
SLOT2          =  %0028
NAME1          =  LIN1
ADDRESSES1     =  %8081
TXBNUM1        =  3
               =  0
RXBNUM1        =  6
UNITS1         =  2
RXBSIZE1       =  610
TXBSIZE1       =  610
RXSKIP1        =  26
```

```
TXSKIP1          =  26
OPTIONS1         =  %0100
T1_1             =  50
T2_1             =  0
T3_1             =  100
CHTYPE1          =  0
EAROM1           =  0
INT_CLOCK1       =  %0608
NRZSPEED1        =  %0111
WIRES1           =  %0001
CODE_BCC1        =  %0100
SYN_NUM1         =  %0703
VRCINFO_RTX1     =  %0010
TMODEM_RTRY1     =  %0007
```

UNIT 4

```
GLO_PORT_NUM     =  20
PRI_PORT_NUM     =  20
TX_PORT_NUM      =  20
USR_MSG_SIZ      =  540
LOC_BUF_NUM      =  20
DELAYECHO        =  50
NUM_MACH         =  3
NUM_DRIV         =  1
NUM_NACK         =  7
NUM_FSR          =  2
TIME_OUT         =  20000
Include file:    L_RS
```

UNIT 30

This unit has no parameters

LAN LOCAL AGENT $CON

UNIT 1

```
USRFAMILY      =  40
NUMPROCESS     =  60
MACHNAMEFLAG   =  1
NETWORKNAME    =  0
HOSTNAME       =  130
NUMDAMSEG      =  5
SEG_NUMBER     =  %0008
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %000C
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0015
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0016
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0017
SEG_SIZE       =  %FFFE
NUM_HWDSEG     =  2
SEGNUMBER      =  %0018
SEGNUMBER      =  %0019
```

UNIT 33

```
VDKMEMPAGES    =  256
VDKKEY         =  VDK0
```

UNIT 31

This unit has no parameters

UNIT 5

```
               =  %7773
WSNUMBER       =  4
WSID           =  1                    Work station 1 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEA0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  EAA0
PRREQUEST      =  1
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  1
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
```

```
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  2                        Work station 2 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEB0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  3                        Work station 3 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEC0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  4                        Work station 4 definition
SKREQUEST       =  1
OSKARCHKEY      =  EED0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
```

```
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
               =  %7370
SYSPRTNUMBE    =  1                         System printer definition
SYSPRTCONF     =  1
SYSPRTCHKEY    =  RAA0
SYSPRTMODE     =  %0000
               =  %7264
RSDRINSTNUM    =  0                         RS232 driver definition
               =  %636C
CHNUMBER       =  6                         Channels definition
CHITEMCHKEY    =  EEA0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0B0B
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %FBE3
WSTYPEID       =  %00F1
CHITEMCHKEY    =  EAA0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0A0A
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  EEB0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0B0B
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  256
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %FBEA
```

```
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEC0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBF1
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EED0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  RAA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  256
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %0000
WSTYPEID        =  %0000
                =  %6C6C
LNNUMBER        =  4                       Line number
LNITEMCHKEY     =  EEA0
LNITEMRXTX      =  %0B0B
LNITEMCHKEY     =  EEB0
LNITEMRXTX      =  %0B0B
LNITEMCHKEY     =  EEC0
LNITEMRXTX      =  %0B0B
LNITEMCHKEY     =  EED0
LNITEMRXTX      =  %0B0B
                =  %6E6D
NIMWSNUMB       =  %0000
                =  %6D33
M31M40LINEN     =  0
                =  %7462
TRANSCODTBN     =  4
```

```
include          =  GERMANY_DP
include          =  ITALY_DP
include          =  USAASCI_DP
include          =  ITALY_DP
                 =  %7FFF
```

UNIT 14

```
FDUPRIORITY      =  1
FDUNUMOPERAT     =  62
FDUUNITSNUM      =  2
```

UNIT 13

```
MFDPRIORITY      =  1
MFDNUMOPERAT     =  62
MFDUNITSNUM      =  1
```

UNIT 21

```
HDUPRIORITY      =  1
HDUNUMOPERAT     =  62
HDUUNITSNUM      =  1
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
HDUUNITSNUM      =  0
```

UNIT 18

This unit has no parameters

UNIT 54

```
TERMNUM          =  %0000
KEYCODE          =  %0042
```

UNIT 10

```
NUMCHANNEL       =  40
NUMPROGRAM       =  60
FILLER4          =  0
FILLER5          =  0
FILLER6          =  0
FILLER7          =  0
FILLER8          =  0
```

UNIT 50

```
FILENUMBER       =  200
NUMSID           =  400
NUMSYSBUFF       =  8
SYSBUFFSIZE      =  512
```

```
NUMPRIVBPOOL    =  3
NUMPRIVBUFF     =  5
FILFORMEMVOL    =  90
O_T_SIZE        =  50
LOGHD           =  1
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 32

```
NSERVERTYPE    =  0
GLOBNSNET      =  0
GLOBNSHOST     =  0
FSERVERINST    =  2
GSERVERINST    =  2
RFA_TIMEOUT    =  90
RESERVED       =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  15
PACKPOSFILES   =  2
KEYEDFILES     =  13
LOCKEDREC      =  5
LOCKEDPHAN     =  3
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 28

```
MAXUSRNUM      =  10
```

UNIT 66

This unit has no parameters

UNIT 38

```
LINES1         =  1
SLOT1          =  %0027
LINES2         =  0
SLOT2          =  %0028
NAME1          =  LIN1
ADDRESSES1     =  %8082
TXBNUM1        =  2
```

```
                  =  0
RXBNUM1           =  4
UNITS1            =  1
RXBSIZE1          =  610
TXBSIZE1          =  610
RXSKIP1           =  26
TXSKIP1           =  26
OPTIONS1          =  %0000
T1_1              =  5000
T2_1              =  0
T3_1              =  0
CHTYPE1           =  0
EAROM1            =  0
INT_CLOCK1        =  %0608
NRZSPEED1         =  %0111
WIRES1            =  %0002
CODE_BCC1         =  %0100
SYN_NUM1          =  %0703
VRCINFO_RTX1      =  %0010
TMODEM_RTRY1      =  %0007
```

UNIT 4

```
GLO_PORT_NUM      =  20
PRI_PORT_NUM      =  20
TX_PORT_NUM       =  20
USR_MSG_SIZ       =  540
LOC_BUF_NUM       =  20
DELAYECHO         =  50
NUM_MACH          =  3
NUM_DRIV          =  1
NUM_NACK          =  7
NUM_FSR           =  2
TIME_OUT          =  20000
Include file:  STANDALONE
```

UNIT 30

This unit has no parameters

GRANDPA CONFIGURATION FILE FOR THE LAN ROUTE SERVER

```
PFG:/IPL/SYS/$SERVER;
PCALL:/IPL/DPC/MTS/MTSCTLG,<DBENV>SERVER<>f;
PCALL:/IPL/DPC/MTS/MSWMAN,<>f;
START:/IPL/DPC/MTS/MSWDIS;
FG:MTS=/IPL/DPC/MTS/GMAN,<DMTS>MTSA;
ALLT:
CONF=/IPL/DPC/MTS/CMD/CONFMTS!
MTSLOG=/IPL/DPC/MTS/SMAN,<ERMS>IPL/DPC/MTS/GMAN/ERRMSG
<OVLI>IPL/DPC/MTS/OVLSMAN<DMTS>MTSA!
JOU=/IPL/DPC/MTS/CMD/DISPJOU!
DATE!GMASTER!MCL=/IPL/SYS/$VSH!
SHUTDOWN=SHUTDOWN;
```

GRANDPA CONFIGURATION FILE FOR THE LAN LOCAL AGENT

```
SERV:/IPL/SYS/$SERVER;
CALL:/IPL/DPC/MTS/MTSCTLG,<>f;
CALL:/IPL/DB/CONC_HANDL,<DBENV>CLIENT;
CALL:/IPL/DPC/MTS/MSWMAN,<>f;
START:/IPL/DPC/MTS/MSWDIS;
ALLT:
MTSLOG=/IPL/DPC/MTS/SMAN,<ERMS>IPL/DPC/MTS/ERRMSG
<OVLI>IPL/DPC/MTS/OVLSMAN<DMTS>MTSA!
DATE!GMASTER!MCL=/IPL/SYS/$VSH!
SHUTDOWN;
```

EXAMPLE NUMBER 3: LAN (ETHERNET)

This example refers to three systems interconnected in a LAN configuration based on Ethernet support.

Some information about the application program is also given.

The application program in question belongs to the business area (BC).

There are three types of transactions in this program:

- Inventory
- Orders and Invoicing
- Accounting

The program is written in COBOL and VISA, and is handled by BEAM.

The application data base consists of about 40 keyed files.

Information is now given on:

- hardware configuration
- software configuration, and in particular:
  - system configuration ($CON of each system)
  - Grandpa configuration file ($CONFGP of each system)

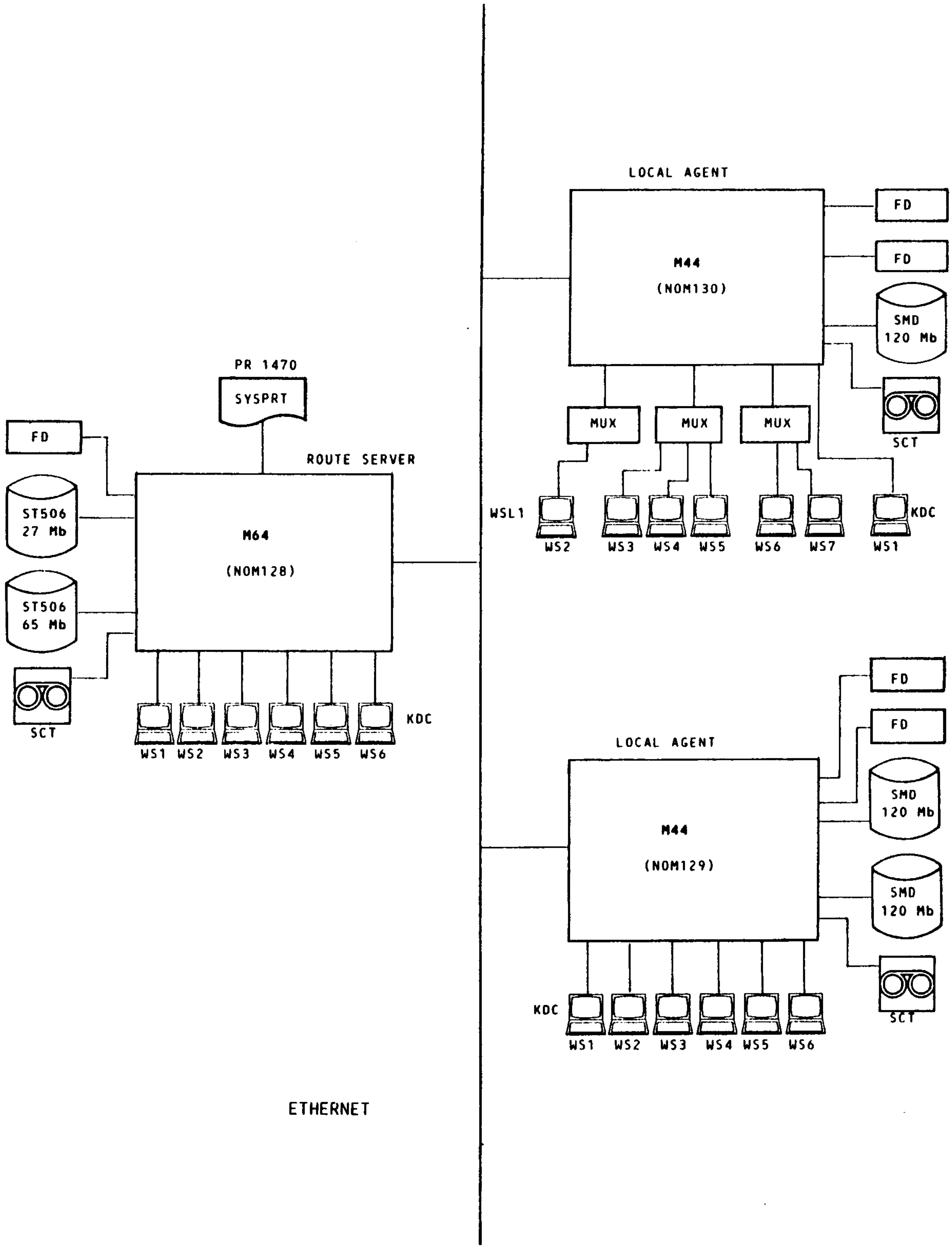


Fig. C-3 Ethernet LAN Hardware Configuration

Considerations on the Configuration Parameters Values on Route Server

The aim of this configuration is to provide a good system performance. Some of the default values provided for the parameters have, therefore, been modified. Those parameters which do not alter the system's performance have not been changed.

The following parameters have been tuned using the NOSE utility (see the manual System Software Maintenance, User Guide):

- USRFAMILY
- NUMPROCESS
- NUMCHANNEL
- NUMPROGRAM

The default values of the following parameters have been slightly increased to improve execution performance:

- GLO_PORT_NUM
- PRI_PORT_NUM
- TX_PORT_NUM

The value of the VDKMEMPAGES has been increased because of a specific need (some tables are loaded in memory by a procedure activated in the Grandpa configuration file).

The value of the FSERVERINST parameter, representing the maximum parallelism allowed for remote terminals to access files on this machine, has been left at 5: no improvement is noted if the value is changed from 5 to 7.

ROUTE SERVER $CON

UNIT 1

```
USRFAMILY       =  40
NUMPROCESS      =  50
MACHNAMEFLAG    =  1
NETWORKNAME     =  0
HOSTNAME        =  128
NUMDAMSEG       =  12
SEG_NUMBER      =  %0040
SEG_SIZE        =  %FFFE
SEG_NUMER       =  %0041
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0042
SEG_SIZE        =  %FFFE
SEG_NUMBR       =  %0043
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0044
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0045
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0046
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0047
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0048
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %0049
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004A
SEG_SIZE        =  %FFFE
SEG_NUMBER      =  %004B
SEG_SIZE        =  %FFFE
NUMHWDSEG       =  8
SEGNUMBER       =  %0050
SEGNUMBER       =  %0051
SEGNUMBER       =  %0052
SEGNUMBER       =  %0053
SEGNUMBER       =  %0054
SEGNUMBER       =  %0055
SEGNUMBER       =  %0056
SEGNUMBER       =  %0057
```

UNIT 33

```
VDKMEMPAGES     =  572
VDKKEY          =  VKDO
```

UNIT 31

This unit has no parameters.

UNIT 5

```
                =  %7773
```

```
WSNUMBER      =  6
WSID          =  1                          Work station 1 definition
SKREQUEST     =  1
OSKARCHKEY    =  EEA0
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  ****
PRREQUEST     =  0
PRT_CHKEY     =  EAA0
PRT_PRMODEF   =  %0000
PRT_HARDCP    =  0
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
BADGEREQUE    =  0
BADGECHKEY    =  EAA0
MICRREQUE     =  0
MICRCHKEY     =  EAA0
MICRMLTYPE    =  0
CADPREQUEST   =  0
CADPCHKEY     =  EAA0
CADPCATYPE    =  0
CADPSHARING   =  0
WSID          =  2                          Work station 2 definition
SKREQUEST     =  1
OSKARCHKEY    =  EEB0
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  ****
PRREQUEST     =  0
PRT_CHKEY     =  EAA0
PRT_PRMODEF   =  %0000
PRT_HARDCP    =  0
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
BADGEREQUE    =  0
BADGECHKEY    =  EAA0
MICRREQUE     =  0
MICRCHKEY     =  EAA0
MICRMLTYPE    =  0
CADPREQUEST   =  0
CADPCHKEY     =  EAA0
CADPCATYPE    =  0
CADPSHARING   =  0
WSID          =  3                          Work station 3 definition
SKREQUEST     =  1
OSKARCHKEY    =  EEC0
WSTYPEOLTER   =  %0101
OSKARHCPRID   =  ****
PRREQUEST     =  0
PRT_CHKEY     =  EAA0
PRT_PRMODEF   =  %0000
PRT_HARDCP    =  0
SHARING       =  %0001
PINPADREQUE   =  0
PINPADCHKEY   =  EAA0
BADGEREQUE    =  0
```

```
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  4                    Work station 4 definition
SKREQUEST      =  1
OSKARCHKEY     =  EED0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  EAA0
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  5                    Work station 5 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEE0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  6                    Work station 6 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEF0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
```

```
PRT_CHKEY     = EAA0
PRT_PRMODEF   = %0000
PRTHARDCP     = 0
SHARING       = %0001
PINPADREQUE   = 0
PINPADCHKEY   = EAA0
BADGEREQUE    = 0
BADGECHKEY    = EAA0
MICRREQUE     = 0
MICRCHKEY     = EAA0
MICRMLTYPE    = 0
CADPREQUEST   = 0
CADPCHKEY     = EAA0
CADPCATYPE    = 0
CADPSHARING   = 0
              = %7370
SYSPRTNUMBE   = 1              System printer definition
SYSPRTCONF    = 1
SYSPRTCHKEY   = RAA0
SYSPRTMODE    = %0001
              = %7264
RSDRINSTNUM   = 0              RS232 driver definition
              = %636C
CHNUMBER      = 7              Channels definition
CHITEMCHKEY   = EEA0
STOPBPAR      = %0002
RXTXCHLK      = %0202
PGMRXTXBL     = %0B0B
TANDEMCTLMO   = %0101
OUTSTATBRPE   = %0000
OUTSTDTRRTS   = %0101
INITRXBUFF    = 128
INITTXBUFF    = 128
INITXOFFLEV   = 96
INITXONLEV    = 64
KBTABLEID     = %FBEA
WSTYPEID      = %00F1
CHITEMCHKEY   = EEB0
STOPBPAR      = %0002
RXTXCHLK      = %0202
PGMRXTXBL     = %0B0B
TANDEMCTLMO   = %0101
OUTSTATBRPE   = %0000
OUTSTDTRRTS   = %0101
INITRXBUFF    = 128
INITTXBUFF    = 128
INITXOFFLEV   = 96
INITXONLEV    = 64
KBTABLEID     = %FBEA
WSTYPEID      = %00F1
CHITEMCHKEY   = EEC0
STOPBPAR      = %0002
RXTXCHLK      = %0202
PGMRXTXBL     = %0B0B
TANDEMCTLMO   = %0101
OUTSTATBRPE   = %0000
```

```
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EED0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEE0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEF0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBF1
WSTYPEID        =  %00F1
CHITEMCHKEY     =  RAA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
```

```
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
               =  %6C6C
LNNUMBER       =  6                                Line definition
LNITEMCHKEY    =  EEA0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEB0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEC0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EED0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEE0
LNITEMRXTX     =  %0B0B
LNITEMCHKEY    =  EEF0
LNITEMRXTX     =  %0B0B
               =  %6E6D
NIMWSNUMB      =  %0000
               =  %6D33
M31M40LINEN    =  0
               =  %7462
TRANSCODTBN    =  2
include        =  ITALY_DP
include        =  USAASCI_DP
               =  %7FFF
```

UNIT 14

```
FDUPRIORITY    =  1
FDUNUMOPERAT   =  52
FDUUNITSNUM    =  2
```

UNIT 15

```
HDCPRIORITY    =  1
HDCNUMOPERAT   =  52
HDCUNITSNUM    =  1
HDCUNITSNUM    =  1
HDCUNITSNUM    =  0
HDCUNITSNUM    =  0
HDCUNITSNUM    =  0
HDCUNITSNUM    =  0
HDCUNITSNUM    =  0
HDCUNITSNUM    =  0
```

UNIT 18

This unit has no parameters

UNIT 54

```
TERMNUM        =  %0000
KEYCODE        =  %0006
```

UNIT 10

```
NUMCHANNEL     =  40
NUMPROGRAM     =  60
FILLER4        =  0
FILLER5        =  0
FILLER6        =  0
FILLER7        =  0
FILLER8        =  0
```

UNIT 50

```
FILENUMBER     =  400
NUMSID         =  600
NUMSYSBUFF     =  24
SYSBUFFSIZE    =  512
NUMPRIVBPOOL   =  8
NUMPRIVBUFF    =  16
FILFORMEMVOL   =  70
O_T_SIZE       =  70
LOGHD          =  1
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  2
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 32

```
NSERVERTYPE    =  2
GLOBNSNET      =  0
GLOBNSHOST     =  0
FSERVERINST    =  5
GSERVERINST    =  1
RFA_TIMEOUT    =  800
RESERVED       =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  80
PACKPOSFILES   =  20
KEYEDFILES     =  60
LOCKEDREC      =  20
LOCKEDPHAN     =  10
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 28

```
MAXUSRNUM      =  10
```

UNIT 66

This unit has no parameters

UNIT 27

```
BOARDS_NUM     =  1
BOARD_ID       =  %006F
DRIVER_ID      =  ETHR
PHYS_ADD1      =  %0000
PHYS_ADD2      =  %0000
PHYS_ADD3      =  %0000
TYPE_FLAG      =  1
PROM_FLAG      =  0
RX_SIZE        =  622
RX_SKIP        =  28
TX_SIZE        =  622
```

UNIT 4

```
GLO_PORT_NUM   =  22
PRI_PORT_NUM   =  22
TX_PORT_NUM    =  22
USR_MSG_SIZ    =  540
LOC_BUF_NUM    =  20
DELAYECHO      =  50
NUM_MACH       =  3
NUM_DRIV       =  1
NUM_NACK       =  7
NUM_FSR        =  2
TIME_OUT       =  20000
Include file: E_RS
```

LOCAL AGENT (NOM129) $CON

UNIT 1

```
USRFAMILY      =  40
NUMPROCESS     =  50
MACHNAMEFLAG   =  1
NETWORKNAME    =  0
HOSTNAME       =  129
NUMDAMSEG      =  5
SEG_NUMBER     =  %0008
SEG_SIZE       =  %FFFE
SEG_NUMER      =  %000C
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0015
SEG_SIZE       =  %FFFE
SEG_NUMBR      =  %0016
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0017
SEG_SIZE       =  %FFFE
NUM_HWDSEG     =  2
SEGNUMBER      =  %0018
SEGNUMBER      =  %0019
```

UNIT 33

```
VDKMEMPAGES    =  572
VDKKEY         =  VKDO
```

UNIT 31

This unit has no parameters.

UNIT 5

```
               =  %7773
WSNUMBER       =  6
WSID           =  1                            Work station 1 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEAO
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAAO
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAAO
BADGEREQUE     =  0
BADGECHKEY     =  EAAO
MICRREQUE      =  0
MICRCHKEY      =  EAAO
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAAO
```

```
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  2                          Work station 2 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEB0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  3                          Work station 3 definition
SKREQUEST       =  1
OSKARCHKEY      =  EEC0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  4                          Work station 4 definition
SKREQUEST       =  1
OSKARCHKEY      =  EED0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  EAA0
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRT_HARDCP      =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
```

```
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  5                         Work station 5 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEE0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRTHARDCP      =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  6                         Work station 6 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEF0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRTHARDCP      =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
               =  %7370
SYSPRTNUMBE    =  0                         System printer definition
               =  %7264
RSDRINSTNUM    =  0                         RS232 driver definition
               =  %636C
```

```
CHNUMBER        =  6                       Channels definition
CHITEMCHKEY     =  EEA0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEB0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEC0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EED0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
PGMRXTXBL       =  %0B0B
TANDEMCTLMO     =  %0101
OUTSTATBRPE     =  %0000
OUTSTDTRRTS     =  %0101
INITRXBUFF      =  128
INITTXBUFF      =  128
INITXOFFLEV     =  96
INITXONLEV      =  64
KBTABLEID       =  %FBEA
WSTYPEID        =  %00F1
CHITEMCHKEY     =  EEE0
STOPBPAR        =  %0002
RXTXCHLK        =  %0202
```

```
PGMRXTXBL      = %0B0B
TANDEMCTLMO    = %0101
OUTSTATBRPE    = %0000
OUTSTDTRRTS    = %0101
INITRXBUFF     = 128
INITTXBUFF     = 128
INITXOFFLEV    = 96
INITXONLEV     = 64
KBTABLEID      = %FBEA
WSTYPEID       = %00F1
CHITEMCHKEY    = EEF0
STOPBPAR       = %0002
RXTXCHLK       = %0202
PGMRXTXBL      = %0B0B
TANDEMCTLMO    = %0101
OUTSTATBRPE    = %0000
OUTSTDTRRTS    = %0101
INITRXBUFF     = 128
INITTXBUFF     = 128
INITXOFFLEV    = 96
INITXONLEV     = 64
KBTABLEID      = %FBF1
WSTYPEID       = %00F1
               = %6C6C
LNNUMBER       = 6                          Line definition
LNITEMCHKEY    = EEA0
LNITEMRXTX     = %0B0B
LNITEMCHKEY    = EEB0
LNITEMRXTX     = %0B0B
LNITEMCHKEY    = EEC0
LNITEMRXTX     = %0B0B
LNITEMCHKEY    = EED0
LNITEMRXTX     = %0B0B
LNITEMCHKEY    = EEE0
LNITEMRXTX     = %0B0B
LNITEMCHKEY    = EEF0
LNITEMRXTX     = %0B0B
               = %6E6D
NIMWSNUMB      = %0000
               = %6D33
M31M40LINEN    = 0
               = %7462
TRANSCODTBN    = 2
include        = ITALY_DP
include        = USAASCI_DP
               = %7FFF
```

UNIT 14

```
FDUPRIORITY    = 1
FDUNUMOPERAT   = 52
FDUUNITSNUM    = 2
```

UNIT 20

```
SMDPRIORITY    = 1
```

```
SMDNUMOPERAT    =  52
SMDUNITSNUM     =  1
SMDUNITSNUM     =  1
SMDUNITSNUM     =  0
SMDUNITSNUM     =  0
SMDUNITSNUM     =  0
SMDUNITSNUM     =  0
SMDUNITSNUM     =  0
SMDUNITSNUM     =  0
```

UNIT 18

This unit has no parameters

UNIT 54

```
TERMNUM         =  %0000
KEYCODE         =  %0006
```

UNIT 10

```
NUMCHANNEL      =  40
NUMPROGRAM      =  50
FILLER4         =  0
FILLER5         =  0
FILLER6         =  0
FILLER7         =  0
FILLER8         =  0
```

UNIT 50

```
FILENUMBER      =  300
NUMSID          =  400
NUMSYSBUFF      =  24
SYSBUFFSIZE     =  512
NUMPRIVBPOOL    =  3
NUMPRIVBUFF     =  5
FILFORMEMVOL    =  60
O_T_SIZE        =  60
LOGHD           =  1
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
```

```
LOGHD          =  3
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 32

```
NSERVERTYPE    =  0
GLOBNSNET      =  0
GLOBNSHOST     =  0
FSERVERINST    =  5
GSERVERINST    =  1
RFA_TIMEOUT    =  800
```

```
RESERVED       =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  100
PACKPOSFILES   =  25
KEYEDFILES     =  75
LOCKEDREC      =  10
LOCKEDPHAN     =  10
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 28

```
MAXUSRNUM      =  10
```

UNIT 66

This unit has no parameters

UNIT 27

```
BOARDS_NUM     =  1
BOARD_ID       =  %006F
DRIVER_ID      =  ETHR
PHYS_ADD1      =  %0000
PHYS_ADD2      =  %0000
PHYS_ADD3      =  %0000
TYPE_FLAG      =  1
PROM_FLAG      =  0
RX_SIZE        =  622
RX_SKIP        =  28
TX_SIZE        =  622
```

UNIT 4

```
GLO_PORT_NUM   =  22
PRI_PORT_NUM   =  22
TX_PORT_NUM    =  30
USR_MSG_SIZ    =  540
LOC_BUF_NUM    =  30
DELAYECHO      =  50
NUM_MACH       =  3
NUM_DRIV       =  1
NUM_NACK       =  7
NUM_FSR        =  2
TIME_OUT       =  20000
Include file: E_LRA
```

LOCAL AGENT (NOM130) $CON

UNIT 1

```
USRFAMILY      =  40
NUMPROCESS     =  50
MACHNAMEFLAG   =  1
NETWORKNAME    =  0
HOSTNAME       =  130
NUMDAMSEG      =  4
SEG_NUMBER     =  %0008
SEG_SIZE       =  %FFFE
SEG_NUMER      =  %000C
SEG_SIZE       =  %FFFE
SEG_NUMBER     =  %0015
SEG_SIZE       =  %FFFE
SEG_NUMBR      =  %0016
SEG_SIZE       =  %FFFE
NUM_HWDSEG     =  3
SEGNUMBER      =  %0017
SEGNUMBER      =  %0018
SEGNUMBER      =  %0019
```

UNIT 33

```
VDKMEMPAGES    =  572
VDKKEY         =  VKD0
```

UNIT 31

This unit has no parameters.

UNIT 5

```
               =  %7773
WSNUMBER       =  7
WSID           =  1                    Work station 1 definition
SKREQUEST      =  1
OSKARCHKEY     =  EEA0
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
```

```
CADPSHARING     =  0
WSID            =  2                    Work station 2 definition
SKREQUEST       =  1
OSKARCHKEY      =  MEA0
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRTHARDCP       =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  3                    Work station 3 definition
SKREQUEST       =  1
OSKARCHKEY      =  MEB1
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  ****
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRTHARDCP       =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
BADGECHKEY      =  EAA0
MICRREQUE       =  0
MICRCHKEY       =  EAA0
MICRMLTYPE      =  0
CADPREQUEST     =  0
CADPCHKEY       =  EAA0
CADPCATYPE      =  0
CADPSHARING     =  0
WSID            =  4                    Work station 4 definition
SKREQUEST       =  1
OSKARCHKEY      =  MEB2
WSTYPEOLTER     =  %0101
OSKARHCPRID     =  EAA0
PRREQUEST       =  0
PRT_CHKEY       =  EAA0
PRT_PRMODEF     =  %0000
PRTHARDCP       =  0
SHARING         =  %0001
PINPADREQUE     =  0
PINPADCHKEY     =  EAA0
BADGEREQUE      =  0
```

```
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  5                    Work station 5 definition
SKREQUEST      =  1
OSKARCHKEY     =  MEB3
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  6                    Work station 6 definition
SKREQUEST      =  1
OSKARCHKEY     =  MEC1
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRT_HARDCP     =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
WSID           =  7                    Work station 7 definition
SKREQUEST      =  1
OSKARCHKEY     =  MEC3
WSTYPEOLTER    =  %0101
OSKARHCPRID    =  ****
PRREQUEST      =  0
```

```
PRT_CHKEY      =  EAA0
PRT_PRMODEF    =  %0000
PRTHARDCP      =  0
SHARING        =  %0001
PINPADREQUE    =  0
PINPADCHKEY    =  EAA0
BADGEREQUE     =  0
BADGECHKEY     =  EAA0
MICRREQUE      =  0
MICRCHKEY      =  EAA0
MICRMLTYPE     =  0
CADPREQUEST    =  0
CADPCHKEY      =  EAA0
CADPCATYPE     =  0
CADPSHARING    =  0
               =  %7370
SYSPRTNUMBE    =  0                  System printer definition
               =  %7264
RSDRINSTNUM    =  0                  RS232 driver definition
               =  %636C
CHNUMBER       =  7                  Channels definition
CHITEMCHKEY    =  EEA0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0B0B
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %FBEA
WSTYPEID       =  %00F1
CHITEMCHKEY    =  MEA0
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  MEB1
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
```

```
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  MEB2
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  MEB3
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  MEC1
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
CHITEMCHKEY    =  MEC3
STOPBPAR       =  %0002
RXTXCHLK       =  %0202
PGMRXTXBL      =  %0C0C
TANDEMCTLMO    =  %0101
OUTSTATBRPE    =  %0000
OUTSTDTRRTS    =  %0101
INITRXBUFF     =  128
INITTXBUFF     =  128
INITXOFFLEV    =  96
INITXONLEV     =  64
KBTABLEID      =  %0000
WSTYPEID       =  %0000
```

```
                  =  %6C6C
LNNUMBER          =  7                      Line definition
LNITEMCHKEY       =  EEA0
LNITEMRXTX        =  %0B0B
LNITEMCHKEY       =  MEA0
LNITEMRXTX        =  %0C0C
LNITEMCHKEY       =  MEB1
LNITEMRXTX        =  %0C0C
LNITEMCHKEY       =  MEB2
LNITEMRXTX        =  %0C0C
LNITEMCHKEY       =  MEB3
LNITEMRXTX        =  %0C0C
LNITEMCHKEY       =  MEC1
LNITEMRXTX        =  %0C0C
LNITEMCHKEY       =  MEC3
LNITEMRXTX        =  %0C0C
                  =  %6E6D
NIMWSNUMB         =  %0000
                  =  %6D33
M31M40LINEN       =  0
                  =  %7462
TRANSCODTBN       =  2
include           =  ITALY_DP
include           =  USAASCI_DP
                  =  %7FFF
```

UNIT 14

```
FDUPRIORITY       =  1
FDUNUMOPERAT      =  52
FDUUNITSNUM       =  2
```

UNIT 20

```
SMDPRIORITY       =  1
SMDNUMOPERAT      =  52
SMDUNITSNUM       =  1
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
SMDUNITSNUM       =  0
```

UNIT 18

This unit has no parameters

UNIT 54

```
TERMNUM           =  %0000
KEYCODE           =  %0006
```

UNIT 10

```
NUMCHANNEL      =  40
NUMPROGRAM      =  50
FILLER4         =  0
FILLER5         =  0
FILLER6         =  0
FILLER7         =  0
FILLER8         =  0
```

UNIT 50

```
FILENUMBER      =  300
NUMSID          =  400
NUMSYSBUFF      =  24
SYSBUFFSIZE     =  512
NUMPRIVBPOOL    =  3
NUMPRIVBUFF     =  5
FILFORMEMVOL    =  60
O_T_SIZE        =  60
LOGHD           =  1
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
LOGHD           =  0
```

```
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
LOGHD          =  0
```

UNIT 32

```
NSERVERTYPE    =  0
GLOBNSNET      =  0
GLOBNSHOST     =  0
FSERVERINST    =  5
GSERVERINST    =  1
RFA_TIMEOUT    =  800
RESERVED       =  0
```

UNIT 51

This unit has no parameters

UNIT 52

```
MULTIPLFILES   =  100
PACKPOSFILES   =  25
KEYEDFILES     =  75
LOCKEDREC      =  10
LOCKEDPHAN     =  10
```

UNIT 53

```
SECONDARYIND   =  40
```

UNIT 28

```
MAXUSRNUM      =  10
```

UNIT 66

This unit has no parameters

UNIT 27

```
BOARDS_NUM     =  1
BOARD_ID       =  %006F
DRIVER_ID      =  ETHR
PHYS_ADD1      =  %0000
PHYS_ADD2      =  %0000
PHYS_ADD3      =  %0000
TYPE_FLAG      =  1
PROM_FLAG      =  0
RX_SIZE        =  622
RX_SKIP        =  28
TX_SIZE        =  622
```

UNIT 4

```
GLO_PORT_NUM   =  22
PRI_PORT_NUM   =  22
TX_PORT_NUM    =  30
USR_MSG_SIZ    =  540
LOC_BUF_NUM    =  30
DELAYECHO      =  50
NUM_MACH       =  3
NUM_DRIV       =  1
NUM_NACK       =  7
NUM_FSR        =  2
TIME_OUT       =  20000
Include file: E_LRA
```

GRANDPA CONFIGURATION FILE FOR THE ROUTE SERVER

```
INIT:MCL=/IPL/SYS/$VSH,<$user>ROOT<$cmd>IPL/END/CMD/TLOAD;
TTYA:GMASTER!DATE;
TTYA:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN!SHUTDOWN=SHUTDOWN;
TTYB:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYC:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYD:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYE:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYF:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
PFG:RFA=/IPL/SYS/$SERVER;
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>f;
PCALL:BEAMMON=/IPL/DPC/BEAM/BEAMMON,<BEAMDB>/IPL/DIC;
BG:/IPL/SP/SPGPA,<>1;
BG:/IPL/BE/BATCHGPA;
```

GRANDPA CONFIGURATION FILE FOR NOM129 LOCAL AGENT

```
INIT:MCL=/IPL/SYS/$VSH,<$user>ROOT<$cmd>IPL/END/CMD/TLOAD;
TTYA:GMASTER!DATE;
TTYA:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN!SHUTDOWN=SHUTDOWN;
TTYB:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYC:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYD:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYE:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYF:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
PFG:RFA=/IPL/SYS/$SERVER;
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>f;
BG:/IPL/BE/BATCHGPA;
```

GRANDPA CONFIGURATION FILE FOR NOM130 LOCAL AGENT

```
INIT:MCL=/IPL/SYS/$VSH,<$user>ROOT<$cmd>IPL/END/CMD/TLOAD;
TTYA:GMASTER!DATE;
TTYA:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN!SHUTDOWN=SHUTDOWN;
TTYB:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYC:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYD:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYE:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYF:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
TTYG:MCL=/IPL/SYS/$VSH!APPLICATION=/IPL/DPC/BEAM/BEAMMAIN;
PFG:RFA=/IPL/SYS/$SERVER;
PCALL:QUEMAN=/IPL/$QM/CODES/QUEMAN,<>f;
BG:/IPL/BE/BATCHGPA;
```

I. INDEX

D

E

Code 4002160 B(9)
Printed in Italy