

English



FUJITSU Software BS2000

OMNIS V8.5 / OMNIS-MENU V3.5

Administration and Programming

User Guide

Edition April 2015

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:

manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH

www.cognitas.de

Copyright and Trademarks

Copyright © 2015 Fujitsu Technology Solutions GmbH.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

Contents

1	Preface	9
1.1	Objectives and target groups of this manual	9
1.2	Summary of contents	10
1.3	Changes since the last edition of the manual	11
1.4	Notational conventions	11
2	SNMP management for OMNIS	13
2.1	Architecture of SNMP management	13
2.1.1	The master agent	15
2.1.2	The subagent	16
2.1.3	Management applications	17
2.2	Declaring the SNMP management system in OMNIS	17
2.3	SNMP monitoring in OMNIS	18
2.3.1	Predefined traps	19
2.3.2	Trap for freely-selectable OMNIS messages	20

Contents

3	Files and libraries	23
3.1	Bulletin file	26
3.2	ENTER file	27
3.3	Command files	28
3.4	Logging file	30
3.5	Paging file	35
3.6	Trace file	36
3.7	Message file	38
3.8	Start-up file	39
3.9	Text file	40
3.10	Libraries	42
3.11	Overview of files required by OMNIS-MENU	43
4	Installation and startup	47
4.1	Creating the OMNIS files and user ID	47
4.2	Starting OMNIS	48
4.2.1	Start parameters	49
	APPLICATION - Define DCAM resources	51
	DATA - Defining variable substitution	52
	END - Terminate start parameters	52
	FILE - Define file names	53
	LENGTH - Define length of areas	54
	SECURITY - Define level of operational security	55
	SETSW - Set switches	57
4.2.2	Multiple OMNISes	58
4.3	Notes on installing OMNIS-MENU	59

5	Administration	61
5.1	Tasks of the administrator	61
5.2	Administration concept	63
5.3	Administration commands at a glance	66
5.4	Special OMNIS-MENU administration requirements	68
5.4.1	Unlocking a terminal	69
5.4.2	Dynamic modification of configuration	70
5.5	Variable concept	71
6	Data protection	79
6.1	Partitioning	79
6.2	Access to OMNIS	79
6.3	Access to the partners of OMNIS	80
6.4	Exit routines and security	80
6.5	Diagnostic information	81
6.6	Passwords	83
6.7	Level of operational security: SECURITY	84
6.8	Behavior in the event of an error	85
7	Settings in BS2000 and partners	87
7.1	BS2000 system	87
7.2	UTM generation	91
7.2.1	Generation without multiplex connections	91
7.2.2	Generation for multiplex connections	94
7.3	Prerequisites for DCAM programming	99

8	OMNISKD: Declaring a configuration	101
8.1	Using OMNISKD	101
8.2	Notational conventions	109
8.3	OMNISKD control statements	112
	DECLARE-APPLICATION - Declare an application	112
	DECLARE-CONSOLE - Declare a BS2000 console	120
	DECLARE-EXIT - Declare an exit group	121
	DECLARE-GROUP - Declare a partner group	122
	DECLARE-HARDCOPY - Declare a printer	123
	DECLARE-MENU - Declare a menu	125
	DECLARE-MUX - Declare a multiplex connection	129
	DECLARE-OMNIS - Declare OMNIS	131
	DECLARE-PARAMETER - Declare parameters	133
	DECLARE-PARTNER - Declaring a partner	138
	DECLARE-PKEYSET - Declare a P key set	149
	DECLARE-START-SEQUENCE - Declare a start sequence	150
	DECLARE-TAC - Declare an entry in the TAC table	151
	DECLARE-TERMINAL - Declare a terminal	153
	DECLARE-USER - Declare a user	167
9	OMNISMD: generation program	173
9.1	Starting OMNISMD	173
9.2	Working with OMNISMD	175
9.3	Masks	179
	Function Menu	180
	Define User	181
	Define User Menu	186
	Define Menu Entry	190
	Define Application	192
	Define Parameters	200
	Print Tables	203
	Edit start sequence	205
9.4	Terminating OMNISMD	206

10	Configuration examples	207
10.1	Example 1: OMNIS configuration	207
10.2	Example 2: OMNISKD / OMNIS-MENU	210
11	OMNISFC: Converting configuration files	225
11.1	Working with OMNISFC	225
11.2	Changing versions	229
11.2.1	Changing the OMNIS version	229
11.2.2	Changing the OMNIS-MENU version	231
12	OMNISLC: Editing the logging file	235
12.1	Using OMNISLC	235
12.2	OMNISLC control statements	238
	END - Terminate OMNISLC	238
	HELP - Display command syntax	238
	INPUT - Define input file	239
	OUTPUT - Defining the output file	239
	RUN - Starting conversion	240
	SELECT - Select partners/terminals	241
12.3	Format of the output	244
12.3.1	Output when EDIT=NO	244
12.3.2	Output when EDIT=SAM	245
12.3.3	Output when EDIT=ISAM	246
13	OMNISTC: Converting text files	247

14	Programming with OMNIS	249
14.1	Description	249
14.2	Data areas	257
14.3	Operation codes	261
14.3.1	OMNIS calls exit routine	261
14.3.2	Exit routine calls OMNIS	263
14.4	Parameter areas	269
14.5	Assembler interface	278
14.5.1	Calling an exit routine in Assembler (OMEPARL)	278
14.5.2	Data structures (OMECPAR)	279
14.5.3	Definitions for supplying the data structures with values (OMEEQU)	282
14.5.4	H-Assembler	289
14.5.5	Columbus-Assembler	289
14.6	COBOL interface	290
14.6.1	Structure of an exit routine in COBOL	290
14.6.2	Data structures (OMECPAR)	291
14.6.3	Definitions for supplying the data structures with values (OMECEQU)	294
14.6.4	Resolution of recursive calls	299
14.7	Messages from exit routines	300
14.8	Preparations for use	301
14.8.1	Testing	303
14.9	Examples	305
14.9.1	Designing an exit routine	305
14.9.2	COBOL examples	314
	Related publications	327
	Index	329

1 Preface

The software product **OMNIS** is a control system that lets you use more than one application in BS2000.

A terminal can maintain only one connection to one communication partner at any one time. OMNIS is a program that allows you to overcome this restriction.

OMNIS lets you set up connections from one terminal to a number of partners in a network concurrently. Using OMNIS you can also maintain multiple connections to one partner. OMNIS further allows you to use terminals for applications that are not normally supported by terminals, such as UCON.

You can customize the functions offered by OMNIS by writing your own exit routines.

OMNIS-MENU offers a menu-driven interface that enables users to set up and clear down connections to their applications without an in-depth knowledge of OMNIS. Switching from one application to another is considerably simplified by the use of keys.

It also allows you to switch from OMNIS-MENU directly to the OMNIS command mode and use other OMNIS functions. Authorization to switch to command mode is granted by the administrator.

The data to create menus can be entered by the OMNIS administrator with the aid of the **OMNISKD** configuration definition program and **OMNISMD (OMNIS MENU Definition)**.

1.1 Objectives and target groups of this manual

This manual is intended for the OMNIS administrator and for those users who want to create OMNIS exit routines.

It is assumed that readers have a basic knowledge of BS2000.

1.2 Summary of contents

The description of OMNIS and OMNIS-MENU comprises of two manuals:

- The present manual “Functions and Commands” contains basic information for ordinary users plus full, in-depth descriptions of the OMNIS commands.
- The manual “Administration and Programming” contains in-depth informations for the OMNIS administrator and those users, who wish to create OMNIS exit routines.

Readme file

The functional changes to the current product version and revisions to this manual are described in the product-specific Readme file.

Readme files are available to you online in addition to the product manuals under the various products at <http://manuals.ts.fujitsu.com>. You will also find the Readme files on the Softbook DVD.

Information under BS2000

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

```
SYSRME.<product>.<version>.<lang>
```

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the `/SHOW-FILE` command or an editor. The `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` command shows the user ID under which the product's files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online at <http://manuals.ts.fujitsu.com>.

1.3 Changes since the last edition of the manual

The following major changes have been made to the OMNIS V8.5 / OMNIS-MENU V3.5 manuals since the previous editions:

- The manuals have been adjusted to the conditions of OMNIS V8.5 / OMNIS-MENU V3.5.
- The Readme file for OMNIS/OMNIS-MENU has been integrated.
- The “Supplement” manual of OMNIS/OMNIS-MENU had been integrated.
- The use of OMNIS over the internet is obsolete. The description has been removed.
- SKP and SVP partners are obsolete. Their description has been removed.
- Information on PDN, TRANSDATA, SKP and SVP is obsolete. Their description has been removed.
- The software products FHS-DOORS and PROP-PPM are obsolete. Their description has been removed.
- You can also find the standard messages of OMNIS on our manual server using the HTML application in the place of the former “System Messages” manual (URL: <http://manuals.ts.fujitsu.com>) and on the DVD “BS2000 SoftBooks”.
- The glossary and abbreviations have been removed from the “Administration and Programming” manual. They are contained in the “Functions and Commands” manual.

1.4 Notational conventions

The following typographical elements are used in this manual:



For notes on particularly important information



This symbol designates special information that points out the possibility that data can be lost or that other serious damage may occur.

References to other publications within the text are given in abbreviated form followed by numbers; the full titles are listed in the “References” section at the back of this manual.

In the text, the names of macros, commands and operands are shown in *italics*, as are file names, path names and screen elements.

2 SNMP management for OMNIS

Some of the functionality of OMNIS is also implemented via products of the SNMP (BS2000) product family.

SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) was developed as the protocol for network management services in TCP/IP to enable LAN components (routers, bridges, etc.) to be monitored and managed in heterogeneous TCP/IP networks. Since then, the area of application of SNMP has been extended to include systems and application management. Like TCP/IP, which is used to refer not just to the protocols, but to a type of network, SNMP also represents not only a protocol, but a type of management system. Two new system and application management options are offered via this management system, i.e., by integrating SNMP:

1. Terminals, partners and applications can be monitored from an SNMP agent. For each terminal, partner and application that is to be monitored, the operand *SNMP=YES* must be set. This operand is available in the commands *OPNCON* and *PARAM* and in the OMNISKD statements *DECLARE-PARTNER*, *DECLARE-APPLICATION* and *DECLARE-TERMINAL*.
2. OMNIS administration is also possible from the management platform via the OMNIS subagent. OMNIS commands can be sent and the corresponding responses (OMNIS messages) can be received.



Both for the administration and for the monitoring function, one subagent must be defined as a terminal in OMNIS with the *DECLARE-TERMINAL* statement in OMNISKD (see [page 17](#) for details).

2.1 Architecture of SNMP management

The system and application management functions of OMNIS can be followed from the Network Operation Center (NOC) or from any agent in the SNMP network. A terminal with a graphical user interface is used on the SNMP system, usually a UNIX system, to display the management activities and output the monitoring messages.

The basis for the interoperation of OMNIS and the SNMP components is the precise description of the objects to be managed (OMNIS commands and messages). This description is contained in the IT backbone of every agent in the SNMP network, the MIB (**M**anagement **I**nformation **B**ase). The MIB maps the commands and messages from the OMNIS world (BS2000) to the SNMP world, and vice versa.

Where SNMP is not used, i.e. in the OMNIS world of OMNIS V8.0 or lower and OMNIS-MENU V3.0 or lower, the administrator can call up information about the current connections of the terminals and partners using the OMNIS command *INF* at a terminal at OMNIS. In accordance with this information, any OMNIS commands can also be entered at the terminal and acknowledgments of these commands received there. The *CANCEL* command can be used at any time to clear down a connection, although this can also be carried out by the partners.

Where SNMP is used, this functionality, in addition to the OMNIS command *INF*, is also implemented by the subagent. The following architecture is required:

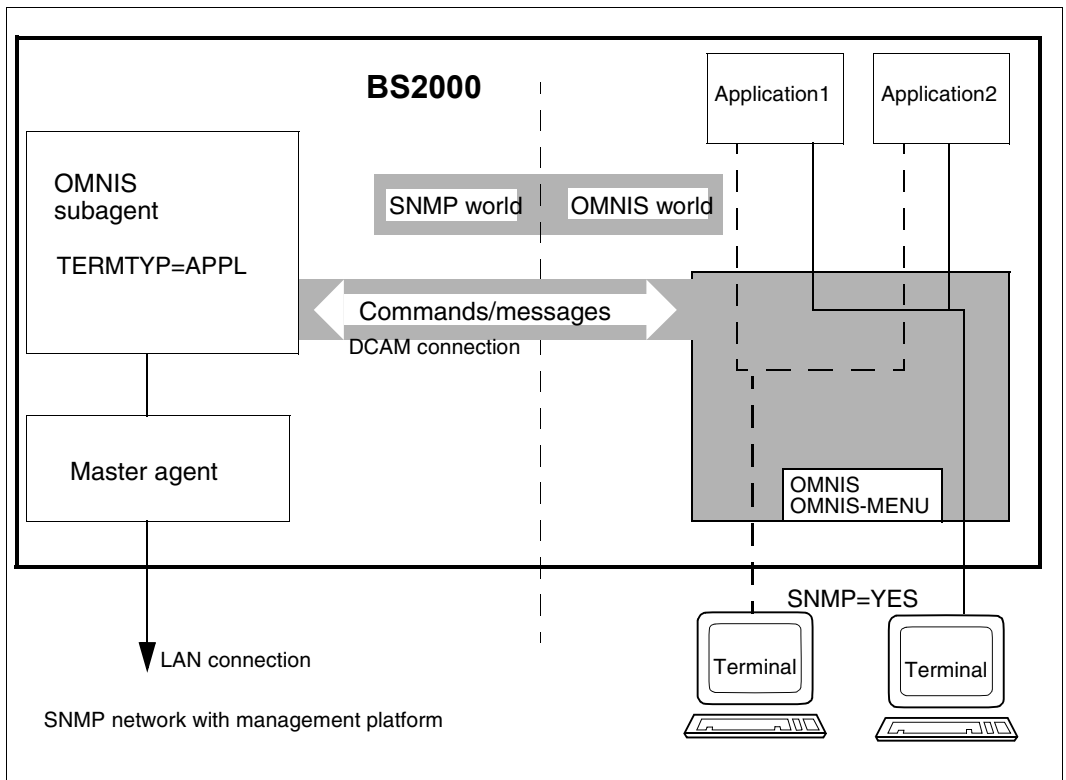


Figure 1: Architecture of the OMNIS-SNMP connection

OMNIS is linked to SNMP management with the products SNMP Basic Agent BS2000 (SBA-BS2), which contains the master agent, and SNMP Standard Collection BS2000 (SSC-BS2), which contains the OMNIS subagent.

These products enable BS2000 systems to be integrated into SNMP-based management platforms (e.g. CA Unicenter) by implementing the SNMP protocol in BS2000. The SNMP link implemented in BS2000 consists of a master agent and several subagents. One of these subagents is the OMNIS subagent, which communicates with OMNIS via a DCAM connection. The BS2000 system is linked to the SNMP management system by a LAN connection which operates with TCP/IP protocols.

The basis for this solution is the EMANATE product from SNMP Research. EMANATE has been ported to BS2000 and is also available on LINUX and on UNIX derivatives from almost all well-known vendors, as well as on WINDOWS. The interface between the master agent and the subagents is also automatically generated by EMANATE.

2.1.1 The master agent

The master agent is master to the subagents, and thus to the OMNIS subagent as well. It has basic functions, such as those involved in processing the SNMP protocol, in security and in subagent management, and can also run without subagents.

Since subagents can be started and terminated individually, they can also be modified individually; it is not necessary to terminate the entire SNMP management system. The master agent communicates with the subagents via an asynchronous message interface, which consists of a system part of the master agent and a system-independent part (API) of the subagent.

The master agent is shipped with the product SNMP Basic Agent BS2000 (SBA-BS2). The SNMP management system communicates with OMNIS via the master agent and from there via the OMNIS subagent, which communicates with OMNIS via a DCAM connection.

2.1.2 The subagent

The subagents are only functional if the master agent has been started.

They are event-oriented, i.e., are activated when a certain event (command, message) occurs.

The OMNIS subagent is part of the product SNMP Standard Collection BS2000 (SSC-BS2). It offers the following functions:

- **Configuration processing**

When it is started, the OMNIS subagent reads and saves the configuration file, which defines the OMNISs to be monitored.

- **Structure of the configuration file**

Each OMNIS to be monitored is specified by *omnisname* in a separate record in the file. Example of a configuration file:

```
OMSVTT..  
OMSEKT..
```

If *omnisname* is less than 8 characters (e.g. *OMSEKT*), you will need to pad the name to 8 characters by adding an appropriate number of blanks.

- **Establishment of connections to the OMNISs**

Once the subagent has been started, it attempts to set up connections to every OMNIS that is not currently connected to SNMP. A connection to an OMNIS can also be requested via the management platform. This is required especially if the OMNISs to be monitored were started when the OMNIS subagent was already active. Only local OMNISs can be monitored with the OMNIS subagent.

- **Administration of OMNIS via the SNMP manager**

The subagent receives the commands (requests in the SNMP world) from the SNMP management system, processes them and passes them on to OMNIS. It then synchronously waits for the response(s) from OMNIS and sends back the modified result to the management platform.

- **SNMP monitoring in OMNIS**

Terminals, partners and applications can be monitored from an SNMP management system via the subagent (see [page 18](#)).

- **Timer function**

By default, an attempt is made every minute to set up a connection to inactive OMNISs defined in the configuration file.

This time interval can be controlled via the *Timer-Interval* parameter on starting the agent.

2.1.3 Management applications

SNMP agents in BS2000 can be used for any management application of the SNMP network. Management applications from different vendors with different features can therefore also be used for monitoring and controlling OMNIS.

2.2 Declaring the SNMP management system in OMNIS

To administer OMNIS via an SNMP management system, you must declare the subagent as a special SNMP terminal. This logical SNMP terminal is declared in OMNISKD with the statement *DECLARE-TERMINAL*. The following fixed values must be used for the operands below:

- The *PTNNAME* operand must have the value *OMNISAGT*. The terminal specified with *PRONAME/PTNNAME* must be unique throughout the network. You cannot make multiple declarations of terminals defined with *PTNNAME=OMNISAGT*
- The *TERMTYP* operand must have the value *TERMTYP=APPL*. This declares the SNMP terminal as a DCAM application, since there is a DCAM connection between the subagent and OMNIS.
- The *USER-PROTOCOL* operand must have the value *USER-PROTOCOL=NO*. No protocol is then used, i.e. only the net message is transmitted to the subagent.
- The *ADM* operand must have the value *ADM=YES*. This allows all commands (including administration commands) to be entered from the SNMP terminal, and thus on the SNMP management system.
- The *EXIT* operand must have the value *EXIT=NO*. This prevents the SNMP terminal from using exit groups for message traffic.

Example:

```
DECLARE-TERMINAL PTNNAME=OMNISAGT, -
                  PRONAME=OMNISHOST, -
                  TERMTYP=APPL, USER-PROTOCOL=NO, ADM=YES, EXIT=NO
```

Administration of OMNIS via subagents can be performed only via the SNMP terminal, which can be specified only once in OMNISKD.



The SNMP management system can be connected to OMNIS via the subagent only if OMNIS is not protected by a connection password.

2.3 SNMP monitoring in OMNIS

In OMNIS and OMNIS-MENU, terminals, partners and applications can be monitored from an SNMP management system via a subagent. The operand *SNMP=YES* must be set for each terminal, partner or application to be monitored.

This operand can be specified on-line with the OMNIS commands *OPNCON* and *PARAM*. SNMP monitoring can then be activated and deactivated **dynamically**.

The OMNISKD statements *DECLARE-PARTNER*, *DECLARE-APPLICATION*, and *DECLARE-TERMINAL* also include the operand *SNMP=YES/NO*. This operand can be used to activate and deactivate SNMP monitoring **statically** for a specific terminal, partner or application.

Traps to the SNMP manager

In order to support monitoring of OMNIS via SNMP, certain events and messages must be reported as traps to the SNMP manager. A trap is a message or event in OMNIS that is transmitted by the OMNIS subagent via the master agent to the SNMP manager. The SNMP manager is therefore automatically informed about certain events, to which it can then respond appropriately.

To generate a trap from an OMNIS message or an OMNIS event, the OMNIS subagent takes the relevant output (OMNIS messages, return codes) from DCAM and maps it into the objects defined for this purpose in the MIB. These trap objects are then sent by the OMNIS subagent to the SNMP manager via the master agent.

Two variants for generating traps from OMNIS messages are available:

- A number of OMNIS messages are mapped as usual by the OMNIS subagent to predefined traps, see [page 19](#).
- In the case of OMNIS messages that are not mapped to predefined traps, the user can now generate traps from them with the OMSMSGEX exit routine. A common trap number is available for these messages; see also the [section “Trap for freely-selectable OMNIS messages” on page 20](#).

2.3.1 Predefined traps

Some OMNIS messages are mapped by the OMNIS subagent to predefined traps.

The following traps are provided for in the MIB:

- If a disconnect on a monitored connection occurs, the appropriate message (OMS0006, OMS0020, OMS0055, OMS0087, OMS0155) is sent to the SNMP management system via the subagent as well.
- Messages to the SNMP manager when an OMNIS dump occurs.
- A trap when a MUX connection, an OMNIS-OMNIS connection or a hardcopy connection is cleared or if a connection is lost (OMS0067, OMS0088, OMS0089, OMS0159, OMS0251, OMS0252).
- Messages to the SNMP manager on reaching specific thresholds such as the maximum number of DST, PTN and PAC. The messages OMS0021, OMS0022 or OMS0057 should be forwarded to the manager.
- Sending a trap to the SNMP manager when an OMNIS command to an OMNIS is rejected by DCAM (YSEND with error code 0C04). This means that the connection to that OMNIS was cleared.
- Sending a trap to the SNMP manager for every OMNIS to which a connection was successfully set up on initializing the subagent. A cleared connection at a later point in time is not reported to the SNMP manager via a trap.

2.3.2 Trap for freely-selectable OMNIS messages

You can set up the messages output from OMNIS to SNMP with great flexibility. A special trap number to which you can assign OMNIS messages of your choice is provided for this purpose. These assignments can be made by using the OMNIS exit routine OMSMSGEX.

The **OMSMSGEX exit routine** is an Assembler routine and must have the following structure:

```
OMSMSGEX AMODE ANY
OMSMSGEX RMODE ANY
OMSMSGEX START 0
OMSMSGEX CSECT
        PRINT GEN
        OMSMSGEM MF=L, MSGNR='OMSnnnn',SNMP=*YES
        OMSMSGEM MF=L, MSGNR='OMSnnnn',SYSOUT=*YES
        ...
        END
```

For each desired OMNIS message number, you will need to insert an entry in OMSMSGEX by using the OMNIS message macro OMSMSGEM.

The **OMNIS message macro OMSMSGEM** is available in the interface library *SYSLIB.OMNIS.085*. OMSMSGEM describes the actions following the OMNIS message with the *OMSnnnn* number:

```
OMSMSGEM MF=L/D,
MSGNR='OMSnnnn',[SNMP=*YES],[SYSOUT=*YES],[SYSLST=*YES],[CONSOLE=*YES]
```

MF	Determines the type of macro generation: <i>MF=D</i> generates a DSECT <i>MF=L</i> generates the data area (in the definition section)
MSGNR	Number of the OMNIS message for which an action is defined.
SNMP	If <i>SNMP=*YES</i> , the OMNIS message is also sent to the OMNIS subagent.
SYSOUT	If <i>SYSOUT=*YES</i> , the OMNIS message is also output to SYSOUT.
SYSLST	If <i>SYSLST=*YES</i> , the OMNIS message is also output to SYSLST.
CONSOLE	If <i>CONSOLE=*YES</i> , the OMNIS message is also output to the BS2000 console.

OMNIS executes the actions defined in the OMSMSGEM message macro as soon as the OMNIS message specified in the macro is output. The OMSMSGEM actions are executed by OMNIS in addition to those already generated as fixed actions in OMNIS.

OMNIS loads the exit routine OMSMSGEX dynamically, but does not enter it. The exit routine is only used to obtain information on the actions for messages.

Using OMSMSGEX

The exit routine OMSMSGEX must be compiled and placed in the module library of OMNIS; see also "Preparations for use" on [page 301](#).

You can **generate** the exit routine OMSMSGEX with the command `EXIT EMSG, ACTION=CREATE, MODULES=OMSMSGEX` or with the statement `DECLARE-EXIT EMSG, MODULES=OMSMSGEX`.

Note that the exit routine OMSMSGEX can only be generated once.

The command `EXIT EMSG, ACTION=MODIFY, MODULES=OMSMSGEX, LOAD=YES` **loads** a current version of OMSMSGEX **dynamically**.

The command `EXIT EMSG, ACTION=DELETE` **deletes** the OMSMSGEX exit routine and disables this function.

The OMSMSGEX exit routine cannot be assigned with *SET*, *OPTION* or *PARAM*. It is always enabled **omnis-wide** on generation with the *EXIT* command or the *DECLARE-EXIT* statement.

3 Files and libraries

During execution, OMNIS requires several files, which must either be provided by the administrator before OMNIS is started or are generated by OMNIS itself.

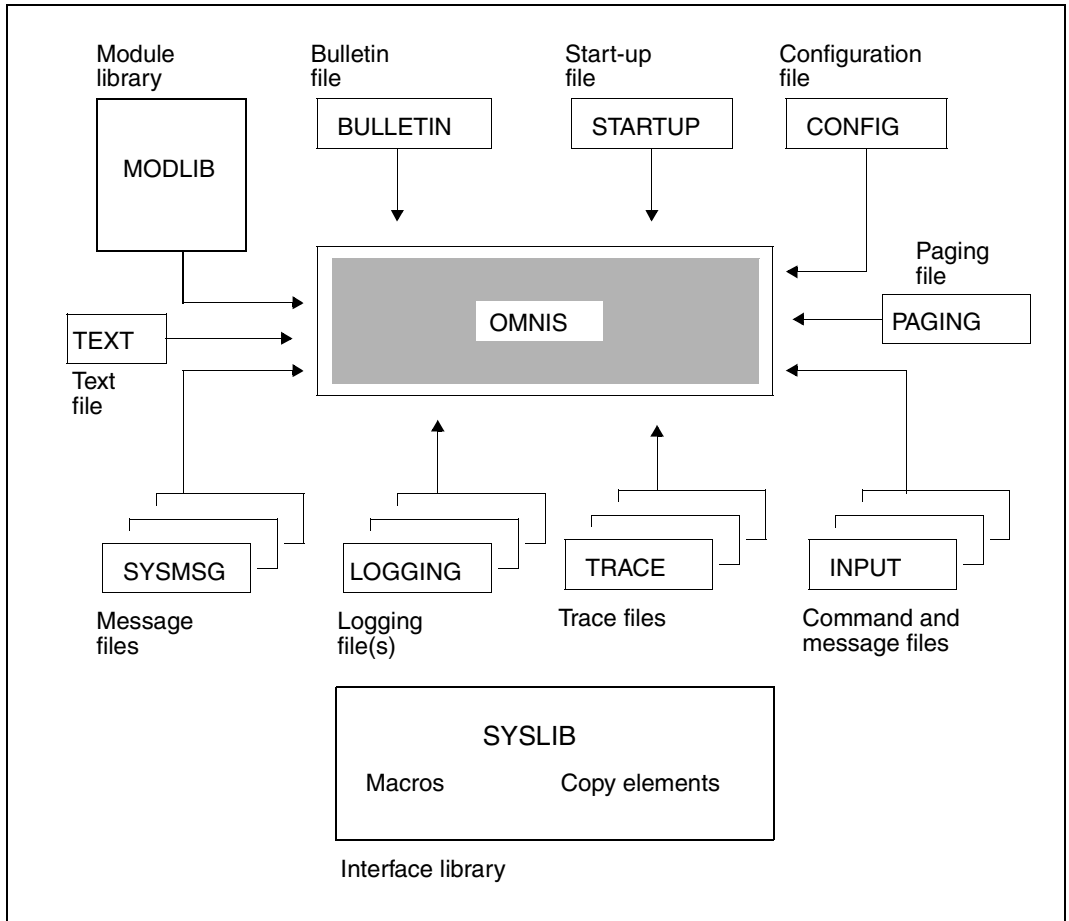


Figure 2: Files required by OMNIS

The **start-up file** contains commands which are processed immediately after OMNIS is loaded. It must at least contain the command `SET NPASS=password` which defines the administration password.

The **bulletin file** is a message file (SAM or ISAM file or a type-S library member) whose contents are output at each terminal when it signs on to OMNIS.

The **logging file** is used to log the OMNIS dialog (when logging is activated). It creates a log of the dialog, so that messages from specific OMNIS partners or all OMNIS partners are also logged in this file. The structure of the logging file is described on [page 31](#). Information on the topic "dialog logging" can also be found in the "Functions and Commands" manual [1].

The **paging file** is used to buffer large amounts of data when there is message congestion. When OMNIS is terminated, the paging file is deleted.

The **trace file** is used to save trace entries (when tracing is activated). The trace file can be edited and evaluated with the TEDDY program, and is therefore an important diagnostic tool with regard to the behavior of the overall system.

The **configuration file** contains a configuration generated using the OMNISKD utility routine. This configuration is set up automatically by OMNIS when the latter is started. Generation of the configuration file is described in detail in the [chapter "OMNISKD: Declaring a configuration" on page 101](#). The configuration file must be created with the same OMNISKD version with which OMNIS is also started.

The **command files** (SAM or ISAM files or type-S library members) contain messages and user-specific command sequences for partners, which can be executed during operation.

The **text file** contains LOGON or KDCSIGN messages which are to be sent to the partner after a partner connection has been opened.

The **message files** (SAM or ISAM files or type-S library members) contain the texts of messages which the OMNIS administrator can send to terminals during operation.

The **interface library** contains macros (for assembling exit routines) and copy elements (for assembling exit routines written in COBOL).

Exit routines are loaded dynamically from the **module library** as required.

The **message files** contain all OMNIS messages and HELP texts in German and English.



OMNIS as a rule denies access to files which have migrated from HSMS (**H**ierar-**S**chical **S**torage **M**anagement **S**ystem).

This is intended to ensure that the automatic read-in associated with any access to such a file does not cause all users connected to OMNIS to be locked out until the file has been read.

This applies to: configuration file, message file, bulletin file, text file, command file, EXIT module library.

The only exception to this rule is the start-up phase of OMNIS, during which the system waits for files (configuration file, start-up file) to be read in.

3.1 Bulletin file

The bulletin file is output to a terminal immediately after the terminal signs on to OMNIS. The contents of this file are the responsibility of the OMNIS administrator. The file can be used to distribute information of general interest to OMNIS users (like a BS2000 bulletin).

The bulletin file is read in via SYSDTA and can therefore be a SAM or ISAM file or type-S library member. Record length is not restricted. The format of the bulletin file is described in the "Commands" manual [3].

The output of the bulletin file in the OMNIS login menu takes place in 8-bit mode if the terminal emulation supports this mode. In this case the OMMFMAI8 format is output for the login. For this purpose the bulletin file should be created using the Coded Character Set (CCS) EDF041. However, it is not necessary to define this CCS in the catalog entry.

There is basically no restriction on the number of records in the file, but it should not exceed the number of lines on a screen. All lines should be shorter than the length of the screen line minus 6.

The name of the bulletin file may be selected freely using the start parameter *FILE BULLETIN*; the default name is *OMNIS.BULLETIN*.

The bulletin file can also be modified during operation since it is only opened for a short period by OMNIS. If its contents are changed, it may be useful to have the file broadcast to all users once again:

```
@:OPT ADM=YES,AP=password
@:MES ALL,INPUT=OMNIS.BULLETIN
```

If you do this, however, you should remember that the command *MES ALL* temporarily blocks OMNIS operation for all users and, if there is a large number of terminals, may result in memory bottlenecks in the data communication system.

Example of a bulletin file

```
*****
* System DVA7                Computer Center Vienna *
*-----*
* This is an important announcement for all          *
* OMNIS users.                                         *
*-----*
* Telephone      OMNIS administrator 47658          *
* numbers:      Operator              36459          *
*               System administrator 47712          *
*****
```

3.2 ENTER file

You can call OMNIS not only by using the *START-PROGRAM* command but also by means of *START-OMNIS*; see the [section “Starting OMNIS” on page 48](#).

It makes sense to start OMNIS in batch mode.

For the **NEA application** you can use the following ENTER file (SAM or ISAM):

```

/.OMNIS LOGON
[ /      SET-DCAM-APPLICATION-LINK LINK-NAME=OMNIS           -
      [ ,APPLICATION-NAME=omnis-name]                       -
      [ ,PROTECTION=PARAMETERS(CONNECTION-PASSWORD=password)] ]
/      START-PROGRAM FROM-FILE=*M(LIB=SYSLNK.OMNIS.085,      -
/                                           ELEM=OMNIS,PROGRAM-MODE=*ANY,  -
/                                           RUN-MODE=ADV)
[ start-parameter                                           ]
[ /      REMARK * THIS IS WHERE OMNIS CONTINUES AFTER NORMAL ]
[ /      REMARK * TERMINATION (END COMMAND, /BCLOSE, /BCEND) ]
[ /      SKIP-COMMANDS TO-LABEL=ENDE                        ]
[ /      SET-JOB-STEP                                       ]
[ /      REMARK * THIS IS WHERE OMNIS CONTINUES AFTER      ]
[ /      REMARK * A PROGRAM ERROR                          ]
[ /      ENTER-JOB FROM-FILE=OMNIS.ENTER,further-operands ]
[ /.ENDE SET-JOB-STEP                                       ]
/      LOGOFF

```

The *SET-DCAM-APPLICATION-LINK* command can be used

- to change the application name of OMNIS (*omnis-name*),
- to define a password which must be entered at every terminal setting up a connection to OMNIS (*password*).

If *omnis-name* is omitted, OMNIS can be accessed with the application name *OMNIS*.

If *CONNECTION-PASSWORD* is omitted, no password is required to connect a terminal.

For the **ISO application** you must specify *LINK-NAME=OMNISISO* and set the start parameter *ISO*:

```

/ SET-DCAM-APPLICATION-LINK LINK-NAME=OMNISISO
      [ ,APPLICATION-NAME=omnis-name-iso]

```

The specification of *CONNECTION-PASSWORD* is meaningless in the case of ISO applications.

The *SET-DCAM-APPLICATION-LINK* command must always be specified if OMNIS is to be loaded more than once (see [page 48](#)).

In the event of program errors, OMNIS terminates itself with *TERM UNIT=STEP,MODE=ABNORMAL*, which means that processing is continued after the next *SET-JOB-STEP* command in the ENTER file. This facility can be used for **restarting OMNIS automatically**.

Examples of ENTER files

```

/.OMNIS LOGON
/      START-PROGRAM FROM-FILE=*M(SYSLNK.OMNIS.085,OMNIS)
/      SKIP-COMMANDS TO-LABEL=END
/      SET-JOB-STEP
/      ENTER-JOB FROM-FILE=OMNIS.ENTER,
/          RESOURCES=PARAMETERS(RUN-PRIORITY=80,CPU-LIMIT=NO)
/.ENDE SET-JOB-STEP
/      LOGOFF

/.OMNIS LOGON
/      START-PROGRAM FROM-FILE=*M(SYSLNK.OMNIS.085,OMNIS,RUN-MODE=ADV)
/      LOGOFF

```

3.3 Command files

Command files contain a sequence of OMNIS commands or messages to partners which are processed or sent one after the other without interruption being possible.

Processing of a command file is initiated in OMNIS by means of the *FILE* command. The commands or messages read from the file are processed as though they had been received from the terminal at which the *FILE* command was entered.

As the general address code rules also apply for command files, the @: in command files containing only OMNIS commands can be omitted.

The command file is read from SYSDTA, and may therefore be a SAM or ISAM file or type-S library member. It can have any record length.

When command files are created, the following information should be noted:

- If the command file contains another *FILE* command, the current command file is aborted and processing continues with the new command file.
- If the command file contains a *MESSAGE* command with an *INPUT=...* parameter, processing of the command file is aborted because SYSDTA is reassigned. Such a command, if included, should always be the last command in the file.

- During the processing of a command file, OMNIS is blocked for all other activities (e.g. receiving messages). Very extensive command files can have a disruptive effect on runtime performance.
- The execution of a command file is aborted if an *END* or *HALT* command is encountered.
- It is recommended that all commands and operands used in command files be written out in full, since the abbreviations may be changed as new commands or operands are implemented (to preserve the uniqueness of keywords).

Example of a command file

```
@:OPTION EXIT=ATOP,ACK=YES,LOGGING=YES,HOLD=YES
@:GROUP  ORT1,ADD=(A,B,C,D)
@:GROUP  UTM,ADD=(C,D)
@:OPNCON A,TYP=UCON,OPASS=C'OPASS',LM=CON1,C'PASS1'
@:PARAM  A,MTAB=M1
@:OPNCON B,TYP=TIAM,OPASS=X'45ADF7'
@:OPNCON C,TYP=UTM,PTN=UTMANW1,APP=ADMI1,PRO=HOST1,ROU=MUX
@:OPNCON D,TYP=UTM,PTN=UTMANW2,APP=ADMI2,PRO=HOST2,ROU=MUX
U:/STA MSG
TIA1:SHOW-USER-STATUS
@:INF    TERMINAL
@:INF    PARTNER
@:INF    GROUP
```

3.4 Logging file

The logging file is used by OMNIS to write a log of the dialog, which can be paged on-line.

The logging file is a PAM file with the link name SOMSLOG, which is automatically created when OMNIS is started.

PAMKEY simulation must be deactivated for the logging file, since OMNIS assumes that 2048 bytes per PAM page or 4*2048 bytes per PAM block are available for the data. This is done with the following command:

```
/SET-FILE-LINK LINK-NAME=SOMSLOG, BLOCK-CONTROL-INFO=NO
```

The size of the primary and secondary allocations must be a multiple of 96 (*SPACE* operand).

If *SECURITY LEVEL=HIGH* or *SECURITY LEVEL=MEDIUM* is specified in the start parameters, all OMNIS commands and messages are logged in the logging file, even if the *LOGGING=NO* operand is specified. If the logging function fails, OMNIS is terminated.

The start parameter *LOGGING* can be used to select a file name. OMNIS suffixes this file name with *.001*. The specified file name may not be more than $n-4$ characters long, where n is the maximum file name length (which depends on the BS2000 version). The default name is *OMNIS.LOGGING.001*.

Each time OMNIS is started, the logging file suffix is *.001*. Each time the logging file is switched, the number of the file (suffix *.00n*) is incremented by one. The logging file names in an OMNIS run when the file is switched several times are thus:

```
OMNIS.LOGGING.001  
OMNIS.LOGGING.002  
OMNIS.LOGGING.003  
OMNIS.LOGGING.004  
...
```

For more information on switching the logging file, see the section "Dialog protocol" in the "Functions and Commands" manual [1].

The "old" logging file can be analyzed off-line. The utility routine OMNISLC (see [page 235](#)) is used to edit the logging file and to convert it into a SAM or ISAM file.

If an error occurs whilst accessing the logging file, OMNIS is terminated immediately.

When written to the logging file, messages are truncated to a length of approx. 8000 bytes. The exact maximum number of bytes of net data depends on the volume of administration data. If a record is not completely logged, it will be reported in the administration data. *OMNISLC* can access this information when preparing the logging file and mark the record accordingly.

Structure of the logging file

Data is written sequentially to the logging file in the form of variable-length records. These records are chained to one another; there are four record chains:

- A terminal chain, where all records for one terminal are chained together; there is one chain for each terminal.
- A partner chain, where all records for one partner are chained together; there is one chain for each partner.
- A command chain, where all OMNIS commands are chained together.
- A message chain, where all OMNIS messages are chained together.

The records of each chain are chained both forward and backward.

Each record of the logging file is a member of just two of the above chains: a terminal chain and one of the other three.

Format of a record in the logging file

The format of a logging file record is described as a DSECT in the *OMSXLOG* macro:

OMSXLOG MF=D (OMSXLOG)		Key:	
-----		*-----*	
* DSECT	FUER EINEN SATZ IN DER PAM-DATEI	* DSECT	FOR A RECORD IN THE PAM FILE
-----		*-----*	
LOGSATZ	DSECT		
SALAENGE	DS Y	GESAMT-SATZLAENGE	total record length
SALOGPAR	DS X	LOGGINGPARAMETER	logging parameter
SATYP	DS X	SATZTYP	record type
SANACHR	EQU X'80'	NACHRICHT	message
SAMESMAP	EQU X'40'	OMNIS-MELDUNG MIT MAPPING FORMAT	OMNIS message with mapping format
SAVTSUCB	EQU X'20'	SATZ HAT VTSUCB	record has VTSUCB
SAOUTYES	EQU X'10'	SATZ SCHON AUSGEGEBEN	record already output
SAINPUT	EQU X'0C'	INPUT SATZ	input record
SALOGANW	EQU X'01'	ANWENDER LOGGING-SATZ (LOG = YES)	user logging record (LOG = YES)
SANBMHE	EQU X'02'	SATZ HAT NBMHE-HEADER	record has NBMHE header
SATAC	DS CL4	TAC DES PTN	partner's tac
SAPTID	DS A	ZVT-ID DES PTN	partner ID
SADSTID	DS A	ZVT-ID DER DST	terminal ID
SADATE	DS CL10	DATUM AN DEM EINTRAG ERFOLGTE	date entry was made
	ORG SADATE+2		
SADATE8	DS CL8	DATUM OHNE JAHRHUNDERT	date without century
SATIME	DS CL8	ZEIT ZU DER EINTRAG ERFOLGTE	time entry was made
SAUSER	DC CL8' '	BENUTZERNAME ODER BLANK	user name or blank
SANETZ	DS QL16	NETZADRESSE DER DATENSTATION	network address of terminal
SAPTNAME	DC CL8' '	STATIONSNAME	station name
SAPRONAME	DC CL8' '	PROCESSORNAME	processor name
	DS H		
SAPLNKF	DS OA	PTN-SPEZ. VORWAERTSVERKETTUNG	partner-specific forw. chaining
SAPLNKF#	DS Y	PAM-BLOCKNUMMER	PAM block number
SAPLNKF§	DS Y	DISTANZ	displacement
SAPLNKB	DS OA	PTN-SPEZ. RUECKWAERTSVERKETTUNG	partner-specific backw. chaining
SAPLNKB#	DS Y	PAM-BLOCKNUMMER	PAM block number
SAPLNKB§	DS Y	DISTANZ	displacement
SADLNKF	DS OA	DST-SPEZ. VORWAERTSVERKETTUNG	terminal-specific forw. chaining
SADLNKF#	DS Y	PAM-BLOCKNUMMER	PAM block number
SADLNKF§	DS Y	DISTANZ	displacement
SADLNKB	DS OA	DST-SPEZ. RUECKWAERTSVERKETTUNG	terminal-specific backw. chaining
SADLNKB#	DS Y	PAM-BLOCKNUMMER	PAM block number
SADLNKB§	DS Y	DISTANZ	displacement
SAOLNKF	DS OA	OMNIS-SPEZ. VORWAERTSVERKETTUNG	OMNIS-specific forw. chaining
SAOLNKF#	DS Y	PAM-BLOCKNUMMER	PAM block number
SAOLNKF§	DS Y	DISTANZ	displacement
SAHLNKF	DS OA	HCY-SPEZ. VORWAERTSVERKETTUNG	hardcopy-specific forw. chaining
SAHLNKF#	DS Y	PAM-BLOCKNUMMER	PAM block number
SAHLNKF§	DS Y	DISTANZ	displacement
SALOGTYP	DS X	LOGGINGTYP	logging type
SARECP	EQU X'80'	LOG IN SENDPTN	logging in SENDPTN
SASENDP	EQU X'40'	LOG IN SENDDST	logging in SENDDST
SARECD	EQU X'20'	LOG IN RECPTN	logging in RECPTN
SASEND	EQU X'10'	LOG IN RECDST	logging in RECDST
SADCM	EQU X'08'	NACHRICHT WURDE ZWISCHENGESPEICHERT	message has been buffered
SADIAG	EQU X'04'	OMNIS-OMNIS DIAGNOSE LOGGING	OMNIS-OMNIS diagnostic logging
SASEND	EQU X'02'	LOG IN SENDOMS	logging in SENDOMS
SARECO	EQU X'01'	LOG IN RECOMS	logging in RECOMS
SATACTYP	DS X	TACTYP (PRE,LINE,NO STD)	tac type (PRE,LINE,NO,STD)
SATALEN	DS H	TACLAENGE	tac length
SAEXCOLP	DS XL1	PTN-FARBE AUS EXIT-ROUTINE	partner color from exit routine
SAEXCOLD	DS XL1	DST-FARBE AUS EXIT-ROUTINE	term. color from exit routine
SAMESTYP	DS XL1	WEITERE ANGABEN ZUM SATZTYP	further record type specifications
SAMESSTA	EQU X'80'	SATZ IST STAUSANFORDERUNG	record is status request
SAMESCHI	EQU X'40'	SATZ IST CHIPKARTENANFORDERUNG	record is chip card request
SA7BITS	EQU X'20'	7-BIT-CODIERTE NACHRICHT	7-bit coded message
SATPROT	EQU X'10'	DST IST ISO-ANWENDUNG	terminal is ISO application

SACDS	DS	XL1	CODE DER SENDETASTE	code of send key
SACOL	DS	XL1	AKTUELLE FARBE DES PARTNERS	current partner color
SATYP	DS	XL1	TERMINALTYP	terminal type
SANBMLEN	DS	XL2	LAENGE NBMHE ; NMAP HEADER	length NBMHE ; NMAP header
SAPTTY	DS	XL1	PARTNERTYP	partner type
SADIMO	DS	XL1	DISPLAY-MODE FUER UCON PARTNER	display mode for UCON partner
	DS	XL1	FREI	free
SATYP1	DS	X	SATZTYP	record type
SASCRANF	EQU	X'80'	SCREENSAVE	screen save
SAKEYANF	EQU	X'40'	P-TASTEN	P keys
SAFIRSTMES	EQU	X'20'	NACHRICHT IN DIE 25. ZEILE	message in 25th line
SASAVEYES	EQU	X'10'	NACHRICHT EINES INAKTIVEN PARTNERS	message of an inactive partner
SASSH	DS	0XL8	SSH MIT EDIT-OPTIONS	SSH with edit options
SASSHTYP	DS	X	NACHRICHTENTYP	message type
SASSHZLE	DS	X	PHYSIKALISCHE ZEILENLAENGE	physical line length
SASSHTTY	DS	X	TERMINALTYP	terminal type
SASSHE01	DS	X	OUTPUT-EDIT-OPTIONS	output edit options
SASSHE02	DS	X		
SASSHE11	DS	X	INPUT EDIT OPTIONS	input edit options
SASSHE12	DS	X		
	DS	XL1		
SAFIXLEN	EQU	*-LOGSATZ	LAENGE DER VERWALTUNGSDATEN	length of administrative data
SADATEN	EQU	*	DATENBEREICH (LAENGE VARIABLE)	data area (variable length)

The *SATYP* field indicates whether the logged record is an input or an output.

The fields *SANACHR* and *SAMELD* may not be used. The PID should be used to distinguish between messages and system messages.

A chaining address within the logging file consists of four bytes:

link	DS	0A	chaining address
link#	DS	Y	PAM block number
link@	DS	Y	displacement in the PAM block

The PAM block numbers are relative to the 8K blocks in the PAM file (1, 2, 3, etc.). The associated PAM page numbers (1, 5, 9, etc.) are obtained as follows:

$$(\text{PAM page number}) = (\text{PAM block number}) * 4 - 3.$$

The forward chaining of the last record of a chain, and the backward chaining of the first record, are set to binary zero.

The logged messages are to be found starting at address *SADATEN*. The length is calculated from the record length in the half-word *SALAENGE* minus the fixed record length (EQU *SAFIXLEN*).

It should be noted that the records in the logging file are concatenated without gaps, i.e. not justified.

Processing the logging file

All system message records have PID 1, all command records PID 7. The PID of the partner records is allocated dynamically and is greater than 9. The first record in the logging file is a record containing the version number followed by a system message record.

To search for all records associated with a particular terminal, the following steps are required:

1. Read in the first record of the logging file (address X'00010000').
2. Read further records in the system message chain with PID 1 until the first record of the terminal with the desired TID is found.
3. Read the records of the terminal chain up to the end of the chain.

Printing the logging file

The logging file can be used to document the dialog after OMNIS has been terminated or the logging file has been switched, or evaluated according to different program view points.

The OMNISLC utility can be used to prepare any of the following for printing:

- the entire logging file
- the dialog with a selection of terminals
- the dialog with a selection of partners
- the dialog with a selection of users (if OMNIS-MENU is used)

The messages (and their formats), and also all passwords are prepared as they would normally appear on the monitor.

3.5 Paging file

The paging file is used by OMNIS to buffer large amounts of data during operation. This is necessary when messages cannot immediately be output to a terminal and must be temporarily saved by OMNIS. This is specifically the case when

- the terminal is in the CALL interrupt state
- paging mode is active
- an acknowledgment point has been reached
- the sender of the message is assigned to message flow class SAVE
- the last format output at the terminal is one which must not be destroyed
- the terminal is in the HOLD state (and the partners are active)

The paging file is a PAM file with the link name SOMSPAGE, which is automatically created when OMNIS is started.

PAMKEY simulation must be deactivated for the paging file, since OMNIS assumes that 2048 bytes per PAM page or 16*2048 bytes per PAM block are available for the data. This is done with the following command

```
/SET-FILE-LINK LINK-NAME=SOMSPAGE, BLOCK-CONTROL-INFO=NO
```

The size of the primary allocations must be a multiple of 384;
the size of the secondary allocations must be a multiple of 384 or 0.

The primary and secondary allocations are made in the *CREATE-FILE* command:

```
/CREATE-FILE FILE-NAME=<filename>,  
            SPACE=RELATIVE(PRIMARY-ALLOCATION=n, SECONDARY-ALLOCATION=m)
```



If the secondary allocation = 0, the paging file is not extended. As soon as the file size defined by the primary allocation is exceeded, the paging file is overwritten, and the existing content is discarded.

Name of the paging file

The name of the paging file can be defined in the start parameters (see [page 49](#)) with the statement

```
FILE PAGING=<filename>
```

If no file name is specified for the paging file, it is composed as follows:

```
S.OMS.tsn.yyyy-mm-dd.PAGE
```

tsn is the task number of the OMNIS task.

yyyy-mm-dd is the current date (year-month-day).



The temporary paging file is deleted once OMNIS has been terminated. It can no longer be evaluated nor printed.

3.6 Trace file

The trace file is used by OMNIS to write traces for better diagnostics during operation. Tracing of connection, transport and exit trace entries in the trace file can be activated and deactivated by the system service or the administrator using the TRACE command. For more information on the topic of diagnostics, refer to the "Functions and Commands" manual [1].

The following trace entries are written to the trace file when certain events occur:

- **Connection trace entries** log DCAM connections. A trace entry is written to the trace file when the following events occur:
 - a connection is set up, released or lost
 - there is a connection response from the transport system
 - there is connection information or a connection modification
- **Exit trace entries** log exit routines. A trace entry is written to the trace file when the following events occur:
 - "OMNIS calls exit routine"
 - "Exit routine calls OMNIS"
 - there is an exchange of information between exit routines exchanging parameters in a group

- **Transport trace entries** log DCAM messages. A trace entry is written to the trace file when the following events occur:
 - a message is sent via a DCAM connection
 - a message is received via a DCAM connection

Name of the trace file

The name of the trace file is defined by OMNIS when it opens a new trace file. A new trace file is opened when no trace was active and at least one of the 3 trace types is being activated. If a trace file is already open when another trace type is activated, the additional trace type is also written to the already open trace file. The name of the trace file is formed as follows.

`SYS.DIA.OMNIS.tsn.<progressive number>`

tsn represents the number of the OMNIS task.

<progressive number> represents a counter which OMNIS, starting at 1, increments by 1 each time a new trace file is opened.

Evaluating the trace file

The trace file can be edited and evaluated with the TEDDY program. Since this program is also used for the various trace files of the transport system (BCAM, DCAM), these trace files can also be merged by OMNIS. This gives the administrator or the system service a very good overview of the behavior of the entire system. The diagnostics can also be improved by OMNIS with the DUMP command (refer to the "Functions and Commands" manual [1]).



OMNIS can edit the trace file only when all traces are inactive and the file has therefore been closed.

3.7 Message file

All messages and HELP texts are maintained in the standard BS2000 message file *SYSMES.OMNIS.085*.

The message ranges are defined as follows:

0000 through 0499	Messages from the OMNIS interactive program
0500 through 0599	Messages from OMNISFC
0600 through 0699	Messages from OMNISLC
0700 through 0899	Messages from OMNISKD
0900 through 0999	Dummy messages for HELP texts
1000 through 6999	(reserved)
7000 through 7999	User messages from exit routines
8000 through 9999	(reserved).

All messages, explanations, and HELP texts are in both German and English; other languages can be incorporated using the standard tools; the prefix is *OMS*.

Irrespective of the system generation, the messages are output with 7-byte message codes.

The language in which the messages and HELP texts are issued is defined during system generation.

Console messages are issued with routing code *A*. The routing code is defined in the system message files, and can be altered if necessary.

The message file also includes help texts for the DCAM return codes. These can be identified by their prefix *OMY*.

3.8 Start-up file

The start-up file is a special command file that is processed immediately after OMNIS is started.

Only the *SET*, *MDEF*, *EXIT*, *PRIORITY* and *TRACE* commands are allowed in the start-up file.

The name of the start-up file can be chosen using the start parameter *STARTUP*; the default name is *OMNIS.STARTUP*.

Example of a start-up file

```
SET      NPASS=X'123456',DSTMAX=10,PTNMAX=3,PACMAX=30
SET      ACK=YES
EXIT     ABC,ACTION=CREATE,MODULES=(MOD1,MOD2)
SET      EXIT=ABC
MDEF     POP1,AC=C
MDEF     POP1,MSG=(AAAAAAA,35),D=N
MDEF     POP1,MSG=BBBBBBB,REPLY=(' .NO', '/CAN &TSN')
MDEF     POP1,MSG=CCCCCCC,D=N
MDEF     POP2,AC=C
MDEF     POP2,MSG=(AAAAAAA,35),D=N
MDEF     POP2,MSG=CCCCCCC,D=N
MDEF     POP2,MSG=DDDDDDD,D=N
MDEF     POP2,MSG=EEEEEEE,REPLY=(' .1')
```

NPASS is used to define a new administration password. *APASS* authorizes the administrator to issue this command. *SET NPASS* must always be specified, since OMNIS cannot be started without the administration password.

3.9 Text file

The text file is an ISAM file (key ISAMKEY, key length *KEYLEN=8*) which can be created using the OMNISKD statement *DECLARE-START-SEQUENCE*. The link name of the text file is *SOMSTEXT*. It contains a sequence of *LOGON/KDCSIGN* commands which would otherwise have to be entered at the terminal manually after a connection to a partner has been set up.

All records whose ISAMKEYs begin with the same *n* characters (= index group, $4 \leq n \leq 7$) constitute a *LOGON/KDCSIGN* sequence. The number of digits to be evaluated in the key is defined using the *TEXTKEY* operand of the start parameter *LENGTH* (*LENGTH TEXTKEY=n*). With the OMNISTC utility, text files can be converted to various key lengths, see [page 247](#).

OMNIS reads the records of a start sequence from the text file and, after connection setup, automatically sends them to the partner to which the start sequence is assigned.

The first record that is read is the one with a key whose first *n* digits were specified in the *DECLARE-PARTNER* statement. All subsequent records are then sent sequentially to the partner until a record is read which has a key whose first *n* digits differ from the specified key or the end of file is reached.

All messages must be contained in the file without a PAC.

Example of a text file with TEXTKEY=4

```
0001.00* KDCSIGN for partner UTM1
0002.00KDCSIGN UTM1
0002.01FORM1
0004.00* TIAM=partner TIA1
0005.00/.OMNIS LOGON OMNIS.OMNIS.C'OMNIS'
0005.01/ENTER-JOB FROM-FILE=OMNIS.ENTER
0005.02/SHOW-USER-STATUS
0005.03/START-PROGRAM FROM-FILE=$EDT
0006.00* Procedure for creating an OMNIS configuration file
0010.00/.OMKONF LOGON OMNIS.CONF.C'OMNIS'
0010.01/ASSIGN-SYSDTA TO-FILE=OMNIS.CON.IN
0010.02/ASSIGN-SYSLST TO-FILE=OMNIS.CON.LST
0010.03/SET-FILE-LINK LINK-NAME=OMNIS.CON.OUT,FILE-NAME=SOMSKD
0010.04/START-PROGRAM *(SYSLNK.OMNIS.085,OMNISKD,RUN-MODE=ADV)
0010.05/ASSIGN-SYSDTA TO-FILE=*PRIMARY
0010.06/ASSIGN-SYSLST TO-FILE=*PRIMARY
0010.07/PRINT-FILE FILE-NAME=OMNIS.CON.LIST
```

In this example the keys 2, 5 and 10 are valid *LOGON/KDCSIGN* sequences. The other records are merely comments.

In the *DECLARE-PARTNER* statement, the user can also specify whether *LOGOFF* is to be sent to the TIAM partner when the text file has been processed.

If OMNIS receives a format from a UTM or TIAM partner, text file processing is automatically interrupted. Text file processing is automatically resumed if messages are output in line mode.

The text file assigned to the (OMNIS) slave is used for indirect connections.

User-specific start sequences for OMNIS-MENU

An index key is assigned to each start sequence in the text file. The index key consists of a four-digit user number, a two-digit application number and a two-digit serial number.

User number				Appl. no.		Serial no.	
0	0	0	1	0	1	0	1
0	0	0	1	0	1	0	2
0	0	0	3	0	2	0	1
			⋮				
			⋮				
2	4	1	3	7	2	9	9

← max. 99 entries
per start sequence

Figure 3: Index key of the start sequence in the text file

The four-digit user number can be assigned when the user is defined, when the menu is defined or when the application is defined.

If a user number has been defined more than once, interpretation is performed according to fixed priorities. These priorities are defined with the *USER-NUMBER-PRIORITY* operand in the *DECLARE-APPLICATION* statement. Depending on the specifications in this operand, the following priorities apply to the evaluation of the user number and its use in the start sequences:

USER-NUMBER-PRIORITY=USER-MENU-APPLICATION (operand default)

1. Evaluation for definition of a user
2. Evaluation for definition of a menu
3. Evaluation for definition of an application

If no user number was specified at the time of user definition, the specification of the user number at the time of menu definition is checked. If no user number was assigned here either, a check is made at the time of application definition. If no user numbers have been assigned, no start sequences are processed.

USER-NUMBER-PRIORITY=APPLICATION-USER-MENU

1. Evaluation for definition of an application
2. Evaluation for definition of a user
3. Evaluation for definition of a menu

If no user number was specified at the time of application definition, the specification of the user number at the time of user definition is checked. If no user number was assigned here either, a check is made at the time of menu definition. If no user numbers have been assigned, no start sequences are processed.

The two-digit application number is assigned at the time of application definition. If no application number is defined, no start sequences are processed.

If a start sequence is defined, OMNIS-MENU generates the following in the *OPNCON* command:

```
@:OPNCON pac, ... , START-SEQUENCE=user-number|application-number,...
```

If no start sequence can be allocated, the *OPNCON* command is executed without start sequence parameters.

3.10 Libraries

Interface library

The *SYSLIB.OMNIS.085* interface library contains all of the OMNIS macros and the OMNIS copy elements.

OMNIS uses the macros to assemble/compile exit routines or evaluation programs for the logging file. Standard macros can be found in the library *\$TSOS.MACROLIB*. The macro libraries are not required on-line in order to run OMNIS.

OMNIS uses the copy elements to compile exit routines written in COBOL.

Module library

OMNIS dynamically loads exit routines from the user-created module library specified in the start parameter *MODLIB*.

The BS2000 call *BIND* is used to load these user modules.

3.11 Overview of files required by OMNIS-MENU

In addition to the files described starting on [page 23](#), OMNIS-MENU works with the following files.

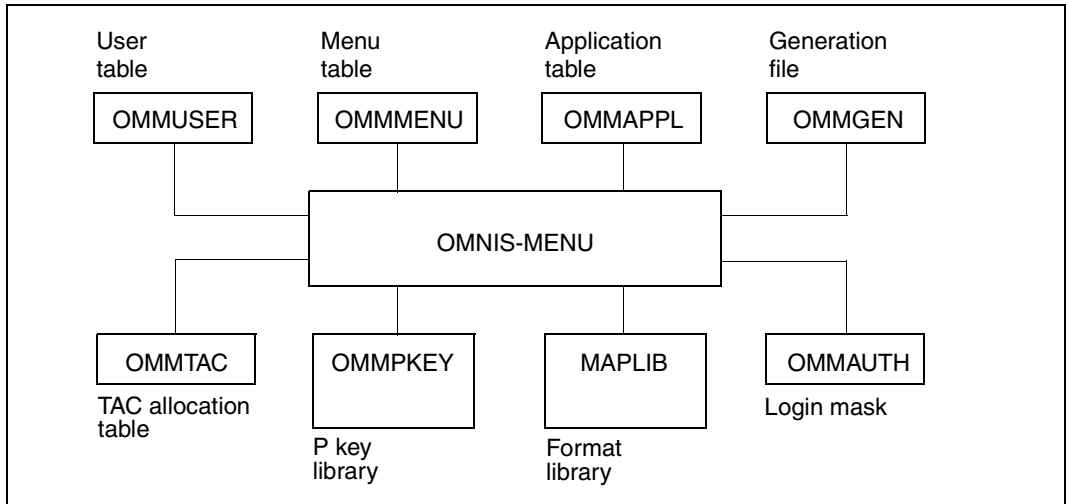


Figure 4: Files required by OMNIS-MENU

The file names and the name of the format library are freely selectable and can be assigned in the ENTER file on starting up the OMNIS-MENU with the *SET-FILE-LINK* command, e.g. as follows:

```

/SET-FILE-LINK FILE-NAME=filename1, LINK-NAME=OMMGEN
/SET-FILE-LINK FILE-NAME=filename2, LINK-NAME=OMMUSER
/SET-FILE-LINK FILE-NAME=filename3, LINK-NAME=OMMMENU,
    BUFFER-LENGTH=STD(SIZE=16)
/SET-FILE-LINK FILE-NAME=filename4, LINK-NAME=OMMAPPL
/SET-FILE-LINK FILE-NAME=filename5, LINK-NAME=OMMPKEY
/SET-FILE-LINK FILE-NAME=filename6, LINK-NAME=OMMAUTH
/SET-FILE-LINK FILE-NAME=filename7, LINK-NAME=OMMTAC
/SET-FILE-LINK FILE-NAME=format-library, LINK-NAME=MAPLIB
  
```

filename1...filename7 are fully qualified file names; *format-library* is the name of the OMNIS-MENU format library (*SYSFHS.OMNIS-MENU.085.E*). The user has the option of choosing between the German (*.D) and English (*.E) format libraries.

The associated files are created on initial installation of OMNISKD. If no file assignment is made via *SET-FILE-LINK* commands, OMNISKD creates the files *omm.gentab*, *omm.usertab*, *omm.menutab* and *omm.apptab* under the current user ID.

For reasons of uniformity, these predefined file names will be used in the following text in this manual.

OMNIS-MENU loads the module *MFHSROUT* dynamically from the library *\$TSOS.MFHSROUT*. If the module *MFHSROUT* is located in a different library, the latter must be assigned by means of the command:

```
/SET-FILE-LINK FILE-NAME=module-library, LINK-NAME=BLSLIB01
```

Generation file **omm.gentab**

File type ISAM, link name OMMGEN

This file contains global generation data. It does not have any direct links to the other files.

User table **omm.usertab**

File type ISAM, key USER-NAME, link name OMMUSER

File containing an entry for each user authorized to work with OMNIS-MENU. A user entry defines the identity of a user by means a user name, password and, where applicable, ID card information, and refers to an entry in the

Menu table **omm.menutab**

File type ISAM, key MENU-NAME, link name OMMMENU

This file needs to be created first using `/CREATE-FILE` (see example in [section "Using OMNISKD" on page 101](#)).

Each entry in the menu table is assigned to one or more users and defines all the applications to which access is possible via the user main menu of OMNIS-MENU, with names and selection options (K, F and where appropriate, P keys).

The primary and secondary allocation and the block size of this file must be a multiple of 16. Each application name refers to an entry in the

Application table **omm.appltab**

File type ISAM, key MENU-NAME PAC for local applications, or file type ISAM, key * PAC for global applications, link name OMMAPPL

The application table describes all the applications that can be selected via OMNIS-MENU. Applications can be defined globally, i.e. for all menus, or locally, i.e. for a particular menu.

Parameters for partners not within the scope of OMNIS-MENU can be predefined through declaration with OMNISKD. An entry contains all the data required for connection setup to this application and, where applicable, a reference to the set assigned to the application in the

P key library omm.pkeylib

File type PAM, link name OMMPKEY

This library contains sets of P key assignments, allocated to the individual applications by the administrator. The P key assignments are defined using OMNISKD. OMNISKD generates the PLUS input file (LINK=OMMPLUS). You use PLUS to generate the P key library from this PLUS file (LINK=OMMPKEY).

Login mask file omm.authtab

File type SAM, link name OMMAUTH

This file contains messages whose contents are displayed in the login mask. Note that information from the OMNIS bulletin file is suppressed by OMNIS-MENU. The file can be created with any editor (e.g. EDT). A maximum of 13 lines of 80 columns each can be displayed.

TAC allocation table omm.tactab

File type SAM, link name OMMTAC

The TAC allocation table specifies which application is activated in the event of an implicit application switch. It also defines the behavior associated with an implicit application switch. Also refer to the "Functions and Commands" manual [1].

The diagram below shows which files have linked contents.

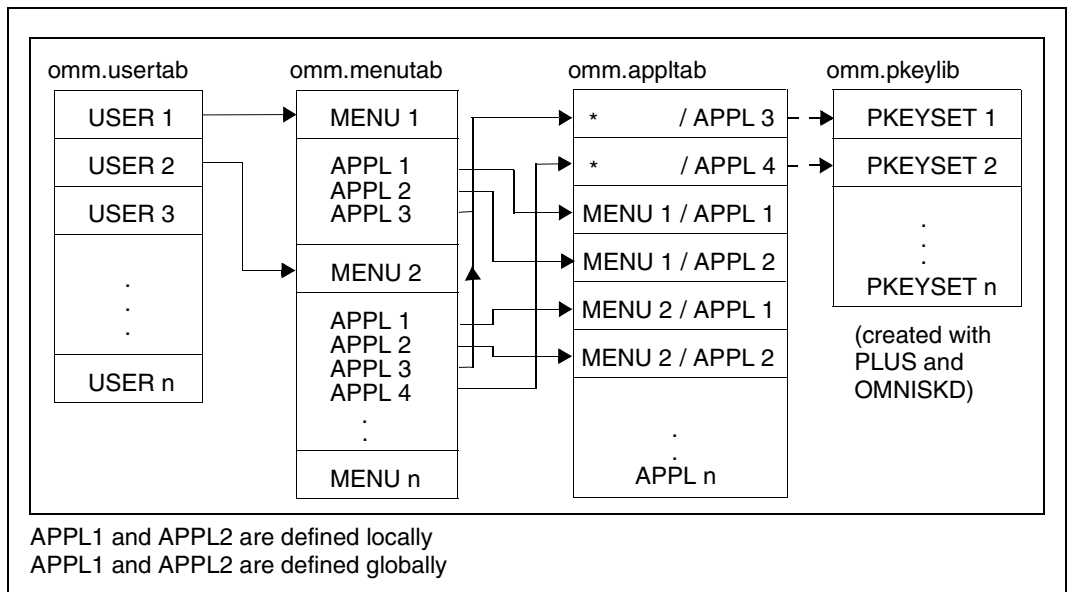


Figure 5: Overview of file links

4 Installation and startup

4.1 Creating the OMNIS files and user ID

Creating OMNIS files

The files from the release tape must be copied to each host on which OMNIS is to run, and a number of user-specific files must then be created (ENTER file, bulletin file, start-up file, command files).

The minimum requirement for starting up OMNIS is the existence of a start-up file containing the statement *SET NPASS=...* (see [page 39](#)).

The files need not be installed on hosts to which OMNIS makes (direct) connections, but on which it does not run.

The default value for the USER-ID is *\$OMNIS*.

OMNIS can be installed under any USER-ID.

The *SINPRC.OMNIS.085* procedure is used to adapt OMNIS to a specific USER-ID. The USER-ID is requested as input. If the USER syntax file *SYSSDF.OMNIS.085.USER* is to be adapted, you will need the software product SDF-A.

If OMNIS-MENU is already installed under a user ID (e.g. *\$RZOMNIS*), you must specify that user ID in the *OLDUSID* operand (e.g. *OLDUSID=\$RZOMNIS*).

Creating the USER-ID

OMNIS should execute as a real-time system with high priority. To attain optimum runtime performance, it is necessary to grant a number of privileges to the user ID under which OMNIS is started:

```
/ADD-USER USER-IDENTIFICATION=user-id,MAX-ALLOWED-CATEGORY=TP,  
          ACCOUNT-ATTRIBUTES=PARAMETERS(ACCOUNT=1,  
          MAXIMUM-RUN-PRIORITY=priority,PRIVILEGE=INHIBIT-DEACTIVATION)
```

OMNIS will, however, still run if *USER-IDENTIFICATION* has none (or only some) of the privileges shown above.

4.2 Starting OMNIS

If you want to use the OMNIS/OMNIS-MENU configuration data of an earlier version, please refer to the [section “Changing versions” on page 229](#).

You can call OMNIS either via the *START-PROGRAM* command or via *START-OMNIS*. For this the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* must be activated.



If you wish to start OMNIS using the *START-PROGRAM* command, and OMNIS is not installed under the user ID *OMNIS*, then the procedure *SINPRC.OMNIS.085* must be started when installing OMNIS. The required input is the name of the user ID (with \$) under which the library *SYSLNK.OMNIS.085* is located.

OMNIS is a non-privileged program and can execute under any user name, preferably as a batch process.

An Enter file is required for this purpose (see [page 27](#)).

As OMNIS can, in principle, remain loaded for any length of time, OMNIS processes should execute without being subject to any time limits. OMNIS should also be started with a high (fixed) priority:

```
/ENTER-JOB FROM-FILE=filename,  
RESOURCES=PARAMETERS(RUN-PRIORITY=80,CPU-LIMIT=NO)
```

If the CPU-TIME-LIMIT assigned to the OMNIS task is reached, OMNIS is terminated and no memory dump is output. No userdump is then written, but the OMS0008 message is output with Interrupt Weight X'80' (RUNOUT).

In the initialization phase, OMNIS always outputs error messages to SYSOUT. If no configuration file has been allocated, the messages OMS0005 and OMS0059 can be ignored.

4.2.1 Start parameters

The start parameters are used to tailor OMNIS to the given application (defining file names, specifying the level of operational security, optimizing memory requirements).

They are read in from SYSDTA immediately after OMNIS is initiated. OMNIS can process any number of start statements (up to an *END* statement or *EOF*). The order is not important.

APPLICATION

This statement defines resources for DCAM.

DATA

This statement defines the *DATA-ESCAPE-CHARACTER*, the *DATA-CHARACTER* and the behavior of variable substitution in the start sequences.

END

This the last statement and terminates processing of the start parameters.

FILE

This statement defines the names of files needed during an OMNIS run (startup file, bulletin file, logging file, configuration file, text file, module library for exit routines).

LENGTH

This statement defines the lengths of various memory areas (e.g. I/O areas, work areas for exit routines) and of the key for start sequences.

SECURITY

This statement makes it possible to specify restrictions for particular user groups in order to increase the operational security of OMNIS.

SETSW

This statement switches internal OMNIS switches on or off. These switches need to be set only in exceptional cases.

General syntax, statement format

The syntax of the start statements is described with the same notation and the same statement format as are used for the description of the OMNIS commands and the OMNISKD utility routine (see [page 109](#)).

Notation	Explanation	Example
UPPERCASE LETTERS	Uppercase letters indicate constants which must be entered exactly as shown.	LEVEL=LOW Enter following: LEVEL=LOW
lowercase letters	Lowercase letters indicate variables which the user must replace with a concrete value.	TEXTKEY=number Specify for example: TEXTKEY=3
{ }	Braces enclose alternative entries of which either one or none may be specified. Default values are <u>underlined</u> .	LEVEL={ HIGH MEDIUM <u>LOW</u> } Specify for example: LEVEL=HIGH, LEVEL=MEDIUM or LEVEL=LOW. If LEVEL is not specified, LEVEL=LOW applies.
[]	Brackets enclose entries which may be omitted.	H[IGH] Specify for example: HIGH, HI or H.

When commands are entered, there is **no** comma before the first operand.

APPLICATION - Define DCAM resources

The *APPLICATION* statement defines resources for DCAM.

Operation	Operands
A[PPPLICATION]	P[REFIX]=prefix [,I[SO]= { $\left. \begin{array}{c} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\}$]

PREFIX specifies three characters that OMNIS is to use to create names for auxiliary applications. The name consists of these three characters, followed by a five-digit number. *prefix* must begin with an alphabetical character. The default value is *OMS*.

PREFIX must be specified if OMNIS is loaded several times. Each OMNIS must have its own *PREFIX* which is different from all others.

ISO specifies whether the ISO application is opened.

NO Only the NEA application is opened.

YES Both the NEA and the ISO applications are opened.

DATA - Defining variable substitution

The *DATA* statement defines the *DATA-ESCAPE-CHARACTER* and the *DATA-CHARACTER*, as well as substitution of variables in the start sequences.

Operation	Operands
DATA	ESCAPE = { & # @ * \$ } CHARACTER = { & # @ * \$ _ } SUBSTITUTION = { <u>FIRST</u> ALL NO }]

ESCAPE defines the character with which a variable name must be prefixed.

CHARACTER defines the character with which a variable name ends.

SUBSTITUTION defines the records in a start sequence of a TIAM partner where S variables and job variables are to be substituted.

FIRST substitution occurs only in the first record.

ALL substitution occurs in all records.

NO no substitution occurs.

Detailed specifications on the concept of variable substitution can be found on [page 71](#).

For partner types other than TIAM, variable substitution occurs in all records, regardless of this parameter.

END - Terminate start parameters

The *END* statement terminates processing of start parameters.

Operation	Operands
END	

FILE - Define file names

The *FILE* statement defines the names of files needed for an OMNIS run.

Operation	Operands
F[ILE]	[B[ULLETIN]=filename] [,C[ONFIG]=filename] [,L[OGGING]=filename] [,M[ODLIB]=filename] [,P[AGING]=filename] [,S[TARTUP]=filename] [,T[EXT]=filename]

BULLETIN	defines the name of the bulletin file. The default name is <i>OMNIS.BULLETIN</i> .
CONFIG	defines the name of the configuration file. The default name is <i>OMNIS.CONFIG</i> .
LOGGING	defines the name of the logging file. The default name is <i>OMNIS.LOGGING.001</i> .
MODLIB	defines the name of the module library from which OMNIS is to fetch exit routines if needed. The default name is <i>OMNIS.MODLIB</i> .
PAGING	defines the name of the paging file in which OMNIS buffers messages when there is message congestion. The default name is <i>S.OMS.tsnn.year-month-day.PAGE</i> .
STARTUP	defines the name of the start-up file. The default name is <i>OMNIS.STARTUP</i> .
TEXT	defines the name of the text file containing the <i>LOGON/KDCSIGN</i> messages. The default name is <i>OMNIS.TEXT</i> .

LENGTH - Define length of areas

The *LENGTH* statement defines the lengths of various internal memory areas.

Operation	Operands
L[ENGTH]	[IO[AREA]=number] [,PW[ORK]=number] [,TE[XTKEY]=number] [,TW[ORK]=number]

- IOAREA** defines the length of the message receive buffer ($256 \leq \text{number} \leq 32767$). The default value is *4096*.
- It is only worthwhile specifying *IOAREA* if the particular implementation lets you define a maximum message length.
- If OMNIS receives messages which are longer than the receive buffer defined with *IOAREA*, these messages are truncated without warning.
- It should be noted that formats are generally longer than 1920 characters (screen size) since they include message headers and control characters.
- Outputs from OMNIS (e.g. for *HELP*, *MES*) are not limited by *IOAREA*.
- In particular with a large number of terminals and/or partners, the length of the *IOAREA* is decisive for paging and thus for the performance of OMNIS.
- PWORK** defines the length of the partner-specific work area for exit routines ($0 \leq \text{number} \leq 32767$). The default value is *256*. A value of *0* means that no partner-specific work area is needed.
- TEXTKEY** specifies the number of digits of a key to be evaluated in the text file for start sequences. A start sequence contains all the records in the text file whose keys begin with the same *n* characters (index group, $4 \leq n \leq 7$). The default value of *TEXTKEY* is *4*. The key values of the start sequences can be specified in the commands *DECLARE-PARTNER* and *OPNCON* (*START-SEQUENCE* operand). OMNISKD has no way of checking whether *TEXTKEY* matches the configuration during OMNIS operation. With OMNISTC, text files can be converted to various key lengths (see [page 247](#)).
- TWORK** defines the length of the terminal-specific work area for exit routines ($0 \leq \text{number} \leq 32767$). The default value is *256*. A value of *0* means that no terminal-specific work area is needed.

SECURITY - Define level of operational security

This statement makes it possible to specify restrictions for particular user groups in order to customize data protection in OMNIS.

Operation	Operands
SEC[URITY]	L[EVEL]= $\left\{ \begin{array}{l} L[OW] \\ H[IGH] \\ M[EDIUM] \end{array} \right\}$

LEVEL	defines the level of access restriction. The default value is <i>LOW</i> .
LOW	imposes no restrictions.
HIGH	<p>imposes restrictions affecting users and administrators. No access to BS2000 files is permitted, regardless of whether or not OMNIS is running under TSOS. This concerns the commands <i>FILE</i> and <i>MESSAGE</i>.</p> <p>All administration functions are restricted to the BS2000 console. The test mode is prohibited.</p> <p>Connection takeover and automatic connection re-setup are also forbidden.</p> <p>Even if the operand <i>LOGGING=NO</i> is specified, OMNIS commands and system messages are always logged in the logging file. If the logging function fails, OMNIS is terminated immediately.</p>
MEDIUM	<p>imposes restrictions affecting users. Access to BS2000 files is permitted only if administration authorization has been assigned. This concerns the commands <i>FILE</i> and <i>MESSAGE</i>. For administrators, these two commands are forbidden only if OMNIS is running under TSOS. Connection takeover and automatic connection re-setup are also forbidden.</p> <p>Even if the operand <i>LOGGING=NO</i> is specified, OMNIS commands and system messages are always logged in the logging file. If the logging function fails, OMNIS is terminated immediately.</p>



When written to the logging file, messages are truncated to a length of approx. 8000 bytes. The exact maximum number of bytes of net data depends on the volume of administration data. If a record is not completely logged, it will be reported in the administration data. *OMNISLC* can access this information when preparing the logging file and mark the record accordingly.

*Examples***ENTER file 1**

```
/LOGON
/START-OMNIS
LENGTH IOAREA=2800
LENGTH TWORK=60,PWORK=350
FILE BULLETIN=OM.X.BULLETIN,STARTUP=OM.X.STARTUP
FILE MODLIB=OM.R.EXIT
FILE LOGGING=OM.X.LOGGING
END
/LOGOFF
```

ENTER file 2

```
/LOGON
/ASSIGN-SYSDTA TO-FILE=OM.X.STARTPAR
/START-PROGRAM FROM-FILE=*M(SYSLNK.OMNIS.085,OMNIS)
/LOGOFF
```

File OM.X.STARTPAR

```
LENGTH IOAREA=2800
LENGTH TWORK=60,PWORK=350
FILE BULLETIN=OM.X.BULLETIN,STARTUP=OM.X.STARTUP
FILE MODLIB=OM.R.EXIT
FILE LOGGING=OM.X.LOGGING
FILE TEXT=OM.X.TEXT
END
```


SETSW - Set switches

This statement switches internal OMNIS switches. The switches need to be set only in exceptional cases.

Operation	Operands
SET[SW]	ON= (0...31)

ON Sets the specified switch(es). If several switches are to be set simultaneously, the switch numbers must be separated by commas.
Example: *SETSW ON=(0,1,15,31)*

Each *SETSW* command resets all previously set switches.

4.2.2 Multiple OMNISes

OMNIS can be loaded several times if necessary to distribute the load or functions. Various resources must be allocated exclusively:

- Each OMNIS started must have an application name which is unique within the system. The application name is defined in the ENTER file using the *SET-DCAM-APPLICATION-LINK* command.
- Each OMNIS which is started has a separate logging file (this is the only file that is opened for writing). The name of the logging file is defined using the start parameter *LOGGING* (see [page 53](#)).
- Each OMNIS which is started must be allocated a different prefix for dynamically generated applications. This prefix is defined using the start parameter *PREFIX* (see [page 51](#)).

In the initialization phase, OMNIS checks the uniqueness of all the above resources and terminates itself with an error message if one of them is already in use.

4.3 Notes on installing OMNIS-MENU

If you want to use the OMNIS/OMNIS-MENU configuration data of an earlier version, please refer to the [section “Changing versions” on page 229](#).

OMNIS-MENU is implemented as an OMNIS exit routine and requires OMNIS and the OMNIS-specific files in order to run.

The OMNIS administrator must issue the call

```
EXIT eac, ACTION=CREATE, MODULES=OMM
```

in the OMNIS startup file and assign it by means of

```
SET EXIT=eac
```

where *eac* is the exit address code, comprising up to four alphanumeric characters.

If OMNIS-MENU is already loaded, it may not be dynamically unloaded and subsequently reloaded.

In order to permit user-specific start sequences, the startup file must specify that start sequences are allowed in the *OPNCON* command by including the following line:

```
SET OPNSTART=YES,...
```

The index key of a start sequence consists of a four-digit user number and a two-digit application number. Up to 99 entries per start sequence can be specified in the OMNIS text file.

The size of the OMNIS-DST work area must be ≥ 250 bytes. This is provided for in the OMNIS start procedure in the following way:

```
LENGTH TEXTKEY=6,TWORK=250,...
```

The first 250 bytes of the OMNIS-DST work area must not be written to be any other exit routine.

If no *CALL-KEY* is defined, OMNIS interprets any K key press as a call. It is thus necessary to define an unassigned K key as *CALL-KEY*. This can be done with the *DECLARE-TERMINAL* command in the OMNIS configuration file (terminal-specific) or with the *SET* command (global) in the startup file.

If OMNIS-MENU is to be accessible only via terminals that have been specified with *DECLARE-TERMINAL* in the configuration file by the OMNIS administrator, the parameter

```
SET OPNCON=DECLARED
```

must also be specified in the *SET* command in the startup file.

For OMNIS-MENU to operate, the OMNIS message file must be defined. If the OMNIS message file is not available, OMNIS-MENU deactivates itself when it is loaded.

In order to call OMNISKD, OMNISLC, OMNISFC and OMNISMD not only with the *START-PROGRAM* command but also with *START-OMNISKD*, *START-OMNISLC*, *START-OMNISFC* or *START-OMNISMD* respectively, the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* must be activated.

Automatic connection setup

To set up connections to applications automatically, the appropriate entries must be made in the OMNIS configuration file.

Use of parallel exits

When parallel Cobol exits are used, care should be taken that the OMNIS call is entered as follows in Cobol:

```
CALL OMSCALL USING OMEPAR MESSAGE DSTWORK PTNWORK VTSUCB
```

or as follows in Columbus Cobol:

```
@PASS OMSCALL : TYP=E: USING OMEPAR MESSAGE DSTWORK PTNWORK VTSUCB
```

For a detailed description of the interface, refer to [page 249](#).

Changing the layout of the OMNIS-MENU login mask

If you want to modify the login mask, you can create a suitable file with an editor. This file may be of the SAM or ISAM type. The file is assigned during the start procedure with the statement:

```
/SET-FILE-LINK FILE-NAME=filename, LINK-NAME=OMMAUTH
```

The file can be modified during ongoing operation and goes into effect immediately, i.e. when the next user logs in. The first 13 lines with 80 columns are read.

Changing mask layouts with IFG

The IFG format library is included in the delivery package. This enables the administrator to modify all OMNIS-MENU masks. When doing this, note the following points:

- The format name must not be changed.
- The layout of the program transfer area must not be changed (e.g. sequence of fields, lengths, etc.).

5 Administration

This chapter begins by describing the tasks involved in OMNIS administration, the administration concept and the requisite commands. The [section “Special OMNIS-MENU administration requirements” on page 68](#) describes special administration functions for OMNIS-MENU.

For information on automatic system operation based on message tables, refer to the "Functions and Commands" manual [1].

5.1 Tasks of the administrator

The main tasks of the OMNIS administrator are:

- to provide the necessary system environment and
- to prepare and monitor OMNIS operation in accordance with data protection requirements

Each of these tasks involves a number of possible activities, which are listed below:

- **Providing the necessary system environment**
 - consult with the BS2000 system administrator
 - perform special hardware and software installations
- **Preparations for OMNIS operation**
 - initialize and (if necessary) create the files required by OMNIS
 - assign file names
 - create user IDs where necessary
 - tailor OMNIS to the relevant implementations, e.g. optimize memory requirements and user work areas
 - define the configuration
 - consult with the administrators of the OMNIS partners (regarding the software prerequisites)

- consultation with the OMNIS administrators of remote systems with which communication via OMNIS is to take place
- define the level of operational security
- tailor to the specific installation
- **Data protection measures**
 - assign passwords
 - specify connection options to OMNIS
 - specify OMNIS functions available to the users
- **Monitoring and controlling operation**
 - provide user information (bulletins, messages)
 - manage central resources
 - define various operating parameters (default values, behavior after a loss of connection)
 - modify the configuration while OMNIS is running
 - cancel connections
 - support users in the event of a problem
 - generate online diagnostic information (using detailed information not available to the user; in extreme cases, intervene with AID commands)
 - generate and make available offline diagnostic information (analysis of the logging files)
 - start and terminate OMNIS where necessary

5.2 Administration concept

The administration of OMNIS comprises the following tasks:

Definition of a configuration

Using the OMNISKD utility routine, the administrator can define a connection structure. This, on the one hand, simplifies operation for the user and, on the other hand, provides a certain degree of access control in that the use of OMNIS can be restricted to declared partners and terminals. When declaring partners and terminals, it is also possible to specify that the operand settings cannot be modified by command. For certain types of connections, a declaration is (for technical reasons) mandatory (e.g. for OMNIS-OMNIS connections, multiplex connections, connections to BS2000 consoles).

The OMNIS configuration can be modified with the *CONFUPDATE* command while OMNIS is running.

Monitoring and control of operation

For monitoring and control of operation, the administrator has access to certain additional (“privileged”) functions, all of which require a knowledge of the administration password.

Administration password, centralized and decentralized administration

The administration password must (always) be defined in the startup file but can be changed during operation. Providing *SECURITY LEVEL=HIGH* was not specified when OMNIS was started, anyone who knows the administration password can administer OMNIS from any OMNIS terminal. If *SECURITY LEVEL=HIGH* is set, administration is possible only from the console.

With *SECURITY LEVEL≠HIGH*, both centralized and decentralized administration of OMNIS are thus possible.

The administration password can be entered in any command that contains a privileged function. The password specified is effective only for the duration of one command; it must be entered again for any subsequent administration command.

A terminal can also be defined as the administrator terminal (with *OPTION ADM=YES* or *DECLARE-TERMINAL ADM=YES*). All privileged functions, with the exception of the *END* command, can be called up from such a terminal without having to enter the administration password.

Administration command SET

The actual administration command is the *SET* command. It can be used to

- define or alter the administration password,
- define maximum values which may not be exceeded when using OMNIS (maximum number of partners *PTNMAX*, terminals *DSTMAX* and partners per terminal *PACMAX*),
- define default values which apply when a user does not specify explicit values,
- determine whether or not the connection of undeclared terminals to OMNIS is possible or not. This makes it possible to restrict the use of OMNIS to terminals specified by declaration.
- activate or deactivate the *REAL TIMER*.

Hierarchy of command scopes

As the default option, the partner-specific scope (*PARAM*, *OPNCON*, *DECLARE-PARTNER* commands) has the highest priority, followed by the terminal-specific scope (*OPTION* and *DECLARE-TERMINAL* commands), and then the global scope (*SET* command).

The *PRIORITY* command can be used to change the priority of the individual scopes of the *LOGGING*, *EXIT* and *HOLD=AUTO* command operands, so that the following setting applies: *SET* before *PARAM* (*OPNCON*, *DECLARE-PARTNER*) before *OPTION* (*DECLARE-TERMINAL*). *PRIORITY* can also be used to specify whether message tables and exit groups may also be generated and administered by the user (see also the "Functions and Commands" manual [1]).

Test mode, online diagnostics

A further prerequisite for the administration activities, in particular for online diagnostics, is provided by the test mode, which can be switched on with the *OPTION TESTMODE=YES* command. If *SECURITY LEVEL=HIGH* is set, the test mode is prohibited.

The following commands, which are otherwise rejected, are permitted in test mode:

- *CMD* (executes any BS2000 command),
- *DUMP* (generates a dump without aborting OMNIS), and
- *TRACE* (switches the trace on/off).

It is also possible to reassign *SYSLST* and/or *SYSOUT* in test mode.

In addition to the above, the administrator has access, in test mode, to special, detailed information for diagnostic purposes (*INF* command).

Logging, offline diagnostics

The logging file can be switched during operation with the *CHANGELOG* command, so that the old logging file is available for offline analysis with the OMNISLC utility routine. If *SECURITY LEVEL=HIGH* is set, the operand *APASS* is rejected, which means that *CHANGELOG* cannot be executed or can be executed only from the BS2000 console.

The above points are only the most important aspects of the administration concept. A summary of all the privileged functions (commands and command operands) can be found in the following section.

5.3 Administration commands at a glance

This section gives you an alphabetical list of the commands and command operands reserved for use by the administrator. A full description of each command can be found in the “Functions and Commands” manual [1].

CANCEL

The *CANCEL* command terminates a connection. Unlike the *CLSCON* command, which is available to OMNIS users as well, “foreign” (= another OMNIS terminal's) connections can be terminated with *CANCEL*.

CHANGELOG

The *CHANGELOG* command is used to switch to a new logging file during operation. The old logging file is closed and is thus available for offline processing.

CMD

The *CMD* command is only allowed in test mode. It executes a freely selectable BS2000 command and is intended primarily for the BS2000 debugging commands *DISPLAY* and *MOVE* to output and, if necessary, modify internal OMNIS data.

CONFUPDATE

The *CONFUPDATE* command causes OMNIS to read in a new configuration file created with OMNISKD. This permits the OMNIS configuration to be modified while OMNIS is running.

DUMP

The *DUMP* command is only allowed in test mode. It generates a dump without terminating OMNIS. Note, however, that OMNIS is blocked during output of the dump and will not accept any messages during this period.

END

The *END* command terminates OMNIS.

FILE

Specifying the operand *DISPLAY=YES* causes the command file to be displayed at the terminal.

INF

Information about foreign partners, terminals and users is available only to the administrator.

If test mode is active, the information displayed by the commands *INF PARTNER*, *PID=xxx* and *INF TERMINAL*, *TID=yyy* differs from that displayed in normal mode in that it consists of detailed information required for diagnostic purposes.

MESSAGE

The *ALL* operand is reserved for use by the administrator:

MESSAGE ALL can be used to send a message to all OMNIS users. Since OMNIS does not accept messages during execution of the *MESSAGE* command, it should be noted that *MESS ALL* can temporarily block OMNIS operation for all users.

OCCUPY

The administrator can take over partners (terminals) without knowing their *OPASS*.

OPTION

With the operand *ADM=YES*, a terminal can be defined as the administrator terminal while OMNIS is running. All privileged functions, with the exception of the *END* command, can then be called from this terminal without having to enter the administration password.

TESTMODE=YES (SECURITY LEVEL≠HIGH) can be used to activate test mode. This is a prerequisite for the output of detailed information about foreign objects (*INF* command) and for the *DUMP*, *TRACE* and *CMD* commands.

PRIORITY

The *PRIORITY* command defines whether the default command scope (*PARAM* before *OPTION* before *SET*) or *SET* before *PARAM* before *OPTION* is to apply for the *EXIT* and *LOGGING* operands. It can also be used to define whether the setting selected by the administrator or by the user is to apply for *HOLD=AUTO*, and whether the *EXIT* and *MTAB* commands are to be reserved for the OMNIS administrator or whether every OMNIS user can issue these commands.

SET

The *SET* command is used to set global operating parameters.

It is used to

- define or modify the administration password (*NPASS*, *APASS*),
- define maximum values which may not be exceeded when using OMNIS (such as the maximum number of partners *PTNMAX*, terminals *DSTMAX* and partners per terminal *PACMAX*),
- define default values which apply when a user does not specify explicit values,
- define exit groups and message tables for terminals (*EXIT*, *MTAB*),
- control logging of messages sent via OMNIS in the logging file and control the output of these messages on a printer (*LOGGING*, *INPUT-LOGGING*, *OUTPUT-LOGGING*, *HCOPY*),
- define behavior if a connection to a terminal is lost (*HOLD*),
- define whether only declared terminals may sign on to OMNIS or undeclared terminals as well (*OPNCON=DECLARED/FREE*),

- define whether or not the operand *START-SEQUENCE* may be specified in the *OPNCON* command (*OPNSTART=YES/NO*),
- define the CALL and BREAK keys and the K key for switching from formatted mode to OMNIS command mode (*KPAC*) and activate/deactivate screen overflow control (*ACK=YES/NO*),
- define the output format and color (*PAC, LINE25, DISPLAY-MODE*),
- define whether or not the audit trace is to execute (*AUDIT=ON/OFF*).

TRACE

The *TRACE* command is used to switch on and off the trace for connection, transport and exit routine trace entries. The command is only permitted when test mode is active.

5.4 Special OMNIS-MENU administration requirements

Allocation of user authorizations for OMNIS-MENU, assignment of the menu-selectable applications and compilation of all the data required for connection setup and for ongoing operation are carried out offline by the administrator with the OMNISKD program.

At the level of the OMNIS-MENU user interface, the administrator functions **UNLOCK* (=unlock a terminal) and **UPDATE* (=dynamic configuration modification) are available.

For instruction on what to do in the event of errors, refer to the "Functions and Commands" manual [1].

5.4.1 Unlocking a terminal

A user whose *Application Menu* contains the administrator function *UNLOCK and who selects this, has the *Unlock Terminal* mask displayed:

```

Function: OMMFUNL          O M N I S - M E N U          11:41:42/2002-01-22
User   : IDEFIX          Unlock Terminals          STATI019  ATPC3EJC
-----
Max. permissible number of field entry attempts at login: 05

                                Terminal / Processor
                                -----
                                x  STATI020  ATPC3EJC
                                |  STATI021  ATPC3EJC

Command  0  (0/+/-/+/+/-)
-----
F1=Help SEND=Send K12(P12)=Go back

```

Figure 6: Unlocking a terminal

This mask displays a list of locked terminals. To cancel a lock, the relevant terminal must be marked with an *x*. The actual unlocking is performed by entering the command *0*.

Meaning of the fields

Command:

- 0 Unlock terminals marked with *x*.
- + Move one page forward in list.
- Move one page backward in list.
- ++ Move to end of list.
- Move to beginning of list.

5.4.2 Dynamic modification of configuration

A user whose *Application Menu* contains the administrator function **UPDATE* and who selects this has the *Update Configuration* mask displayed:

```

Function: OMMFADM      O M N I S - M E N U      11:43:17/2002-01-22
User   : IDEFIX      Update Configuration    STATI019  ATPC3EJC
-----
Do you really want to update the OMNIS-MENU configuration (Y/N): Y

Notice:  Active terminals have to be disconnected and reconnected
         to OMNIS-MENU to get the new configuration.

         The global parameters take effect immediately for all
         terminals.

-----
SEND=Send K12(P12)=Go back

```

Figure 7: Update Configuration

With this mask, modified generation data can be included in the OMNIS-MENU configuration while the program is running. Entering *Y* (default) executes the update operation; entering *N* returns the user to the application menu without carrying out an update.

The global generation parameters take immediate effect. The other generation parameters do not take effect until the user has logged out of and back into OMNIS-MENU.

Once an OMNIS-MENU update has been completed, an OMNIS *CONFUPDATE* is performed as well.

5.5 Variable concept

OMNIS configuration files can be created more easily and flexibly using variables. Variable names can be specified in definition reports in the OMNISKD input file and in OMNIS commands. During runtime, these variables are substituted for the current value when the corresponding configuration entry is processed.

Variable names from a predefined list, e.g. the names of S variables or job variables can be used. The places where these variables can be used are predefined.

Predefined variables

The predefined variables describe OMNIS/OMNIS-MENU terms whose value is defined at execution time:

Variable name	Area of application	Meaning
USER	OMNIS-MENU	user name
MENU	OMNIS-MENU	menu name
APPL	OMNIS/OMNIS-MENU	name of the simulated terminal
PASS	OMNIS-MENU	user password
TERM	OMNIS/OMNIS-MENU	terminal name
PAC	OMNIS/OMNIS-MENU	partner address code
OMNIS	OMNIS/OMNIS-MENU	application name
OMISO	OMNIS/OMNIS-MENU	ISO application name

S variables (SDF-P)

The current value of S variables is used at runtime. You can freely choose the name of S variables, but the following constraints apply:

- The names of predefined variables may not be used.
- The name length is restricted to the maximum length of the operand value for which the variable is to be used. Since an introductory character is required if the variable is to be recognized (see [page 73](#)), the following maximum length is allowed: $n-1$.
 n is the maximum length of the operand for which the variable is to be used.
- Compound S variables can also be used both as structured variables and as list variables and array variables. A single structure element or one element of the list or of the array, however, must be addressed. The whole structure, list or array may not be used, since it would then not be possible to insert the name value.

In order to use structured variables, it is necessary to change the default character '.' (period) used to identify the end of a variable name. This can be defined in the start parameter statements for OMNIS.

- The S variable must be created within the OMNIS start procedure or globally for the OMNIS task.
- Variable values are case sensitive; no uppercase conversion is performed by OMNIS.

Job variables (JV)

The current value of job variables is inserted at execution time. You can freely choose the name of job variables, but the following constraints apply:

- The names of predefined variables may not be used.
- The name length is restricted to the maximum length of the operand value for which the variable is to be used. Since an introductory character is required if the variable is to be recognized (see [page 73](#)), the following maximum length is allowed: $n-1$.
 n is the maximum length of the operand for which the variable is to be used.
- You can address job variables via link names, but the normal construct **jvlinkname* must be used. If you do this, a further character is lost from the maximum length of the variable name, making the following formula valid for the maximum length for the link names of job variables: $n-2$.
 n is the maximum length of the operand for which the variable is to be used.
- Variable values are case sensitive; no uppercase conversion is performed by OMNIS.
- Access protection for job variables can be used, but the OMNIS task must have the relevant permissions e.g. by declaring the password in the OMNIS start parameters or via the OMNIS command *CMD*.

Substitution sequence

The variables are always substituted in the following sequence:

- predefined variables
- S variables
- job variables

S variables and job variables in TIAM partner start sequences

For TIAM partners, it was possible to specify S variables and job variables in the start sequences, in which case substitution was performed when the command was executed by SDF-P. No variables could be specified for the LOGON procedure of the TIAM partner (*LOGON* command or *SET-PROCEDURE-OPTIONS*, which is normally the first record of the start sequence), since SDF-P did not at that point perform substitutions.

Variable substitution in OMNIS means that this constraint no longer applies. S variables and job variables can be used in the LOGON procedure. When OMNIS performs the substitution, the variables are scoped to the OMNIS start procedure, the OMNIS task or the ID under which OMNIS was started. They are not scoped to the task or the ID of the OMNIS partner that was started.

OMNIS substitutes S variables and job variables for TIAM partners by default only in the first record of the start sequence. This can be changed by making explicit entries in the OMNIS start parameters as follows:

Substitution in the first record only:

```
DATA SUBSTITUTION = FIRST
```

Substitution in all records:

```
DATA SUBSTITUTION = ALL
```

No substitution in the start sequences:

```
DATA SUBSTITUTION = NO
```

The predefined variable names are identified in all records in the start sequence and are substituted.

For partner types other than TIAM, substitution occurs in all records, regardless of this parameter.

Using and specifying variables

Variables can be used in the following commands:

Command	Parameter
OPNCON	PRONAME PTNNAME
OPTION	HCOPY (DEVICE,FORM)
SET	HCOPY (DEVICE,FORM)

Variables can be used in the following OMNISKD statements:

OMNISKD statement	Parameter
DECLARE-APPLICATION	PTAST RESTART-MESSAGE CMMSG/LMSG BERID PRONAME PTNNAME
DECLARE-MENU	HCOPY (DEVICE,FORM)
DECLARE-PARTNER	CMMSG/LMSG BERID PRONAME PTNNAME
DECLARE-START-SEQUENCE	RECORDS
DECLARE-USER	HARDCOPY-DEVICE HARDCOPY-FORM

The variable name must be prefixed with the *DATA-ESCAPE-CHARACTER*. The default *DATA-ESCAPE-CHARACTER* is ' & ' (ampersand). Doubling the *DATA-ESCAPE-CHARACTER* cancels its effect as the start character of a variable name. The repeated *DATA-ESCAPE-CHARACTERS* are then replaced by one *DATA-ESCAPE-CHARACTER*.

The variable name must end with one of the characters shown below:

DATA-CHARACTER	default value: . (period)
DATA-ESCAPE-CHARACTER	default value: & (ampersand)
b	blank
(open bracket
)	close bracket
,	comma
'	single quote
=	equals sign

A *DATA-CHARACTER* that follows the variable name is automatically removed when the variable name is identified and processed. If a *DATA-CHARACTER* is to follow the value of the variable once the variable has been substituted, the *DATA-CHARACTER* must be doubled.

Examples

The following examples use the default values for *DATA-ESCAPE-CHARACTER* and *DATA-CHARACTER*.

Entry in definition record	Value after substitution	Comment
RECORDS='.&USER LOGON OMSE,1'	'&LEI LOGON OMSE,1'	<i>LEI</i> = value of the <i>USER</i> variable
RECORDS='SH-FI &USER..FILE'	'SH-FI &LEI.FILE'	
RECORDS='SH-FI &USER.FILE'	'SH-FI &LEIFILE'	
RECORDS='SH-FI &USERFILE'	'SH-FI &USERFILE'	<i>&USERFILE</i> name is not a predefined variable name

Specifying *DATA-ESCAPE-CHARACTER* and *DATA-CHARACTER* in the OMNIS start parameters:

```
DATA ESCAPE,DATA CHARACTER
```

The beginning of a variable is identified by default with the '&' sign (ampersand), the end of a variable with the '.' (period), among other special characters. These default values can be changed by means of entries in the OMNIS start parameters:

```
DATA ESCAPE = ['&' | '#' | ',' | '*' | '$']
```

```
DATA CHARACTER = ['&' | '#' | ',' | '*' | '$' | '.']
```



When you enter the OMNIS start parameters in a procedure, the '&' character must be entered twice: '&&'.

When using structured S variables, the default value '.' must be changed because the period is used for specifying a structure element.

When using lists and arrays as S variables, the hash character '#' may not be used.

When using link names for job variables, the asterisk '*' may not be used.

Substituting variable names

When processing a parameter, the variable is substituted for the variable value at runtime. Substitution of the variables affects only the current processing of the parameter. In the data structures themselves, the parameter remains entered in the form of the variable name; when it is processed again, the value will be substituted. Renewed evaluation of variables is necessary for S variables and job variables. These variables can be changed during an OMNIS run, thus allowing them to be re-evaluated and substituted every time the parameter is used.

&APPL for the PTAST parameter

The *&APPL* variable can *only* be used for the *PTAST* parameter if *APPNAME* is specified in the *DECLARE-APPLICATION* command, but is not specified in partially qualified form.

If *APPNAME* has not been specified or is partially qualified, OMNIS generates a unique application name at the start of the application. The *PTAST* parameter is processed, however, when a user logs on, before the application is started.

Error situations

Variables cannot be substituted

- This can occur for predefined variables when OMNIS is used (without the OMNIS-MENU exit) if a variable is used that describes an OMNIS MENU term.
- This can occur for S variables and job variables if they have not been declared, or have not been assigned a value.

If this is the case, the OMNIS message OMS0263 will be output to *SYSOUT*, *SYSLST* and the terminal. The parameter is processed further without variable substitution.

Variable substitution produces parameters that are syntactically incorrect

During variable substitution, no special check is performed, and therefore no special error messages are issued. During further processing, the normal error message for a parameter that is syntactically incorrect is issued.

Variable substitution exceeds the maximum length of the parameter

This situation is identified when variables are substituted because otherwise it could lead to a memory overflow. The OMNIS message OMS0264 is output to *SYSOUT*, *SYSLST* and the terminal. The parameter is processed further without variable substitution.

Security aspect



Using variables in parameters that require passwords e.g. *CMSG/LMSG* or *BERID*, can lead to problems with regard to security, since passwords are stored in encrypted form in the OMNIS configuration files for security reasons. If variables are specified for these parameters, the value of these variable names but not the value of the variables is encrypted. Therefore, the variable value (e.g. the set of OMNIS-MENU user names) is not encrypted and hence possibly open to public view.

If security is an issue in an OMNIS application, you should avoid using variables for the *CMSG/LMSG* and *BERID* parameters.

OMNISKD check for MUX partners and indirect partners

The use of variables affects checks on the configuration of MUX partners and indirect partners. OMNIS makes the following checks after variable substitution:

- When partners such as these are declared, *OMNISKD* checks whether the appropriate *DECLARE-MUX* or *DECLARE-OMNIS* statements are available. If this is not the case, an error message to this effect will be issued and the partner entry will not be generated.
- If the host name or partner name contains variables, *OMNISKD* cannot perform a check, since no variables are allowed in the *DECLARE-MUX* or *DECLARE-OMNIS* statements.
- If one of the possible *DATA-ESCAPE-CHARACTERS* ('&', '#', '@', '*', '\$') occurs in the host name or partner name, *OMNISKD* does not perform a check either. The *DATA-ESCAPE-CHARACTER* that is currently set is not known at the time *OMNISKD* is run.

6 Data protection

To ensure secure operation with OMNIS, actions are required in a number of different areas, which are described below. The term "user" in this description is to be equated with "non-administrator".

6.1 Partitioning

Only the OMNIS administrator has the option of displaying files on the screen (*MES INPUT=, FILE DISPLAY=YES*).

A user has no way of requesting information about other users' resources (partners, terminals).

A user has access only to his or her own data in the logging file, not to the data belonging to other users.

6.2 Access to OMNIS

OMNIS may run under any user ID. It does not have to run under TSOS, nor is this recommended.

The ID under which OMNIS runs must be protected with a password, since some of the files required by OMNIS may contain information relevant to security, e.g. passwords (see [page 83](#)).

For the same reason, the files required by OMNIS should be created with NOSHARE under the user ID under which OMNIS is started.

The owner of the user ID under which the files required by OMNIS are located is the same as the OMNIS administrator.

All OMNIS files must be protected against unauthorized access by means of the appropriate BS2000 facilities.

OMNIS can be protected with a password against use by unauthorized persons. This password is defined in the ENTER file and must be specified when signing on to OMNIS.

The use of OMNIS can be restricted to certain terminals predefined by declaration by the administrator.

6.3 Access to the partners of OMNIS

The partners generally protect themselves by means of an authorization mechanism (e.g. *LOGON*, *KDCSIGN*, possibly an ID card or chip card). The use of OMNIS does not circumvent this authorization check.

Start sequences stored in the text file must not contain *LOGON* and *KDCSIGN* passwords. File passwords, on the other hand, are permitted.

Access to the partners can be further restricted by declaring a configuration where only declared connections are permitted.

Without administration authorization, only partners for whom an *OCCUPY* password (*OPASS*) has been defined can be taken over with *OCCUPY*.

The administrator can optionally set up further command restrictions for security purposes by means of the start parameter *SECURITY* (see [page 55](#)).

For secure operation, OMNIS can be started only under a user ID which possesses the network management privilege. All application names must begin with the character "\$".

6.4 Exit routines and security

Exit routines have access to all confidential information present in OMNIS. They must therefore be subjected to strict checks before being entered in the module library.

Exit routines also offer the possibility of incorporating and implementing additional security measures in programs. With these routines, it is possible, for example, to suppress the logging of passwords which are unknown to OMNIS.

The OMNIS module library itself should be protected against unauthorized access using the appropriate BS2000 facilities.

6.5 Diagnostic information

A number of diagnostic options are available to the administrator for on-line OMNIS diagnostics. These include:

DUMP command

With the *DUMP* command, the administrator can create a memory dump without terminating OMNIS. Operation of OMNIS is blocked during the dump output. The memory dump is saved in a dump file under the login being used by OMNIS. This option is therefore an easy way of creating diagnostic documentation about OMNIS errors (particularly those occurring sporadically).

The *DUMP* command can also be defined for a specific OMNIS message, i.e. when a specific message and thus a specific event occurs, a memory dump of OMNIS is created. The OMNIS message is specified by its message number. The dump is only made the first time a message occurs for a user.

Dumps do not contain internal OMNIS passwords (*APASS*, *OPASS*, *CMSG/LMSG*, *CPASS/LPASS*). Messages which contain passwords (if, for example, a *LOGON* command is sent to a TIAM partner without blanking of the password) may, however, be contained in dumps.

Runtime logs (*SYSLST*, *SYSOUT*) do not contain security-relevant information.

OMNIS dumps must be stored under the OMNIS user ID, since the message areas of an OMNIS dump may contain information relevant to security. In addition, the OMNIS load module must be identified as "execute-only". If OMNIS is aborted with *DUMP*, not all information may be contained in the logging file or trace file. The dump should thus be regarded as a supplement to the logging information or the trace file and kept for diagnostic purposes. *DUMP=YES* must always be set for the OMNIS task, since the dump will otherwise be suppressed.

As additional information for a dump, an insert in the message *IDA0N51* shows the OMNIS message number and the address at which the dump was requested.

Logging file

When logging is active, a log of the dialog between OMNIS partners is written to the logging file. Administrators can activate/deactivate logging and define the extent and time of logging with the *SET* command. They can, for example, specify that all messages entered at the terminal (*INPUT-LOGGING*) and/or those arriving from the partner (*OUTPUT-LOGGING*) are to be logged.

Administrators can also switch between logging files at any time, and view these files on the screen with the *TURN* command. The logging file can be edited with the *OMNISLC* utility and then further edited or printed with editors.

Trace

A trace entry is written to a trace file when a certain event occurs, thus enabling diagnostics to be performed on specific program behavior or a malfunction in an application. The following trace options are provided for diagnostics on OMNIS operation:

- **connection traces**, which log every connection setup/cleardown/loss, connection responses from the transport system and each item of connection information or connection modification with a trace entry.
- **transport traces**, which log each transmission and receipt of a message with a trace entry. The transport trace can be activated either for all OMNIS objects or for individual objects.
- **exit routine traces**, which log the behavior of OMNIS when exit routines are used with a trace entry. Trace entries are written when "*OMNIS calls exit routine*" or "*Exit routine calls OMNIS*" occurs, or when several exit routines act as a group.

Traces are activated/deactivated by the administrator using the *TRACE* command. The administrator can also display detailed information about the current trace settings with the *INF* command.

6.6 Passwords

The following files required by OMNIS may contain passwords:

- the text file (passwords in LOGON/KDCSIGN sequences),
- the startup file (administration password *APASS*; this is mandatory), and
- the logging file (the entire dialog of a terminal, including any passwords entered).

All passwords contained in the configuration file are encrypted.

The following measures are therefore recommended in order to maintain data protection:

- The user ID under which OMNIS runs must be protected with a password. The owner of the user ID under which the files required by OMNIS are located is the same as the OMNIS administrator. For secure operation, OMNIS can be started only under a user ID which possesses the network management privilege. All application names must begin with the character "\$".
- All OMNIS files must be protected against unauthorized access with the aid of the appropriate BS2000 facilities. The files required by OMNIS should be created with *NOSHARE*.
- All passwords in OMNIS commands can be blanked out when entered in line mode (see the "OMNIS/OMNIS-MENU Functions and Commands" manual [1]). Such inputs are overwritten by OMNIS in the logging file with an OMNIS message and cannot therefore be found out from the logging file or by printing on the HCOPY printer. Only the administrator can disable this functionality by enabling the test mode for diagnostic purposes.

For security reasons, LINE entries in blanking mode are not buffered by the *INPUT-SAVE* function. If the K key is pressed for screen dump after a request for a LINE entry in blanking mode, the request is not processed for security reasons. If the partner is changed following a LINE entry request in blanking mode and *SAVE-SCREEN* is set for the terminal, the saved screen cannot be restored if you return to the original partner. The screen is deleted.

This applies not only to all passwords entered by the user (*BERID*, *CPASS/LPASS* and *CMMSG/LMSG* in the *OPNCON* command and *OPASS* in the *HALT*, *PARM* and *OCCUPY* commands), but also to all passwords specified in any administration command (*APASS*).

This method can be used to prevent a password required for connection setup from being entered in the logging file. If, when signing on to the partner, it is possible to enter a blanked password in line mode (as is the case, for example, for LOGON to a TIAM partner), the password is also overwritten by OMNIS in the logging file, and the same protection mechanism takes effect.

In the case of applications which require the input of passwords in forms, the passwords are neither written to the logging file nor printed on the hardcopy printer. If desired, the administrator can bypass this functionality for diagnostic purposes (by enabling test mode).

Start sequences stored in the text file must not contain LOGON passwords. File passwords, on the other hand, are permitted.

6.7 Level of operational security: SECURITY

With the start parameter *SECURITY*, the level of operational security for OMNIS can be defined. *SECURITY* cannot be changed while OMNIS is running.

SECURITY controls:

- access to BS2000 resources,
- access to administration functions (these can be restricted to the console),
- authorization for taking over connections,
- logging of OMNIS commands and messages in the logging file.

6.8 Behavior in the event of an error

If an error occurs in OMNIS or at the interfaces to the operating system, the error message OMS0005 is always issued, followed, if necessary, by a dump.

In most cases, OMNIS attempts to continue running after an error (the program is not terminated).

If program termination is unavoidable, OMNIS terminates itself with the message OMS0013 *OMNIS TERMINATED NORMALLY*, which means that the job in which OMNIS was started is continued after the next *ISTEP* statement (or at *ILOGOFF*). On normal termination (*END* command, */BCEND* or */BCLOSE*) or in the event of errors in the initialization phase, OMNIS terminates with the message OMS0013. This permits an automatic restart to be programmed with *JCL* statements (see also the "Commands" manual [3]).

If OMNIS generates a dump, one of the messages OMS0005, OMS0008 or OMS0032 is issued first: The first two messages contain an error code, which should be analyzed.

As additional information for a dump, an insert in the message IDA0N51 shows the OMNIS message number and the address at which the dump was requested.

If analysis of the error code indicates that an OMNIS error has occurred, the following diagnostic aids are required: SYSLSST and SYSOUT logs, dump, configuration input file, configuration file, trace file and logging file (on magnetic tape).

If the error was possibly caused by an exit routine, a listing of this exit routine should also be included.

If OMNIS does not generate a dump automatically, you should always request one yourself in the event of errors which only occur sporadically. You should likewise always request a dump if errors associated with exit routines occur.

Enter the *DUMP* command at a terminal other than that at which the error occurred.

If there are OMNIS-OMNIS connections, always request a dump for each OMNIS.

7 Settings in BS2000 and partners

Both in the BS2000 system and the partners, special settings are required for the use of OMNIS. In some cases, restrictions must also be observed when programming partner applications.

7.1 BS2000 system

Console as a terminal

Both the console and OMNIS must be taken into account in the BS2000 system.

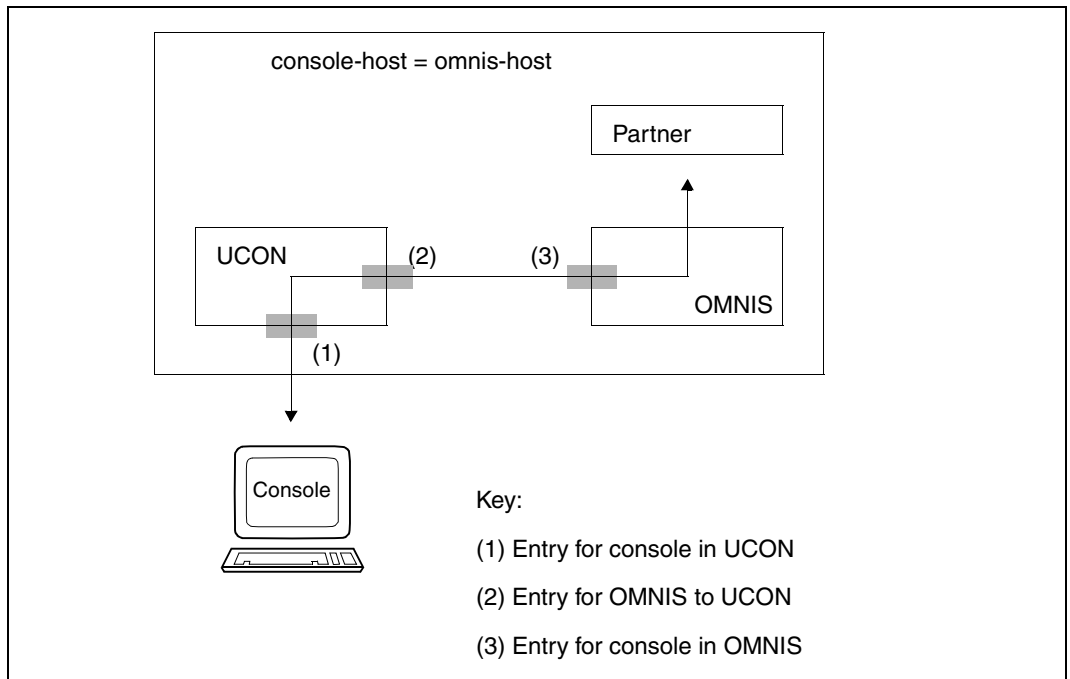


Figure 8: Console as a terminal

The BS2000 consoles wishing to work with OMNIS must be defined in the BS2000 parameterservice (parameter set OPR) and must have the authorization code of the OMNIS commands (see below).

```
SET-CODE CODE=authorization-code, CONSOLE=console1
```

The special command */OMNIS* must be generated.

```
SET-CMD-CODE AUTHORIZATION-CODE=authorization-code, CMD-NAME=command-name
```

e.g.: SET-CMD-CODE AUTHORIZATION-CODE=E, CMD-NAME=OMNIS

The system administrator must set up a user ID with the authorization name:

```
/ADD-USER USER-IDENTIFICATION=authorization-name2,
          PROTECTION-ATTRIBUTE=PARAMETERS
          (LOGON-PASSWORD=C'authorization-password', PUBLIC-SPACE-LIMIT=0,
          ACCOUNT-ATTRIBUTES=PARAMETERS (ACCOUNT=1, CPU-LIMIT=1))
```

All BS2000 consoles on which OMNIS is to be used must be declared with the OMNISKD routine (see [page 101](#)):

```
DECLARE-CONSOLE console1, MSG=authorization-name2, authorization-password
```

e.g. DECLARE-CONSOLE (C1), MSG=CONS, C'17+4'

OMNIS automatically establishes the connection to a BS2000 console after loading.

Operators can then communicate with OMNIS using the command */OMNIS* (or abbreviated to */O*):

```
/OMNIS @:OPNCON pac
```

```
/O pac:message
```

e.g.

```
/OMNIS OPNCON A
```

```
/O A:LOGON USID, ACCNB, X'1984AFEF'
```


Terminal as a console

If a terminal is to be used as a console, it must be authorized to issue console commands and receive console messages.

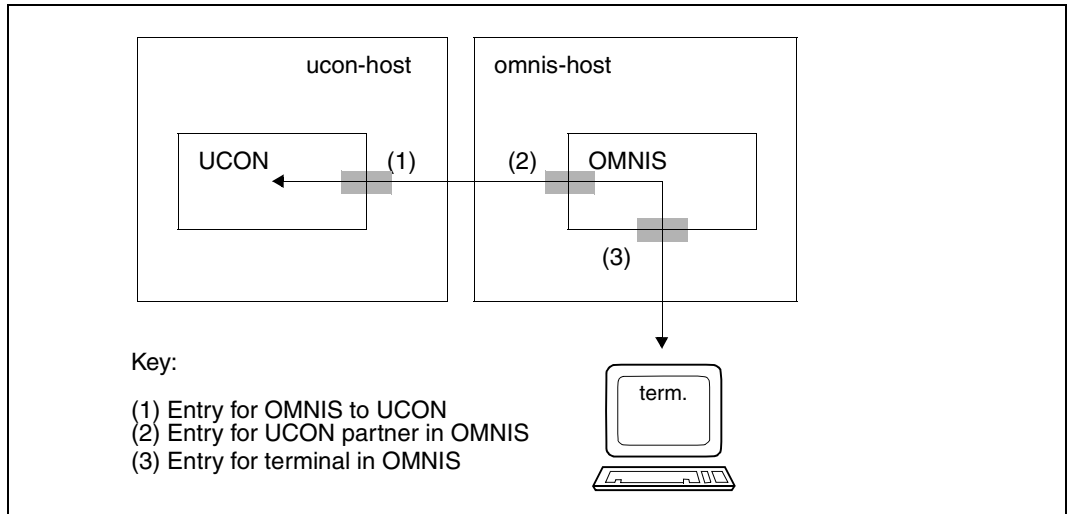


Figure 9: Terminal as a console

ucon-host and omnis-host may be identical.

An authorization code must be assigned for each terminal with console functions.:

```
SET-CODE CODE=authorization-code, CONSOLE=console1
SET-CODE CODE=authorization-code, CONSOLE=console1
SET-CODE CODE=authorization-code, CONSOLE=console3
...
```

Only the console commands and console messages that are assigned to a particular terminal (via authorization codes) may be entered or received at that terminal.

The system administrator must generate a user ID for each authorization name:

```
/ADD-USER USER-IDENTIFICATION=authorization-name,
PROTECTION-ATTRIBUTE=PARAMETERS
(LOGON-PASSWORD=C'authorization-password',PUBLIC-SPACE-LIMIT=0,
ACCOUNT-ATTRIBUTES=PARAMETERS(ACCOUNT=1,CPU-LIMIT=1))
```

A user can then sign on to UCON from a terminal using the following OMNIS command and perform the console functions defined by *authorization-name*.

```
OPNCON pac,TYP=UCON,PRONAME=ucon-host,
MSG=authorization-name,authorization-password
```

e.g. OPNCON A,TYP=UCON,CMSG=CON1,C'QUAX'

In order to prevent the connection to UCON from being aborted if there are too many console outputs to OMNIS, the *MODIFY-CONSOL-PARAMETER* command can be used to make the UCON buffer sufficiently large.

See the "Commands" manual [\[3\]](#).

7.2 UTM generation

Connections from terminals to UTM via OMNIS can be established in two different ways:

- Each terminal is connected to UTM via its own transport connection.
- Several terminals are connected to UTM by OMNIS via a single transport connection (the multiplex connection).

The use of multiplex connections results in considerable savings with regard to BS2000 resources. In both cases, the connection must be generated in UTM. The way in which the connection is generated depends on the connection type selected.

7.2.1 Generation without multiplex connections

Any connection which is to be set up via OMNIS from a terminal to a UTM application must be generated in UTM.

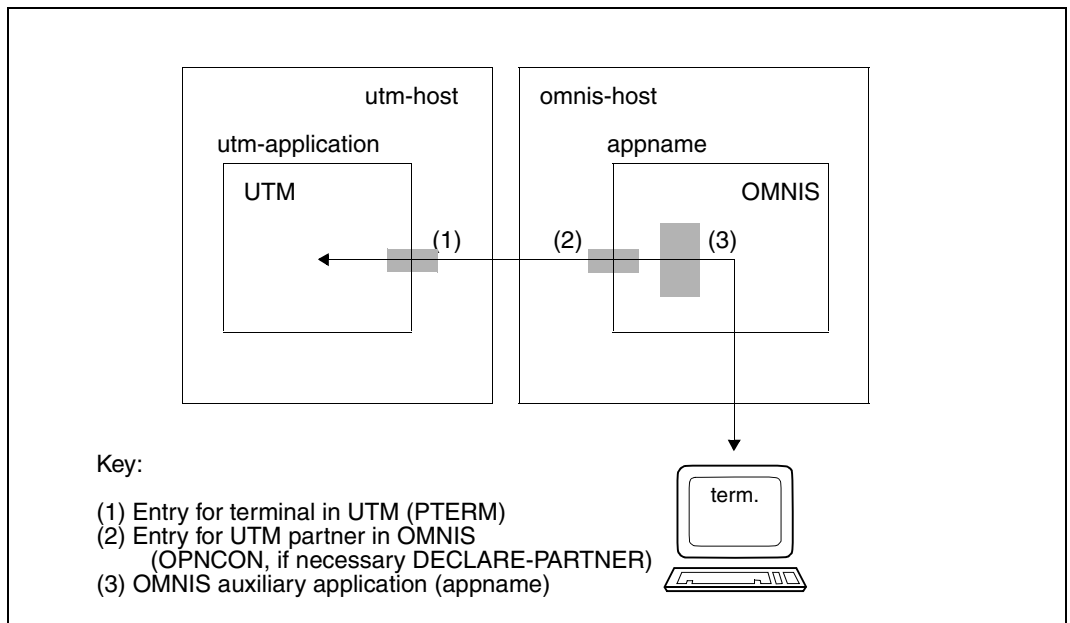


Figure 10: UTM generation without multiplex connection

omnis-host and utm-host may be identical.

Each connection must be generated in the *KDCFILE* of the UTM application:

```
PTERM appname,PRONAM=omnis-host,TERMN=terminal-code,
      LTERM=lterm-name,PROTOCOL=N,PTYPE=terminal-type
```

Any code and name can be chosen for *terminal-code* and *lterm-name*; *PROTOCOL=NO* must always be specified for OMNIS-UTM connections.

The *PRONAM* operand is used to specify the name of the processor on which OMNIS is loaded; *terminal-type* must correspond to the type of OMNIS terminal from which the connection is set up (*OPNCON* command). However, neither UTM nor OMNIS checks the type.

appname is freely selectable and must be specified in the *OPNCON* command which sets up the connection to UTM:

```
OPNCON pac,TYP=UTM,PTNNAME=utm-application,PRONAME=utm-host,APPNAME=appname,
      CPASS=connection-password
```

appname is the name of an auxiliary application set up by OMNIS and must (like all application names) be unique within the system. *utm-application* specifies the name of the UTM application, and *utm-host* the name of the processor in which the UTM application is started.

As the default option, *appname* consists of the prefix *OMS* followed by a five-digit number, these numbers being assigned by OMNIS sequentially.

The connection password is defined when the UTM application is initiated:

```
.UTM PASSWORD=connection-password
```

As an alternative to the above-mentioned *PTERM* statement, the following *TPOOL* statement may be entered:

```
TPOOL PRONAM=omnis-host,TERMN=terminal-code,NUMBER=number
      LTERM=lterm-name,PROTOCOL=N,PTYPE=terminal-type
```

The *APPNAME* operand is no longer specified in the *OPNCON* command used to establish the connection to UTM.

PROTOCOL=NO (UTM connection) may be generated only once for each processor.

UTM administration

It is also possible to administer UTM applications using OMNIS provided that a terminal is entered in UTM as administrator (*USER* and *LTERM* statements) and that an appropriate *OPNCON* command is issued.

If the administration is to be executed under program control (with an exit routine), generation with the following statements is advantageous. If administration is to be executed only via the screen, the generation described on the previous pages is better.

```
USER  useradm,STATUS=ADMIN
PTERM appname,PRONAM=omnis-host,LTERM=lterm-name,PTYPE=APPLI,PROTOCOL=N
LTERM ltermname,USER=useradm
```

The specification *PTYPE=APPLI* in the *PTERM* statement causes inputs and outputs to be executed in line mode (and not in formatted mode, which is not suited to administration activities).

TYP=DCAM must be specified in the OMNIS *OPNCON* command:

```
OPNCON pac,TYP=DCAM,PTNNAME=utm-application,PRONAME=utm-host,APPNAME=appname,
        LPASS=connection-password
```

The UTM administrator is now authorized to administer the UTM application via OMNIS.

7.2.2 Generation for multiplex connections

A multiplex connection must be generated in UTM and OMNIS, and the terminals involved generated in UTM.

Generation in OMNIS

In OMNIS the multiplex connection must be generated as follows:

```
DECLARE-MUX PTNNAME=utm-application-name,PRONAME=utm-host
```

Generation in UTM

The multiplex connection is specified with the following generation statement:

```
MUX omnis-appname,PRONAM=omnis-host
```

where *omnis-appname* is the name of the OMNIS application in the OMNIS ENTER file specified with:

```
/SET-DCAM-APPLICATION-LINK LINK-NAME=OMNIS,APPLICATION-NAME=appname
```

Generation of the terminal in UTM depends on the OMNIS *OPNCON* command issued and whether exit routines are used:

- *OPNCON* without *APPNAME* specification,
- *OPNCON* with *APPNAME* specification,
- *OPNCON* is supplied with values in an exit routine.

OPNCON without specification of *APPNAME*:

```
OPNCON pac,TYP=UTM,ROUTE=MUX,PRONAME=utm-host,...
```

In this case, the terminal is known under the real terminal name. The following applies to UTM generation:

```
PTERM terminal-name,PRONAM=front-end-processor,PTYPE=terminal-type,  
LTERM=lterm-name,PROTOCOL=STATION
```

PRONAM is the name of the partner computer of the terminal from which the user can work with OMNIS.

Generation of the terminal in the net configuration and in UTM must be the same. For example, the following statement must be entered for the computer with terminal name DT100 and the name of the partner computer FEP1:

```
PTERM DT100,PRONAM=FEP1,...,PROTOCOL=STATION
```

OPNCON with specification of APPNAME:

```
OPNCON pac,TYP=UTM,ROUTE=MUX,PRONAME=utm-host,APPNAME=appname,...
```

In this case the terminal name is replaced by the name specified for *APPNAME*. Therefore, the following *PTERM* statement must be entered for the above example:

```
PTERM appname,PRONAM=front-end-processor, ...,PROTOCOL=STATION
```

OPNCON is supplied with values in an exit routine:

The *OPNCON* command can be supplied with the required values in an exit routine (see [chapter "Programming with OMNIS" on page 249](#)). In particular, the terminal name can be replaced by the name specified in *OMEUNAME* and the processor name with the name specified in *OMEUPROC*.

If *OMEUNAME=name1* and *OMEUPROC=proc1*, the following *PTERM* statement must be specified in the UTM generation:

```
PTERM name1,PRONAM=proc1,...
```

The relationship between the specifications made in the *OPNCON* command and the corresponding *PTERM* statements for UTM generation are illustrated in the following figure:

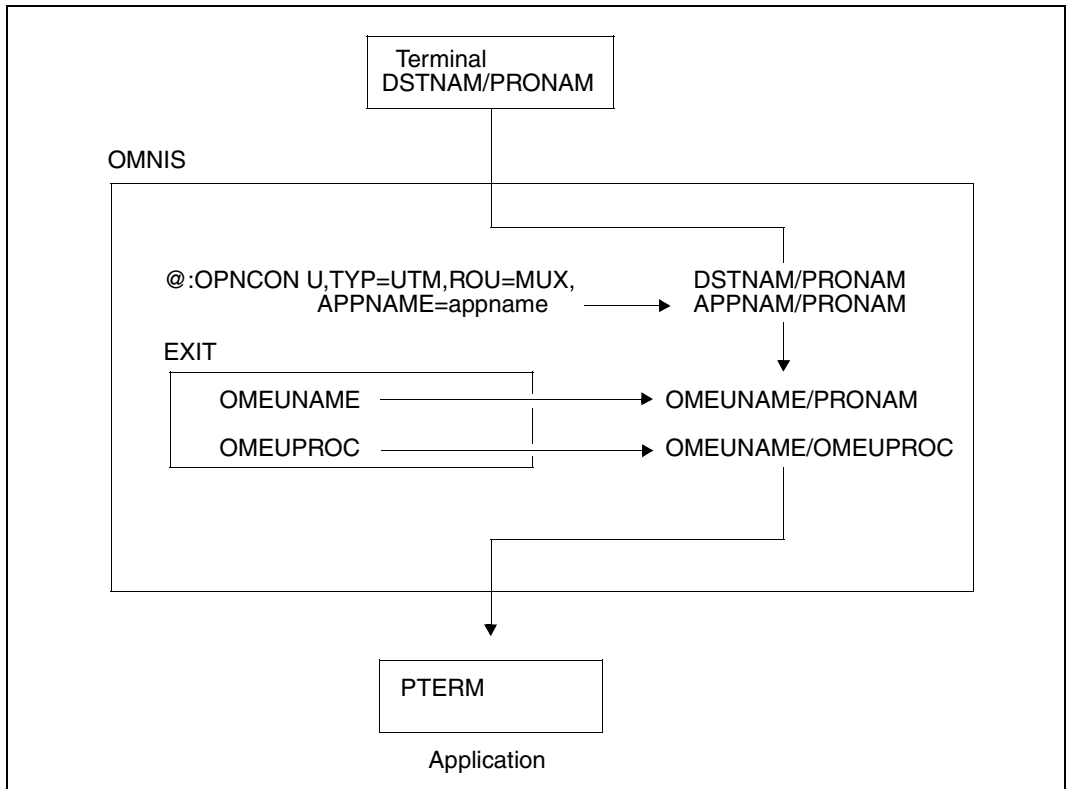


Figure 11: Hierarchy of terminal and processor names

During UTM generation for multiplex connections, the following points should also be noted:

- OMNIS must not possess a connection password.
- The maximum number of simultaneously active sessions may be generated.
- In the *PTERM* statement, the type of the terminal must be specified as the terminal type.
- *PROTOCOL=STATION* must always be specified in the *PTERM/TPOOL* statement (default assumed by UTM).
- In the UTM generation, an appropriate *TPOOL* command may be specified instead of explicit terminal generation (*PTERM/LTERM* statement):

```
TPOOL PRONAM=front-end-processor,TERMN=terminal-code,LTERM=1term-name,
      NUMBER=number PROTOCOL=STATION,PTYPE=terminal-type
```


Load distribution through parallel multiplex connections

It is possible to generate several (parallel) multiplex connections from OMNIS to a UTM application.

To do this, a UTM application is assigned several application names and an appropriate number of multiplex statements is issued. In the OMNIS generation, *DECLARE-MUX* statements must be specified for the various application names.

For example, 200 terminals can be divided between two multiplex connections:

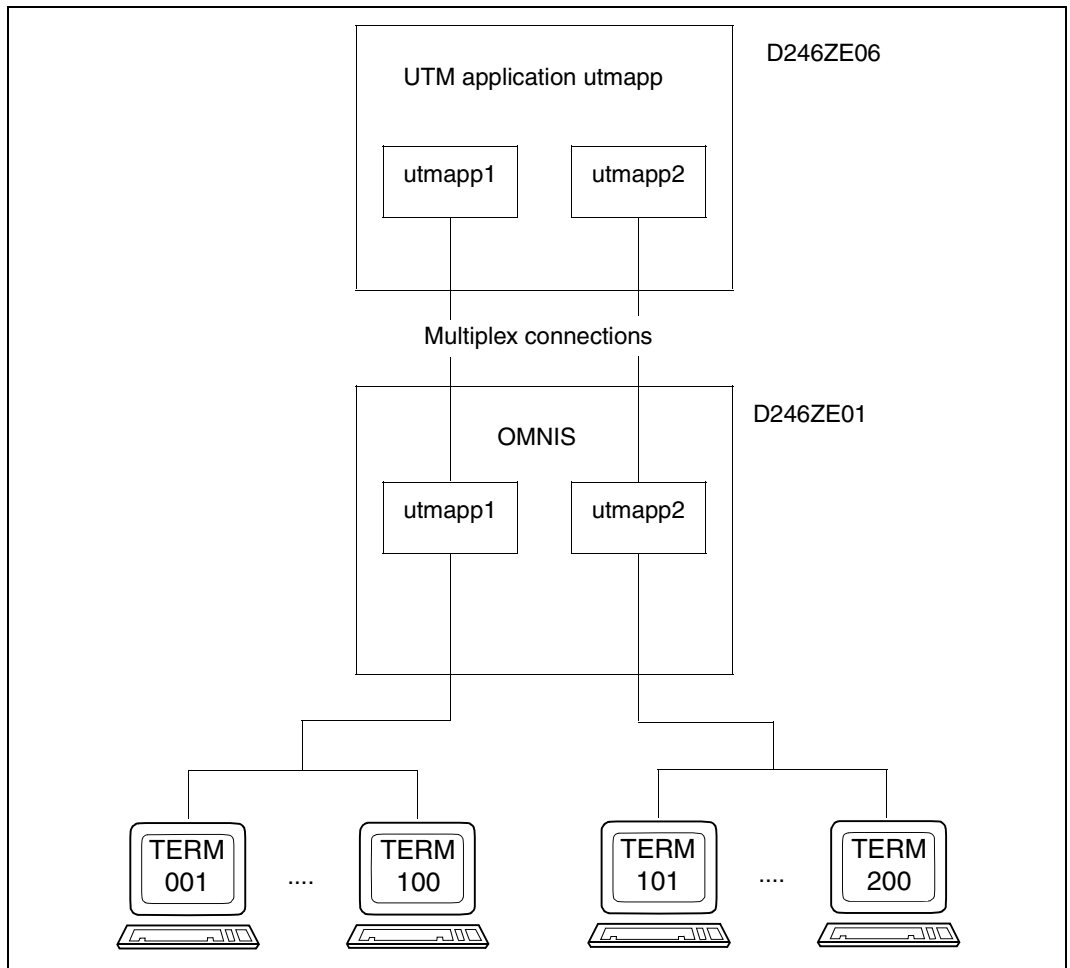


Figure 12: Distribution of 200 terminals between two multiplex connections

The following declarations must be made in UTM for the above example:

```
BCAMAPPL local-appname1
```

```
BCAMAPPL local-appname2
```

```
MUX omnis-appname,PRONAM=omnis-host,BCAMAPPL=local-appname1
```

```
MUX omnis-appname,PRONAM=omnis-host,BCAMAPPL=local-appname2
```

Declarations in OMNIS:

```
DECLARE-MUX PRONAME=utm-host,PTNNAME=uttmuxname1
```

```
DECLARE-MUX PRONAME=utm-host,PTNNAME=uttmuxname2
```

uttmuxname1 and *uttmuxname2* are identical to *local-appname1* and *local-appname2* in *BCAMAPPL*.

7.3 Prerequisites for DCAM programming

For DCAM programming, a distinction must be made between two different types of connections.

DCAM partner with `PROTOCOL=DSSIM`

The operand *PROTOCOL* sets the behavior of OMNIS with respect to a DCAM application. If *PROTOCOL=DSSIM* is specified, OMNIS behaves, from the viewpoint of a DCAM application, like a real terminal. This connection type is recommended for DCAM partners which work only in a strict dialog with formatted messages or which do not accept connection requests from programs. Switching between DCAM partners with *PROTOCOL=DSSIM* is possible only with the aid of K keys.

DCAM partner with `PROTOCOL=OMNIS`

To a DCAM application, OMNIS (and the terminal on which OMNIS is used) appear like a program. The DCAM application has no way of determining whether a "real" program or OMNIS has signed on, and must decide, on the basis of the partner's name, whether or not it is dealing with an OMNIS connection.

Since a DCAM application regards OMNIS as a program, and not as a terminal, the NEABT protocol is not used on the connection. The connection must be set up with *EDIT=USER* (YOPNCON ACCEPT); if *EDIT=SYSTEM* is specified, no connection will be established.

This results in the following restrictions:

- All functions which depend on *EDIT=SYSTEM* have no effect. This affects the following functions:

```
EDITIN=LCASE/NLCASE, GETBS/NGETBS, GETFC/NGETFC ;
EDITOUT=HCOPY/NHCOPY, HOM/NHOM, LOGC/NLOGC, LACK/NLACK ;
PROC=SYSCODE/BINARY .
```

Corresponding entries for *YOPNCON* will be ignored without warning; those for *YCHANGE* will be rejected with *RC=1874*. The entries valid for operation via OMNIS are underlined.

- *EDITOUT=EXTENDED* has no effect and is treated like *EDITOUT=LINE*.
- The partner characteristics are not available, which means that the DCAM application cannot determine the characteristics of the terminals "hidden behind" OMNIS. This also applies to the terminal type.
- *MAXLN* is currently always set to 4Kbytes. This means that the 9750-62 Data Display Terminal cannot be supported at all and the 9763 Data Display Terminal can be supported only with restrictions.

These restrictions can be partially circumvented, as follows:

- Instead of *EDITIN=LCASE/NLCASE*, the operand *LCASE=YES/NO* can be specified in the *OPNCON* and *PARAM* commands or in the *DECLARE-PARTNER* statement. *LCASE* specifies whether or not lowercase letters are to be passed on to the DCAM partner. Note, however, that this setting applies to all messages, while *EDITIN=LCASE/NLCASE* can be selected differently for each individual message.
- Instead of *EDITOUT=HCOPY/NHCOPY*, the corresponding OMNIS function can be used (*HCOPY* operand in the *SET*, *OPTION*, *PARAM* and *OPNCON* commands or *DECLARE-TERMINAL* and *DECLARE-PARTNER* statements). Note, however, that this setting applies to all messages, while *EDITOUT=HCOPY/NHCOPY* can be selected differently for each individual message.
- *EDITIN/EDITOUT=LINE/PHYS/FORM* is not needed since OMNIS correctly recognizes the formatting of a message. However, "mixing" of *LINE/PHYS/FORM* is not possible, i.e. *EDITIN* must always match *EDITOUT*.

The following restriction for connection setup must be noted: OMNIS is always responsible for initiating a connection setup to a DCAM application and cannot accept connection requests from DCAM applications. It is not possible to set up a connection from a DCAM application to a terminal "hidden behind" OMNIS: *YOPNCON ACQUIRE* has no effect.

The format handler FHS can be used without restrictions. This means that the DCAM application can edit a message with FHS and then send it using *YSEND (EDIT=USER)* to the terminal "behind" OMNIS.

The following rules should be observed when programming:

- Special handling of OMNIS terminals is necessary when setting up a connection with *YOPNCON ACCEPT*, as well as with *YCHANGE* and *YINQUIRE PTNCHAR*.
- Changing of partners to a DCAM application with *FORM* or *PHYS* will function only if the DCAM application reacts correctly. If a DCAM application is selected using a K key, OMNIS sends a message which looks as if the user has switched the terminal off and then on again and then pressed K3. If FHS is being used, this results in the return code "screen destroyed" and the DCAM application must react by restarting the screen.

8 OMNISKD: Declaring a configuration

The OMNISKD utility routine described below is used to create a configuration file for OMNIS and OMNIS-MENU.

You can call OMNISKD either with the *START-PROGRAM* command or with *START-OMNISKD*, provided that the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* is enabled.

8.1 Using OMNISKD

OMNISKD reads definition records from an input file to be set up by the user, subjects them to checks for syntax and semantics, and constructs a configuration file from them.

Example

To run OMNISKD you might use the following procedure:

```
/BEGIN-PROCEDURE PARAMETERS=NO, LOGGING=NO
/WRITE-TEXT '**** OMNISKD **** : 085'
/SET-FILE-LINK FILE-NAM=KR.TAB.TACTAB, LINK-NAME=OMMTAC
/ FILE KR.TAB.GEN, BLKSIZE=(STD,2), LINK=OMMGEN
/ FILE KR.TAB.USER, BLKSIZE=(STD,2), LINK=OMMUSER
/ CREATE-FILE FILE-NAME=KR.TAB.MENU, SUPPRESS-ERRORS=-
/ *FILE-EXISTING, SUPPORT=PUBLIC-DISK( SPACE=-
/ RELATIVE(PRIMARY-ALLOCATION=48, SECONDARY-ALLOCATION=16))
/ SET-FILE-LINK FILE-NAME=KR.TAB.MENU, LINK-NAME=OMMMENU, -
/ BUFFER-LENGTH=STD(SIZE=16)
/ FILE KR.TAB.APPL, BLKSIZE=(STD,2), LINK=OMMAPPL
/SET-FILE-LINK LINK-NAME=SOMSKD, -
/ FILE-NAME=KR.KON.AUS
/SET-FILE-LINK LINK-NAME=SOMSTEXT, ACCESS-METHOD=ISAM(KEY-LEN=8), -
/ FILE-NAME=KR.TEXT
/ SET-JOB-STEP
/SET-FILE-LINK LINK-NAME=OMMPLUS, FILE-NAME=KR.PKEY.IN
/ SET-JOB-STEP
/ MODIFY-MSG-FILE-ASSIGNMENT ADD-FILE=SYSMSA.OMNIS.085
/ SET-JOB-STEP
```

```
/          ASSIGN-SYSDTA  TO=*LIBRARY-ELEMENT(LIBRARY=KR.B,-
/                               ELEMENT=GEN.TXT,TYPE=S)
/          ASSIGN-SYSLST  TO-FILE=KR.KON.LST
/START-PROGRAM *M($OMSE.SYSLNK.OMNIS.085,OMNISKD,RUN-MOD=ADV), -
/          TEST-OPTIONS=AID
/          SET-JOB-STEP
/          ASSIGN-SYSDTA TO-FILE=KR.PKEY.IN
/          ERASE KR.TAB.PKEY
/          SET-JOB-STEP
/          START-PROG PLUS
/          ASSIGN-SYSDTA  TO=*PRIMARY
/          ASSIGN-SYSLST  TO=*PRIMARY
/          REMOVE-FILE-LINK LINK-NAME=OMMGEN
/          REMOVE-FILE-LINK LINK-NAME=OMMUSER
/          REMOVE-FILE-LINK LINK-NAME=OMMMENU
/          REMOVE-FILE-LINK LINK-NAME=OMMAPPL
/          REMOVE-FILE-LINK LINK-NAME=OMMTAC
/          REMOVE-FILE-LINK LINK-NAME=OMMPLUS
/          REMOVE-FILE-LINK LINK-NAME=SOMSKD
/          REMOVE-FILE-LINK LINK-NAME=SOMSTEXT
/          REMOVE-FILE-LINK LINK-NAME=BLSLIB01
/SET-JOB-STEP
/EXIT-PROCEDURE
```

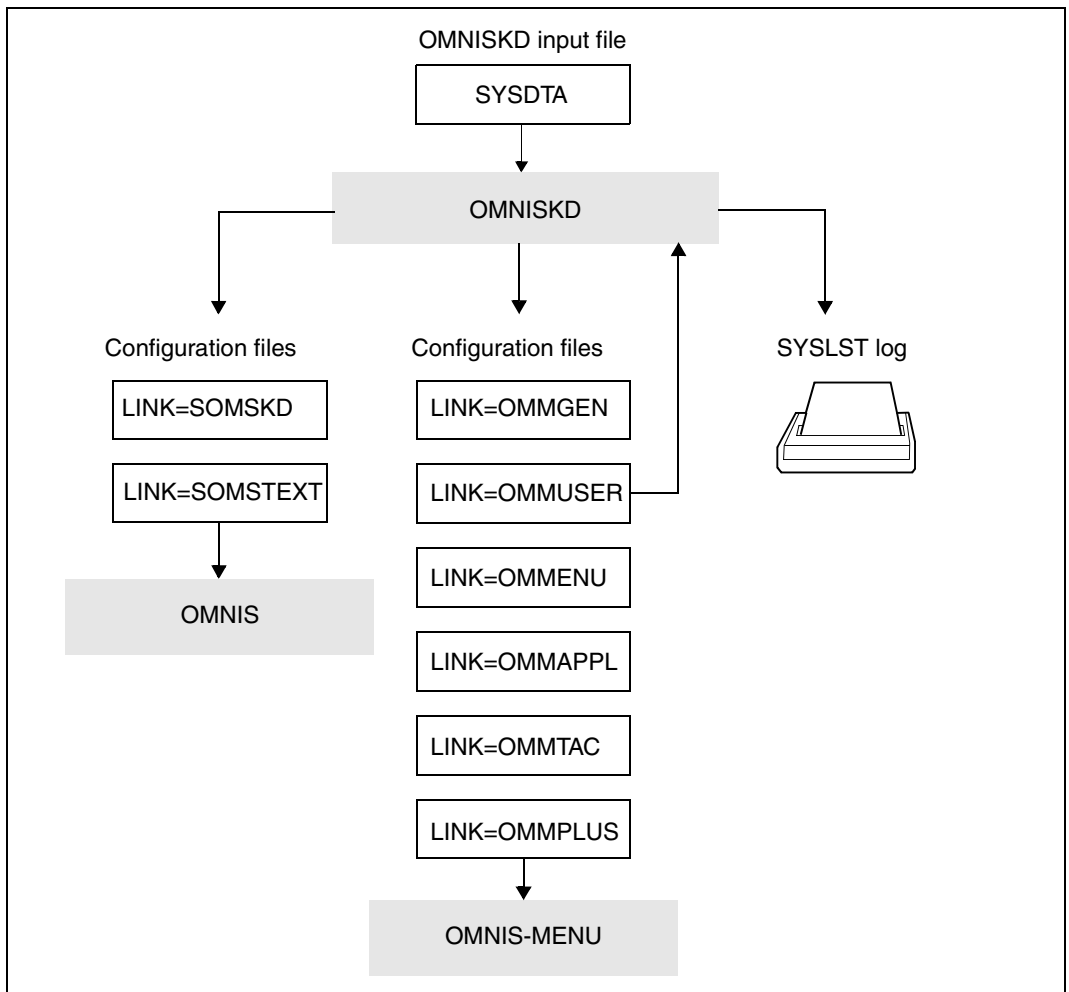


Figure 13: Declaring a configuration with the aid of OMNISKD

To enable the user passwords to be transferred, the user table also acts as an input file for OMNISKD.

Contents of the OMNISKD input file

The OMNISKD input file is a normal SAM or ISAM file or type-S library member. The administrator can use an OMNISFC output file as the input file, or alternatively must create and maintain it using an editor.

The OMNISKD input file covers a complete configuration. Each record describes one object:

DECLARE-PARAMETER

declares global parameters for OMNIS-MENU.

DECLARE-START-SEQUENCE

declares a start sequence for the text file.

DECLARE-OMNIS

declares an OMNIS-OMNIS connection and its characteristics.

DECLARE-EXIT

declares an exit group and its modules.

DECLARE-MUX

declares a multiplex connection to a UTM application.

DECLARE-HARDCOPY

declares a directly connected printer and its characteristics.

DECLARE-CONSOLE

declares a BS2000 console as a terminal.

DECLARE-TERMINAL

declares a terminal and its characteristics.

DECLARE-PARTNER

declares a partner and its characteristics.

DECLARE-GROUP

declares a partner group.

DECLARE-PKEYSET

declares a set of P keys.

DECLARE-TAC

declares an entry in the TAC allocation table.

DECLARE-APPLICATION

declares the individual applications.

DECLARE-MENU

declares the applications which appear on a menu.

DECLARE-USER

declares the users and the menus which are available to them.

The relationships between the objects are expressed sometimes by references, sometimes by the order of declarations.



Passwords from an old user table are transferred only in the following cases:

- OMNISKD is assigned the old user table (OMMUSER).
- The operand *PASSWORD=*UNCHANGED* is declared for the user entry.

OMNISKD records the current OMNIS-MENU version in the user table. As a result, you can no longer use this user table with an earlier version of OMNIS-MENU.

Order of declarations

The permitted order of declarations is fixed, and must be strictly adhered to. As a general principle, an object that refers in any way to another must be declared after that one.

1. The global parameters must be declared first.
2. Then comes the start sequence declaration.
3. Then follows the (optional) declaration of all OMNIS-OMNIS connections. The order of OMNIS-OMNIS declarations within the OMNIS block is not significant.
4. Then follows the (optional) exit block. The order of exit declarations within the block is not significant.
5. Then follows the (optional) multiplex block. Multiplex declarations may also be entered after the printer declarations, but they must be come before the first terminal declaration.
6. Then follows the (optional) printer block. The order of printer declarations within the block is not significant.
7. Any number of terminal blocks can then follow.

A terminal block consists of the declaration of a terminal or BS2000 console, followed by the associated partner and group declarations.

The order of partner declarations within a terminal block is not significant. All members of a group should have been declared previously. If a partner is not declared, they will be added to the group with default attributes.

8. Then follows the declaration of the P key sets, of the entries in the TAC table and of the globally applicable applications. The sequence of individual declarations within the blocks can vary.
9. Then come the menu blocks.

A menu block consists of a menu declaration followed by the associated locally applicable applications.

10. Last comes the declaration of the users.

OMNISKD reads the definition records and in a second pass performs consistency checking to ensure that all the menus defined in the user entries and all the applications defined in the menus actually exist.

Thus any error messages are output at the end of the SYSLST file.

The following figure outlines the structure of an OMNISKD input file.

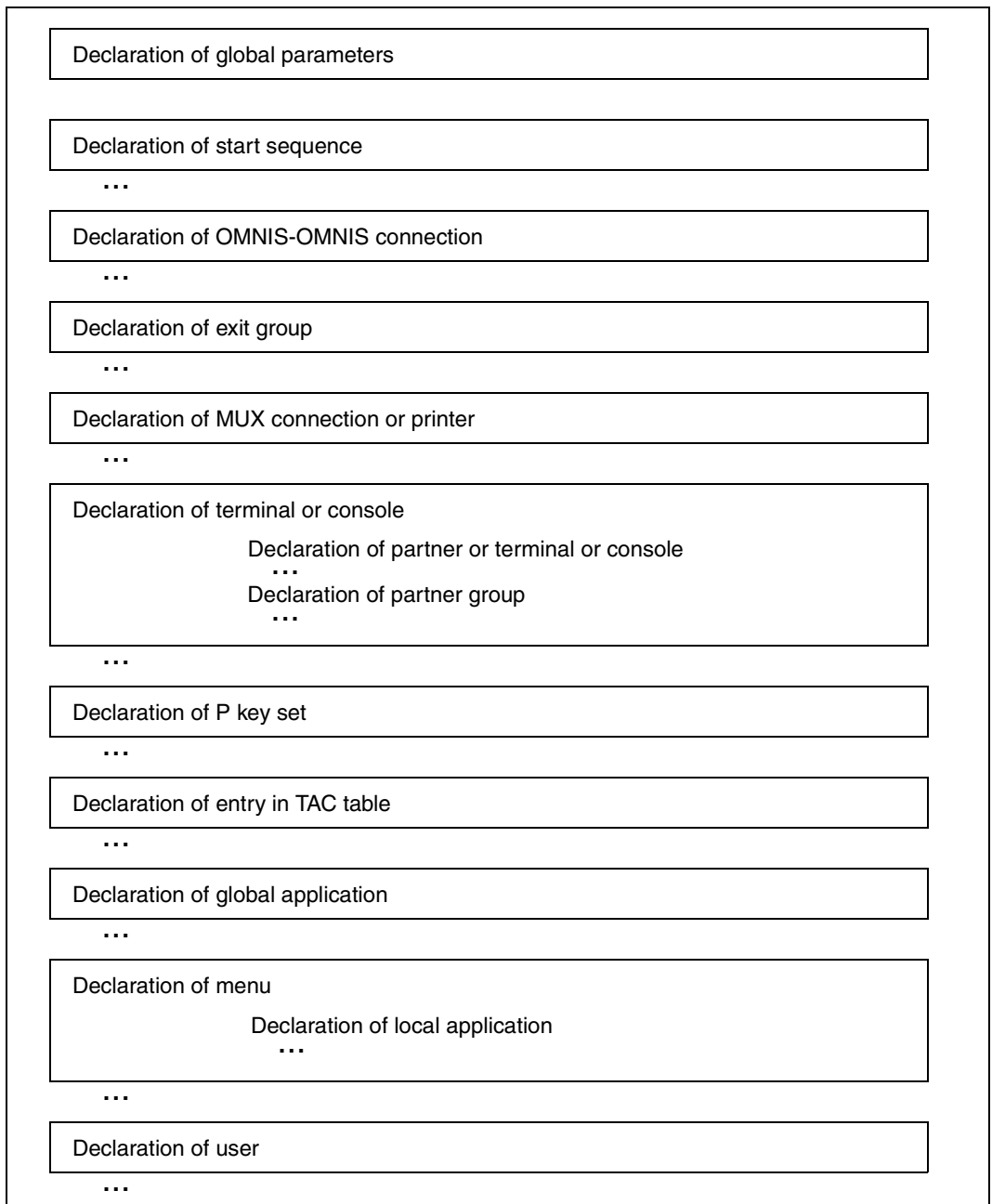


Figure 14: Structure of an OMNISKD input file

Example

```
DECLARE-OMNIS  omnis_1
DECLARE-OMNIS  omnis_2
DECLARE-OMNIS  omnis_3
DECLARE-EXIT  exit1
DECLARE-EXIT  exit2
DECLARE-EXIT  exit3
DECLARE-HARDCOPY  printer1
DECLARE-HARDCOPY  printer2
DECLARE-MUX  application1
DECLARE-TERMINAL  terminal_1
DECLARE-TERMINAL  terminal_2
    DECLARE-PARTNER  partner_11
    DECLARE-PARTNER  partner_12
    DECLARE-PARTNER  partner_13
    DECLARE-GROUP  group_14
    DECLARE-PARTNER  partner_15
    DECLARE-GROUP  group_16
DECLARE-TERMINAL  terminal_3
    DECLARE-PARTNER  partner 21
    DECLARE-PARTNER  partner 22
```

In the above example three OMNIS-OMNIS connections, three terminals, six partners, two groups, three exit groups, two printers and a multiplex connection are declared. Partners 11, 12, 13, and 15, and groups 14 and 16, are allocated to the second terminal; partners 21 and 22 to the third terminal. The first terminal has no partners.

Configuration file

The configuration file contains the information from the OMNISKD input file in a form edited for OMNIS. It must not be altered by the user.

For data protection reasons, all passwords in the configuration file (*CPASS*, *BERID*, *OPASS*, *CMSG*) are encrypted.

OMNISKD log

The OMNISKD log contains a list of the input statements processed by OMNISKD, together with any error messages.

For data protection reasons, all passwords (*CPASS*, *BERID*, *OPASS*, *CMSG*) are overwritten with 'S' during output to SYSLST. All strings enclosed in single quotes are regarded as passwords.

If there is an error in a declaration, the record will not be written.

8.2 Notational conventions

Metasyntax of the OMNISKD statements

The notation used to describe the syntax of the control statements is the same as that used for the OMNIS commands.

Notation	Explanation	Example
UPPERCASE LETTERS	Uppercase letters indicate constants which must be entered exactly as shown.	ADM=YES Enter the following: ADM=YES
lowercase letters	Lowercase letters indicate variables which the user must replace with a current value.	BREAK-KEY=number Specify for example: BREAK-KEY=3
{ }	Braces enclose alternative entries of which either one or none may be specified. Default values are <u>underlined</u> .	ADM={ <u>NO</u> } { YES } Enter the following ADM=NO or ADM=YES. If ADM is not specified, ADM=NO applies.
[]	Brackets enclose entries which may be omitted.	M[ODULES] Specify for example: MODULES, MOD or M.

When entering commands, **no** comma should be placed in front of the first operand.

General statement format

The statement format is also the same as for the OMNIS commands. The only difference is that no OMNIS address codes are needed.

A command has the general format:

```
cmd [operand-list]
```

At least one blank (or a comment) must be entered between *cmd* and *operand-list*.

Any number of blanks can be inserted into the command before and after each separator. Valid separators are commas, blanks, left and right parentheses, start and end of command and the equals sign.

A comment is any desired character string enclosed in quotes (") at the beginning of a record. Furthermore, an entire record can be flagged as a comment by entering an asterisk (*) in the first column.

The statements can be indented as desired in order to format the input more clearly.

A statement may extend over several lines. For this, any line which is to be continued must contain a minus sign (-) as the last character. This continuation character may be in any column of the line (not necessarily column 72). A statement may have any number of continuation lines. The maximum record length that can be processed is 256 characters.

Syntax of the most frequently used operands

pac, gac, eac, hac

consists of a maximum of four alphanumeric characters (letters or digits, but not special characters).

Examples

A, KARL, 1XR, 123K

oac consists of the @ character followed by up to three alphanumeric characters (letters or digits, but not special characters).

Examples

@1, @X1R, @MUE

string can be specified in two formats:

Character format: C'ccc' or 'ccc'

Hexadecimal format: X'xxx'

ccc

stands for n user-selected characters as entered from the terminal (single quotes must be entered twice).

xxx

stands for 2n hexadecimal characters from the character set

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F

n gives the length of the character string. Minimum and maximum lengths are specified in the descriptions of the individual operands.

Examples

C'A@X?', 'AB'C', X'1A4339'

num is a character string consisting of digits from the character set

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

passwd

is a string in which n can assume the values 1 through 8.

Variable concept

The implementation of the variable concept means that variables can be used in some OMNISKD statements. This is true for the following statements:

OMNISKD statement	Parameters
DECLARE-APPLICATION	PTAST RESTART-MESSAGE CMSG/LMSG BERID PRONAME PTNNAME
DECLARE-MENU	HCOPY (DEVICE,FORM)
DECLARE-PARTNER	CMSG/LMSG BERID PRONAME PTNNAME
DECLARE-START-SEQUENCE	RECORDS
DECLARE-USER	HARDCOPY-DEVICE HARDCOPY-FORM

8.3 OMNISKD control statements

DECLARE-APPLICATION - Declare an application

The *DECLARE-APPLICATION* statement is used to declare an application.

Operation	Operands
D[ECLARE]-A[PPLICATION]	$PA[TH] = \left(\begin{array}{c} \text{menu_entry} \\ * \end{array} \right) , \text{pac}$ $,PR[ONAME] = \left(\begin{array}{c} \text{praname} \\ *ANY \\ *PREDIALOG \end{array} \right)$ $[,A[PPLICATION]-N[UMBER] = \underline{0}..99]$ $[,AP[PNAME] = \text{appname}]$ $[,A[UTOLOGOFF]-B[UT] = \left(\begin{array}{c} \underline{N[O]} \\ Y[ES] \end{array} \right)]$ $[,A[UTOMATIC]-C[LSCON] = \left(\begin{array}{c} \underline{N[O]} \\ Y[ES] \end{array} \right)]$ $[,B[ERID]=\text{authorization}[,c'\text{password}']]$ $[,COD[E] = \left(\begin{array}{c} \underline{S[YSCODE]} \\ B[INARY] \end{array} \right)]$ $[,COM[MENT] = \text{'max-string70'}]$ $[,CP[ASS] = \left(\begin{array}{c} c'cpass' \\ x'cpass' \end{array} \right)]$

Operation	Operands
	$[,LC[ASE] = \left\{ \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}]$ $[,LI[NE] = \left\{ \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}]$ $[,PROT[OCOL] = \left\{ \begin{array}{l} \underline{O[MNIS]} \\ D[SSIM] \end{array} \right\}]$ $[,PTA[ST] = pkeyset]$ $[,PTN[NAME] = \left\{ \begin{array}{l} ptnname \\ *PREDIALOG \end{array} \right\}]$ $[,R[ESTART]-M[ESSAGE] = \left\{ \begin{array}{l} X'...' \\ C'...' \end{array} \right\}]$ $[,R[OUTE] = \left\{ \begin{array}{l} \underline{D[IRECT]} \\ I[NDIRECT] \\ M[ULTIPLX] \end{array} \right\}]$ $[,SA[VE] = \left\{ \begin{array}{l} \underline{ST[D]} \\ A[LL] \\ N[O] \\ P[KEY] \\ SC[REEN] \end{array} \right\}]$ $[,SI[GNON] = \left\{ \begin{array}{l} \underline{B[OTH]} \\ U[SER] \\ N[ONE] \\ S[UPPRESS] \end{array} \right\}]$

Operation	Operands
	<p data-bbox="577 243 826 344">[,SN[MP]=$\left. \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\}$]</p> <p data-bbox="577 378 934 546">[,T[YPE]=$\left. \begin{array}{l} \text{TIAM} \\ \text{DCAM} \\ \text{UTM} \\ \text{UCON} \\ \text{*PREDIALOG} \end{array} \right\}$]</p> <p data-bbox="577 554 947 596">[,U[SER]-N[UMBER] = 0..9999]</p> <p data-bbox="577 621 1270 722">[,U[SER]-N[UMBER]-P[RRIORITY]=$\left. \begin{array}{l} \text{USER-MENU-APPLICATION} \\ \text{APPLICATION-USER-MENU} \end{array} \right\}$]</p> <p data-bbox="577 747 840 789">[,CM[SG] = connmsg]</p>

Meaning of the operands

- PATH** specifies the application to which this entry refers.

 - menu_entry** The application is a local application. No other menu is allowed or able to use this application. *menu_entry* is the name of the menu. An asterisk as the first character of *menu_entry* is reserved for administrator functions.
 - *** The application is a global application. The key of these applications is (**,pac*), where the PAC is globally unique. Global applications can be declared in more than one menu.
 - pac** The PAC of the application. The PAC must be globally unique for global applications; for local applications it must be unique within the menu. *pac* must be no more than four (alphanumeric) characters long.
- PRONAME** specifies (together with *PTNNAME*) the application's network address.

 - *ANY** **ANY* causes the required processing load to be distributed over multiple BS2000 hosts: OMNIS determines the two hosts with the lowest loads in the default job class and tries to set up a connection to the host with the lowest load.

This "load balancing" on multiple hosts by OMNIS is offered for the partner types TIAM and DCAM. The load balancing option is not available for indirect partners.

*PREDIALOG *PRONAME* should be determined by the user in dialog mode before the connection is opened.

APPLICATION-NUMBER

Freely selectable two-digit number, used together with the user number in start sequence definitions (see "User-specific start sequences", [page 41](#)).

The default value of 0 means that there is no start sequence defined.

APPNAME

defines the name of the "simulated terminal" under which OMNIS sets up the connection to the application. OMNIS is known to the application by this name.

appname may also be specified as a partially-qualified entry in the form of a three-character prefix, followed by *. In such cases, OMNIS independently creates a unique name by using the prefix and thus enables multiple instances of applications to be opened.

If *ONE-USER-ONE-CONNECTION=NO* has been set, even a local application may be opened concurrently more than once.

If no *appname* is specified, OMNIS generates a unique name.

AUTOLOGOFF-BUT

defines how the connection is set up to the application. This parameter applies only to TIAM applications. Default=*NO*.

NO The TIAM command *LOGOFF BUT* remains in effect, the subsequent *LOGON* can be performed.

YES Following the TIAM command *LOGOFF BUT*, the connection to the application is cleared down; in other words, the *CLSCON pac* command is issued.

AUTOMATIC-CLSCON

defines whether the connection to the application is cleared down when the application is exited. Default=*NO*.

NO When the application is exited, the connection to the application is maintained.

YES When the application is exited, the connection to the application is cleared down. Connection clear-down is not indicated.

BERID	specifies the authorization generated in the SKP and the password. The <i>BERID</i> operand is not a genuine keyword operand and must always be the last operand in the operand list.
authorization	Authorization generated in the SKP (1..8 characters)
password	Password generated in the SKP (1..8 characters; must be specified in the form <i>c'password'</i>).
CODE	specifies the format in which data is sent to the DCAM application. Default= <i>SYSCODE</i> .
SYSCODE	Data is sent to the DCAM application in EBCDIC. Data transport is not transparent (see "DCAM Macros" manual [8]).
BINARY	Data transport is transparent.
COMMENT	is a comment on the application, no more than 70 characters long. The first 20 characters of this comment are used as the default entry for the <i>TEXT</i> operand in the <i>DECLARE-MENU</i> statement.
CPASS	<i>CPASS</i> can be replaced with <i>LPASS</i> . It defines a connection password used in setting up the connection to the application.
c'cpass'	The password itself must be no more than 4 characters long. Example: C'XY12' (7 elements in total)
x'cpass'	The hexadecimal password <i>cpass</i> must be no more than 8 characters long and should have an even number of characters. Example: X'ABC123DF' (11 elements in total)
LCASE	specifies whether lowercase letters are to be passed to DCAM applications. Default=NO. Does not have any effect on the connection message during connection set-up.
NO	Lowercase letters are not passed to DCAM applications.
YES	Lowercase letters are passed to DCAM applications.
LINE	controls LINE messages coming from the application. Default=NO.
NO	LINE messages are displayed on the terminal.
YES	LINE messages are suppressed. This setting is only advisable if a physical or formatted message is output from the start sequence of this application.

PROTOCOL	defines the behavior of OMNIS with respect to a DCAM application. Default= <i>OMNIS</i> .
OMNIS	From the viewpoint of a DCAM application, OMNIS behaves like a program. For a detailed description of this behavior, see page 99 .
DSSIM	From the viewpoint of a DCAM application, OMNIS behaves like a real terminal.
PTAST	Here you can give the name of the P key set created for the application with PLUS. <i>pkeyset</i> must be no more than 8 characters long.
PTNNAME	specifies (together with <i>PRONAME</i>) the application's network address. The default is \$CONSOLE for UCON and \$DIALOG for TIAM.
*PREDIALOG	<i>PTNNAME</i> should be determined by the user in dialog mode before the connection is opened.
RESTART-MESSAGE	defines a restart message to be sent to an application is the <i>SAVE</i> function is not executed for <i>SCREEN</i> .
X'...'	The restart message is entered in hex code (up to 30 characters).
C'...'	The restart message is entered as a character string (up to 60 characters).
ROUTE	specifies whether the connection to the application is to be set up directly or indirectly, or whether a multiplex connection is to be set up. Default= <i>DIRECT</i> .
DIRECT	The connection to the application is to be set up directly.
INDIRECT	The connection to the application is to be set up indirectly (i.e. via an OMNIS present in the target computer).
MUX	A multiplex connection is to be set up to UTM.
SAVE	specifies whether screen contents and/or P keys are saved on quitting the application.
STD	The <i>OPTION</i> or <i>SET</i> parameter setting of OMNIS applies.
ALL	Both the screen contents and the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
PKEY	The P keys are saved individually for each application.

SCREEN	The screen contents are saved and output on the terminal when the application is selected again. (In line mode, the screen is cleared following the first input.) If it has not been possible to save screen contents, the <i>RESTART-MESSAGE</i> function is applied when the application is selected.
SIGNON	defines how BIND data is passed on to UTM. This field is evaluated for multiplex connections only (<i>ROUTE=M</i>). Default= <i>BOTH</i> .
BOTH	If <i>BOTH</i> is specified, the user name (<i>USER</i>) and the password are passed to UTM. The user name and the password must have been generated in UTM.
USER	If <i>USER</i> is specified, the user name (<i>USER</i>) is passed to UTM. This case is meaningful if the user has been generated without a password in UTM (i.e. the password consists of blanks).
NONE	If <i>NONE</i> is specified, no data from the login mask is sent to UTM as BIND data. In this case the automatic sign-on only functions if the UTM application has been generated without a user (<i>USER</i>).
SUPPRESS	If <i>SUPPRESS</i> is specified, no BIND data is sent to UTM even if such data is specified in the <i>Application Menu</i> mask.
SNMP	This operand enables the partner connection to be monitored via SNMP.
YES	The partner connection is monitored via SNMP and OMNIS.
NO	The partner connection is not monitored via SNMP.
TYPE	Here you enter the application type. All the types that OMNIS knows are allowed: TIAM, DCAM, UTM, UCON. Default= <i>TIAM</i> .
*PREDIALOG	<i>TYPE</i> should be determined by the user in dialog mode before the connection is opened.
USER-NUMBER	Freely selectable four-digit number, used in start sequence definitions. Default= <i>0000</i> .

USER-NUMBER-PRIORITY

Here you define the priority of the multiple user number specifications in the OMNISKD. The user number (*USER-NUMBER*) which was specified in *DECLARE-APPLICATION*, *DECLARE-MENU* or *DECLARE-USER* is used to define the start sequences.

USER-MENU-APPLICATION

If there are several user numbers, first the one from *DECLARE-USER* is used to define the start sequences; if it is not defined, the user number from *DECLARE-MENU* is used, and, if this is not defined, the user number from *DECLARE-APPLICATION* is used.

APPLICATION-USER-MENU

If there are several user numbers, first the one from *DECLARE-APPLICATION* is used to define the start sequences; if it is not defined, the user number from *DECLARE-USER* is used, and, if this is not defined, the user number from *DECLARE-MENU* is used.

CMSG

If this parameter is specified, it must come at the end of the definition. *CMSG* (alternative: *LMSG*) defines a connection message to be sent to the partner by OMNIS upon establishing the connection. This message depends on the type of partner; in the case of UCON partners it may, for example, contain the authorization name and password. *connmsg* must be no more than 30 characters long.

See [page 207](#) for an example.

DECLARE-CONSOLE - Declare a BS2000 console

The *DECLARE-CONSOLE* statement is used to declare a BS2000 console which is to execute terminal functions via OMNIS.

Operation	Operands
D[ECLARE]-C[ONSOLE]	console [,CO[MMENT] = 'max-string70'] , { C } { L } M[SG]=authorization-name,'password'

- console specifies the name of the BS2000 console at which OMNIS is to be used. The name can be a device mnemonic in the form (mn) or an authorization name in the form nnnn.
- COMMENT is a comment about the console, up to 70 characters long.
- CMSG (alternative: *LMSG*) specifies which authorization name and password OMNIS is to use for setting up the connection to the BS2000 console. The *CMSG/LMSG* operand is not a genuine keyword operand and must always be the last operand in the operand list.
- LMSG see CMSG.

All connections to BS2000 consoles must be taken into account in the BS2000 system (see also [page 87](#)).

See [page 207](#) for an example.

DECLARE-EXIT - Declare an exit group

The *DECLARE-EXIT* statement is used to declare an exit group.

Operation	Operands
D[ECLARE]-E[XIT]	eac ,M[ODULES]=(modname [...]) [,C[OMMENT] = 'max-string70']

eac defines the exit group's exit address code.

MODULES defines the members of the exit group.

COMMENT is a comment about the exit group, up to 70 characters long.

Using a *DECLARE-EXIT* statement, up to 14 exit routines (modules) can be assigned to one eac. The sum of the exit routines assigned to one eac forms an exit group.

A module may occur only once in an exit group.

The order in which the individual exit routines are defined determines the call sequence. The first exit routine is the one closest to the partner, and the last exit routine is the one closest to the terminal. For outputs, the exit routine closest to the partner is called first and the one closest to the terminal is called last; for inputs, this order is reversed.

An exit group may be assigned to a terminal or partner with *DECLARE-TERMINAL* or *DECLARE-PARTNER*.

See [page 207](#) for an example.

DECLARE-GROUP - Declare a partner group

The *DECLARE-GROUP* statement is used to declare a partner group.

Operation	Operands
D[ECLARE]-G[ROUP]	gac ,ADD=(pac [...]) [,C[OMMENT] = 'max-string70']

gac defines the group address code of the partner group to be declared. Each group address code must be unique for a particular terminal.

ADD defines the members of the group by specification of their partner address codes. Each partner address code must be unique for a particular terminal. Only partners, not other groups, may be members of a partner group. All members of a group should have been declared previously.

If a partner is not declared, they will be added to the group with default attributes.

COMMENT is a comment about the group, up to 70 characters long.

Up to 25 members can be assigned with one *DECLARE-GROUP* statement. The number of members can, however, be increased by further *DECLARE-GROUP* statements with the same gac. These statements must immediately follow each other.

See [page 207](#) for an example.

DECLARE-HARDCOPY - Declare a printer

The *DECLARE-HARDCOPY* statement is used to declare a directly connected printer.

D[ECLARE]-H[ARDCOPY]	<pre> hac ,PR[ONAME]=praname ,PT[NNAME]=ptnname [,COM[MENT] = 'max-string70' [,CON[NECT]= { S[TART] U[SE] } [,FA[ILURE]= { *I[GNORE] *S[AVE] hac } [,U[SE]= { F[REE] D[ECLARED] } </pre>
----------------------	--

hac	defines a hardcopy address code by which the printer (or terminal with <i>LISTENING</i> , see also "Functions and Commands" manual [1]) can be uniquely identified. The hardcopy address code of each printer must be unique "OMNIS-wide".
PRONAME	specifies (together with <i>PTNNAME</i>) the symbolic network address of the printer (or terminal with <i>LISTENING</i> , see also "Functions and Commands" manual [1]).
PTNNAME	specifies (together with <i>PRONAME</i>) the symbolic network address of the printer (or terminal with <i>LISTENING</i> , see also "Functions and Commands" manual [1]).
COMMENT	is a comment about the printer, up to 70 characters long.

CONNECT	specifies when the connection to the printer is to be set up.
START	The connection to the printer is to be set up when OMNIS is started.
USE	The connection to the printer is not to be set up until the printer is needed, i.e. when it is assigned to a partner or terminal.
FAILURE	defines the behavior of OMNIS if the printer fails.
*IGNORE	Messages destined for the printer are not to be output.
*SAVE	Messages destined for the printer are to be written to the paging file and are output when the printer is again available (<i>RESTART</i> command).
hac	Messages destined for the printer are to be output on the alternate device with the hardcopy address code <i>hac</i> . Alternate printer administration is not transitive, i.e. message output on the substitute of the alternate printer itself is not possible. Messages are lost if the alternate printer itself is not operable. A few messages (up to 10) may be lost when a printer fails. These are the messages which OMNIS had already sent to the printer before the failure was detected.
USE	specifies who is permitted to use the printer.
FREE	The printer may be used by all OMNIS users.
DECLARED	The printer may only be used by those OMNIS users declared in the configuration file (using <i>DECLARE-TERMINAL</i> or <i>DECLARE-PARTNER</i>).

See [page 207](#) for an example.

DECLARE-MENU - Declare a menu

the *DECLARE-MENU* statement is used to declare a menu.

Operation	Operands
D[<u>E</u> CLARE]-ME[<u>N</u> U]	NAME = menu_entry [,C[<u>O</u> MMENT] = 'max-string70'] [,E[<u>N</u> TRY]-L[<u>I</u> ST] = (({ [F[<u>K</u> EY] = <u>0</u> ..24] } { [K[<u>K</u> EY] = <u>0</u> ..13] }) [,PA[<u>T</u> H] = ({ menu_entry * *UNLOCK *UPDATE } , pac) [,PK[<u>E</u> Y] = <u>0</u> ..20] [,TE[<u>X</u> T] = 'max-string20'] [,TY[<u>P</u> E] = { AP[<u>P</u> L] } { AD[<u>M</u> I] })] { T[<u>E</u> XT] })] ,...)] [,H[<u>C</u> OPY] = (D[<u>E</u> VICE] = devname , K[<u>K</u> EY] = <u>0</u> ..13 , P[<u>K</u> EY] = <u>0</u> ..20] [,F[<u>O</u> RM] = { STD <form-name> })] [,T[<u>I</u> MEOUT] = <u>0</u> ..1440] [,U[<u>S</u> ER]-N[<u>U</u> MBER] = <u>0</u> ..9999]

You can define a maximum of 900 entries per *DECLARE-MENU* statement.

Meaning of the operands

NAME	specifies the name of the menu assigned to the user. <i>menu_entry</i> has up to 8 characters and acts as a key to the <i>omm.menutab</i> file.
COMMENT	is a comment about the menu, up to 70 characters long.
ENTRY-LIST	defines the menu entries; there can be up to 900 entries per menu.
FKEY	defines the F key that can be used to switch to this application. The available keys are F2 (<i>FKEY=2</i>) through F24 (<i>FKEY=24</i>). <i>FKEY=0</i> means that no key is allocated (=default).
KKEY	defines the K key that can be used to switch to this application. The available keys are K1 (<i>KKEY=1</i>) through K13 (<i>KKEY=13</i>). <i>KKEY=0</i> means that no key is allocated (=default).
PATH	specifies the application to which this entry refers.
<i>menu_entry</i>	The application is a local application. No other menu is allowed or able to use this application. <i>menu_entry</i> is the name of the menu. An asterisk as the first character of <i>menu_entry</i> is reserved for administrator functions.
*	The application is a global application. The key of these applications is (<i>*,pac</i>), where the PAC is globally unique. Global applications can be declared in more than one menu.
*UNLOCK	declares the OMNIS-MENU administrator function *UNLOCK.
*UPDATE	declares the OMNIS-MENU administrator function *UPDATE.
<i>pac</i>	The PAC of the application. The PAC must be globally unique for global applications; for local applications it must be unique within the menu. <i>pac</i> must be no more than four (alphanumeric) characters long.
PKEY	defines the P key that can be used to switch to this application. The available keys are P1 (<i>PKEY=1</i>) through P20 (<i>PKEY=20</i>). <i>PKEY=0</i> means that no key is allocated (=default).
TEXT	describes the menu entry text as it appears on the menu (max. 20 characters). If no text is specified, the first 20 characters from the <i>COMMENT</i> operand in the <i>DECLARE-APPLICATION</i> statement are used. If there is no text in the <i>COMMENT</i> operand, no text is output in the menu. See also page 128 .

TYPE	specifies the type of entry on the menu.
APPL	The menu entry on the user menu is an application.
ADMI	The menu entry is an OMNIS-MENU administrator function.
TEXT	The menu entry is a comment line on the menu.
HCOPY	declares the hardcopy output mechanism.
DEVICE	<i>devname</i> specifies the RSO printer to use.
KKEY	By specifying <i>number</i> , you can define the K key you want to use to start a screen dump and its printout on the <i>devname</i> printer.
PKEY	By specifying <i>number</i> , you can define the P key you want to use to start a screen dump and its printout on the <i>devname</i> printer.
FORM	This operand sets the form for the printer. By specifying <i>STD</i> or <i>form-name</i> (max. 6 characters), you can define the format of the print form to be used on the printer defined with <i>DEVICE</i> . If the form name consists only of blanks, the <i>PRINT</i> command is issued without a form name. If a switchover is made from the specified printer to a directly connected printer with <i>HCOPY=hac</i> , or the print output is suppressed with <i>HCOPY=STD</i> , the form name is automatically set to the standard form <i>STD</i> .
TIMEOUT	Specifies the inactivity time (in minutes) before the connection to a terminal is cleared down. If <i>TIMEOUT=0</i> (=default), there is no time limit. This operand has lower priority than the <i>TIMEOUT</i> operand of the <i>DECLARE-USER</i> statement. A user who requires particular protection can therefore be assigned a shorter time than actually provided for in a menu.
USER-NUMBER	Freely selectable four-digit number, used together with the application number (see <i>DECLARE-APPLICATION APPLICATION-NUMBER=...</i>) in start sequence definitions and as a key to the OMNIS text file. The default of <i>0000</i> means that no start sequence is defined.



K and F keys can be assigned without associated P keys; P keys must be assigned together with K or F keys. Keys F6 through F24 are available on 9756 and later terminals. They can also be entered in the form of ESC sequences. If an application is defined without K or F keys, it will not be assigned any corresponding keys.

Variables for *TEXT='max-string20'*

The following variables are only valid for the OMNISKD statement *DECLARE-MENU*, *TEXT='max-string20'* operand. They will be replaced in the OMMFSEL mask (application selection via OMNIS-MENU).

Variable name	Area of application	Meaning
USERID	OMNIS/OMNIS-MENU	user ID of the dialog application
PTNNAME	OMNIS/OMNIS-MENU	network address of the application
PRONAME	OMNIS/OMNIS-MENU	network address of the application
APPNAME	OMNIS/OMNIS-MENU	name of the “simulated terminal”
TYP	OMNIS/OMNIS-MENU	type of application

Attributes from the DECLARE-APPLICATION statement can be added to the DECLARE-MENU statement with the help of these variables. The OMNIS-MENU user name and the user ID under which a TIAM application has logged on to \$DIALOG can also be displayed. If no values are currently defined for the variables, hyphens (“-”) are displayed instead. No S variables or job variables can be specified.

Example

```
DECL-APPL PATH=(*,UTM1),A-N=0,PR=D016ZE05,T=UTM,PTN=PULSV3
DECL-APPL PATH=(*,TIA1),A-N=4,PR=D016ZE03,T=TIAM,AP=ADMIN316
DECL-APPL PATH=(*,UC01),A-N=6,PR=D016ZE03,T=UCON,LMSG=$CONSOLE,SYSOPR

DECLARE-MENU NAME=MENU1 -
, ENTRY-LIST=( -
  (TYPE=APPL,PATH=(*,UTM1),TEXT='&PRONAME / &PTNNAME') -
  (TYPE=APPL,PATH=(*,TIA1),TEXT='&APPNAME / &USERID ') -
  (TYPE=APPL,PATH=(*,UC01),TEXT='&PRONAME / &TYP ') )
```

Output in the Application Menu:

```
<UTM1> D016ZE05 / PULSV3
<TIA1> ADMIN316 / -----
<UC01> D016ZE03 / UCON
or
<UTM1> D016ZE05 / PULSV3
<TIA1> ADMIN316 / TSOS
<UC01> D016ZE03 / UCON
```


DECLARE-MUX - Declare a multiplex connection

The *DECLARE-MUX* statement is used to declare a multiplex connection.

Operation	Operands
D[<i>ECLARE</i>]-MU[X]	PT[NNAME]=ptnname ,PR[ONAME]=proname [,A[<i>VA</i> ILABILITY]= $\left. \begin{matrix} \underline{N[O]} \\ Y[ES] \end{matrix} \right\}$] [,COM[<i>MENT</i>] = 'max-string70'] [,CON[<i>NECT</i>]= $\left. \begin{matrix} \underline{S[TART]} \\ O[PNCON] \end{matrix} \right\}$] [,CP[<i>ASS</i>]='password'] [,LP[<i>ASS</i>]='password']

- PTNNAME identifies the name of the UTM application to which a multiplex application exists.
- PRONAME identifies the processor on which the UTM application is running.
- AVAILABILITY defines how OMNIS behaves in conjunction with HIPLEX AF; also refer to the "Functions and Commands" manual [1].
 - NO The settings for sessions using this multiplex connection to UTM are not stored by OMNIS.
 - YES If a processor fails, HIPLEX AF makes sure that the UTM application is quickly switched over to another processor system with exactly the same application environment. Once the UTM application reports back to OMNIS, OMNIS makes sure that the sessions are re-established with the same settings as before. Then OMNIS sends OMS0226 to report that the application is available again.
- COMMENT is a comment about the multiplex connection, up to 70 characters long.

CONNECT	specifies when the transport connection to the UTM application is to be set up.
START	The connection is to be set up immediately when OMNIS is started.
OPNCON	The connection is set up before the first request to set up a session is sent to the UTM application.
CPASS	is the password for connection setup. If both <i>CPASS</i> and <i>LPASS</i> are specified in one statement, the value entered last applies.
password	is the password (1..4 characters).
LPASS	see CPASS.

The multiplex declaration must be entered before the first terminal declaration.

OMNISKD checks whether the declarations for all UTM applications to which multiplex connections exist are unique (by comparing *PRONAME* and *PTNNAME*).

If the result of this check is positive, the multiplex entry is initialized and written to the configuration file.

DECLARE-OMNIS - Declare OMNIS

The *DECLARE-OMNIS* statement declares an OMNIS-OMNIS connection.

D[ECLARE]-O[MNIS]	<pre> oac ,PR[ONAME]=praname [,COM[MENT] = 'max-string70'] [,CON[NECT]= { S[TART] O[PNCON] }] [,CP[ASS]='password'] [,LP[ASS]='password'] [,OP[NCON]= { F[REE] D[ECLARED] }] [,PT[NNAME]=ptnname] [,TI[ME]=number] </pre>
-------------------	---

oac	defines an OMNIS address code which uniquely identifies the remote OMNIS. The OMNIS address code of each OMNIS-OMNIS connection must be unique.
PRONAME	specifies (together with <i>PTNNAME</i>) the network address of the remote OMNIS. The same PRONAME may be used only once in the <i>DECLARE-OMNIS</i> statements. It is thus not possible to set up parallel OMNIS-OMNIS connections between two processors.
COMMENT	is a comment about the OMNIS-OMNIS connection, up to 70 characters long.

CONNECT	specifies when the OMNIS-OMNIS connection is to be set up.
OPNCON	The OMNIS-OMNIS connection is set up as soon as an OMNIS user wishes to use it.
START	The OMNIS-OMNIS connection is to be set up when OMNIS is started.
CPASS	specifies a connection password to be used when setting up the OMNIS-OMNIS connection. The password is defined in the ENTER file of the remote OMNIS (with the command <i>SET-DCAM-APPLICATION-LINK</i>). If both <i>CPASS</i> and <i>LPASS</i> are specified in one statement, the value entered last applies.
password	is the connection password (1..4 characters).
LPASS	see CPASS.
OPNCON	defines who may use the OMNIS-OMNIS connection.
DECLARED	The OMNIS-OMNIS connection may be used only by declared partners for indirect connections. Undeclared indirect connections are not possible.
FREE	The OMNIS-OMNIS connection may be used by all OMNIS users. Undeclared indirect connections are possible.
PTNNAME	specifies (together with <i>PRONAME</i>) the network address of the remote OMNIS.
TIME	defines the monitoring period in seconds within which message traffic must take place via the OMNIS-OMNIS connection. If <i>TIME=0</i> , or is not specified, no monitoring takes place. Too short a monitoring time should not be chosen. A time in excess of 30 seconds is recommended. It should be noted that if the remote OMNIS does not react, then twice the monitoring time elapses before a timeout message is issued.



The names and connection passwords of each OMNIS must be coordinated between the various OMNIS administrators.

See [page 207](#) for an example.

DECLARE-PARAMETER - Declare parameters

The *DECLARE-PARAMETER* statement is used to declare global settings for OMNIS-MENU.

Operation	Operands
D[ECLARE]-PARA[METER]	$[C[ALL]-H[OLD] = \left. \begin{array}{l} A[UTO] \\ N[O] \end{array} \right\}]$ $[,CO[MMENT] = 'max-string70']$ $[,C[ONNECTION]-A[TTEMPTS] = \underline{0}..99]$ $[,D[ELETE]-P[KEYS] = \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}]$ $[,D[ISCONNECTION]-M[ASK] = \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}]$ $[,I[MPPLICIT]-A[PPLICATION]-S[WITCH]-K[KEY] = 1..13]$ $[,I[MPPLICIT]-A[PPLICATION]-S[WITCH]-P[KEY] = 0..20]$ $[,I[MPPLICIT]-A[PPLICATION]-S[WITCH]-D[IALOG] = \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}]$ $[,M[ODIFY]-A[UTHORIZATION]-M[ASK] = \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}]$ $[,O[MNIS]-K[KEY] = \underline{0}..13]$ $[,O[MNIS]-P[KEY] = \underline{0}..20]$ $[,O[NE]-U[SER]-O[NE]-C[ONNECTION] = \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}]$

Operation	Operands
	<p data-bbox="602 215 1116 371">[,P[ROTECT]-PW=(length, $\left. \begin{matrix} \text{NONE} \\ \text{MIN} \\ \text{MED} \\ \text{MAX} \end{matrix} \right\} ,\text{time})]$</p> <p data-bbox="602 416 915 450">[,R[ESET]-K[KEY] = 1..13]</p> <p data-bbox="602 495 915 529">[,R[ESET]-P[KEY] = <u>0</u>..20]</p> <p data-bbox="602 574 1180 608">[,S[WITCH]-C[HARACTER]='printable character']</p> <p data-bbox="602 653 924 687">[,T[EXTKEY]-L[ENGTH]=n]</p> <p data-bbox="602 732 878 766">[,T[IMEOUT] = <u>0</u>..1440]</p>

Meaning of the operands

CALL-HOLD defines the HOLD values used when a terminal is disconnected from OMNIS-MENU. This operand applies only with *TIMEOUT*>0. Default=*NO*.

NO The terminal is disconnected with *HALT HOLD=NO*.

AUTO The terminal is disconnected with *HALT HOLD=AUTO*.

COMMENT is a general-purpose comment up to 70 characters long.

CONNECTION-ATTEMPTS defines how many unsuccessful attempts to connect to OMNIS-MENU are allowed at the terminal. If this figure is exceeded, the terminal is disabled. If the operand is not specified, it defaults to 0 (unlimited number of connection attempts). Default=0.

DELETE-PKEYS Specifies whether the P key assignments are discarded on connection/disconnection. Default=*NO*.

NO The P key assignments are not discarded on connection/disconnection.

YES OMNIS-MENU explicitly deletes the P key assignments on connection/disconnection.

DISCONNECTION-MASK

specifies whether OMNIS-MENU is to display the logout mask.

IMPLICIT-APPLICATION-SWITCH-KKEY

specifies the K key for activation or deactivation of implicit application switching.

The default value is K13.

The uniqueness of the K key must be guaranteed.

The same K key must not be specified in the operands OMNIS-KKEY and RESET-KKEY.

IMPLICIT-APPLICATION-SWITCH-PKEY

specifies the P key for activation or deactivation of implicit application switching.

There is no standard value.

The uniqueness of the P key must be guaranteed.

The same P key must not be specified in the operands OMNIS-PKEY and RESET-PKEY.

IMPLICIT-APPLICATION-SWITCH-DIALOG

YES the functionality of implicit application switching applies also in the dialog with partner applications. Compatible with OMNIS V8.3.

NO the functionality of implicit application switching applies only in the OMNIS-MENU selection mask (OMMSEL).

MODIFY-AUTHORIZATION-MASK

determines whether the *login* mask can be modified in OMNIS-MENU. If the value is *NO*, the default mask will be displayed.

OMNIS-KKEY

defines a K key that can be used to switch to OMNIS command mode. The available keys are K1 through K13. *OMNIS-KKEY=0* means that no K key is allocated. The K key will only be used in a menu if it is not already assigned to an application.

The K key for OMNIS command mode must not be confused with the KPAC in OMNIS. No KPAC should be used when OMNIS-MENU is in operation.

This operand is inoperative for users without command mode authorization. Default=*0*.

OMNIS-PKEY

defines a P key that can be used to switch to OMNIS command mode. The available keys are P1 through P20. *OMNIS-PKEY=0* means that no P key is allocated. The P key will only be used in a menu if it is not already assigned to an application.

This operand is inoperative for users without command mode authorization. If *OMNIS-KKEY=0* is set, *OMNIS-PKEY* must not have a nonzero value. Default=*0*.

ONE-USER-ONE-CONNECTION

specifies whether a user can have only one connection at a time to OMNIS-MENU.

YES A user identified by name can have only one connection at a time to OMNIS-MENU, i.e. the user is uniquely identified.

NO A number of users can work with OMNIS-MENU at the same time under the same user name.

PROTECT-PW defines the conditions that must be met by a user's password.

length specifies the minimum length of the password; the value can be between 0 (no password required) and 8.

NONE Any string *length* characters long is a valid password.

MIN The password must not include the same character more than twice in succession. The password must be at least one character long.

MED The password must not include the same character more than twice in succession and must comprise at least one letter and one digit. The password must be at least two characters long.

MAX The password must not include the same character more than twice in succession and must comprise at least one letter, one digit and one special character. Special characters are characters other than A-Z and 0-9. The password must be at least three characters long.

time A value between 0 and 180 can be specified.

A value of 0 means that there is no need to change the password regularly.

A value in the range 1-180 specifies the number of days after which a password must be changed. Two weeks before this period expires, the user will be prompted at logon time to choose a different password. Once the period has expired, the user will no longer be able to log on. A user can assign themselves a new password.

RESET-KKEY defines a K key as a go-back and logout key (equivalent to the quit key). Keys K1 through K13 are available.

RESET-PKEY defines a P key as a go-back and logout key (equivalent to the quit key). Keys P1 through P20 are available. *RESET-PKEY=0* means that no P key will be assigned.

SWITCH-CHARACTER

specifies the printable character must be used before the PAC to switch applications.

'\': no switch character applies.

TEXTKEY- LENGTH

specifies the maximum number of characters in *START-SEQUENCE-NUMBER*. (*START-SEQUENCE-NUMBER* is needs to be specified in the *DECLARE-START-SEQUENCE* statement.)

The default value for *n* is 4. If OMNIS-MENU is being used, *TEXTKEY-LENGTH=6* must be specified. *n* must generally lie in the range $4 \leq n \leq 7$.

TIMEOUT

specifies the time (in minutes) after which the connection to a terminal is cleared down if it remains inactive. With *TIMEOUT=0* (=default) there is no time limit.

This operand has a lower priority than the *TIMEOUT* operands of the *DECLARE-USER* and *DECLARE-MENU* statements, which define in that order the time until the connection is cleared down. This means that if the *TIMEOUT* operand in these statements has a lower value than that in the *DECLARE-PARAMETER* statement, the connection is accordingly cleared down earlier than specified here.

If *TIMEOUT=0* in *DECLARE-PARAMETER*, the *TIMEOUT* operand is not evaluated in *DECLARE-USER* or *DECLARE-MENU*.

DECLARE-PARTNER - Declaring a partner

The *DECLARE-PARTNER* statement is used to declare a partner of a terminal or BS2000 console.

Operation	Operands
D[<u>E</u> CLARE]-PART[<u>N</u> ER]	pac [,AP[PNAME]=appname] [,A[UTO]-L[OGOFF]= $\left. \begin{array}{l} \text{S[TD]} \\ \text{Y[ES]} \\ \text{N[O]} \end{array} \right\}$] [,B[ERID]=authorization[,c'password']] [,CH[ANGE]= $\left. \begin{array}{l} \text{Y[ES]} \\ \text{N[O]} \end{array} \right\}$] [,CL[ASS]= $\left. \begin{array}{l} \text{O[UTPUT]} \\ \text{S[AVE]} \\ \text{D[ELETE]} \end{array} \right\}$] [,CODE= $\left. \begin{array}{l} \text{S[YSCODE]} \\ \text{B[INARY]} \end{array} \right\}$] [,COL[OR]= $\left. \begin{array}{l} \text{N[O]} \\ \text{B[LUE]} \\ \text{C[YAN]} \\ \text{G[REEN]} \\ \text{M[AGENTA]} \\ \text{R[ED]} \\ \text{W[HITE]} \\ \text{Y[ELLOW]} \end{array} \right\}$] [,COM[MENT] = 'max-string70']

Operation	Operands
	$[,CON[NECT]=\left\{\begin{array}{l} O[PNCON] \\ L[OGON] \\ S[TART] \end{array}\right\}]$ <p data-bbox="555 354 825 388">[,CP[ASS]='password']</p> $[,EX[IT]=\left\{\begin{array}{l} S[TD] \\ N[O] \\ eac \end{array}\right\}]$ $[,HC[OPY]=\left\{\begin{array}{l} S[TD] \\ Y[ES] \\ N[O] \\ hac \end{array}\right\}]$ $[,HO[LD]=\left\{\begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \end{array}\right\}]$ $[,KP[AC]=\left\{\begin{array}{l} num \\ N[O] \end{array}\right\}]$ $[,LC[ASE]=\left\{\begin{array}{l} Y[ES] \\ N[O] \end{array}\right\}]$ $[,LIN[E25]=\left\{\begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \\ O[MNIS] \end{array}\right\}]$ $[,LO[GGING]=\left\{\begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \end{array}\right\}]$ <p data-bbox="555 1491 821 1525">[,LP[ASS]='password']</p>

Operation	Operands
	<p>[,MT[AB]=$\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{name} \end{array} \right\}$]</p> <p>[,OP[ASS]='password']</p> <p>[,PAC=$\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ L[INE] \\ P[REFIX] \end{array} \right\}$]</p> <p>[,PR[ONAME]=$\left\{ \begin{array}{l} \text{prname} \\ *ANY \end{array} \right\}$]</p> <p>[,PROT[OCOL] = $\left\{ \begin{array}{l} \underline{O[MNIS]} \\ D[SSIM] \end{array} \right\}$</p> <p>[,PT[NNAME]=ptnname]</p> <p>[,RO[UTE]=$\left\{ \begin{array}{l} \underline{D[IRECT]} \\ I[NDIRECT] \\ M[UX] \end{array} \right\}$]</p> <p>[,SA[VE] = $\left\{ \begin{array}{l} \underline{S[TD]} \\ A[LL] \\ N[O] \\ P[KEY] \\ SC[REEN] \end{array} \right\}$]</p> <p>[,SN[MP]=$\left\{ \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}$]</p> <p>[,S[TART]-S[EQUENCE]=number]</p>

Operation	Operands
	$[, T[YP]= \left\{ \begin{array}{l} \underline{TIAM} \\ DCAM \\ UCON \\ UTM \end{array} \right\}]$ $[, \left\{ \begin{array}{l} C \\ L \end{array} \right\} M[SG]=\text{connection-message}]$

- pac** specifies which partner address code is to be assigned to the partner. Each partner address code must be unique for a particular terminal.
- APPNAME** defines the name of the "simulated terminal" under which OMNIS sets up the connection to the partner. OMNIS is known to the partner by this name. The APPNAME of each partner must be unique within the system. This applies to each OMNIS which is loaded in a system. OMNISKD does not check whether this condition is fulfilled.

If the value \$\$\$\$\$\$ is given for APPNAME (default value), this means that OMNIS generates the name itself on the basis of the prefix (three characters, see the APPLICATION statement in the "Administration and Programming" manual [1]) followed by a 5-digit number assigned consecutively by OMNIS.
- AUTO-LOGOFF** specifies whether *LOGOFF* is to be sent to the TIAM partner after a start sequence has been processed. The operand *AUTO-LOGOFF* is evaluated only if a start sequence is assigned to the partner (with the operand *START-SEQUENCE* in *DECLARE-PARTNER* or *OPNCON*).
- YES** After the start sequence has been processed, *LOGOFF* is sent to the partner.
- NO** The connection to the partner is maintained even after the start sequence has been processed.
- STD** The setting for *AUTO-LOGOFF* specified with *DECLARE-TERMINAL* applies.
- BERID** specifies the authorization and password generated in the SKP. The *BERID* operand is not a genuine keyword operand and must always be the last operand in the operand list.
- authorization authorization generated in the SKP (1-8 characters)
- password password generated in the SKP (1-8 characters in the form *c'password'*).

CHANGE	defines the authorization for the <i>PARAM</i> and <i>OPNCON</i> commands.
YES	The user may modify the partner's characteristics declared with <i>DECLARE-PARTNER</i> , using the <i>PARAM</i> or <i>OPNCON</i> command.
NO	The user may not alter the partner's characteristics declared with <i>DECLARE-PARTNER</i> . The <i>PARAM</i> command is rejected and only the partner address code may be specified in the <i>OPNCON</i> command.
CLASS	defines the message class to which the partner is to be assigned.
OUTPUT	The partner is to be assigned to the class OUTPUT; i.e. all its messages are displayed immediately at the terminal.
DELETE	The partner is to be assigned to the class DELETE; i.e. all its messages are ignored and not displayed at the terminal.
SAVE	The partner is to be assigned to the class SAVE, i.e. its messages are not displayed immediately, but written to the logging file, where they can be read by page-turning.
CODE	specifies whether the partner sends data in EBCDIC (<i>SYSCODE</i>) or binary mode (<i>BINARY</i>).
COLOR	specifies the color in which the partner's messages are to be output, as follows:
NO	OMNIS does not insert color control characters in the messages.
BLUE	The messages are to be output in blue.
CYAN	The messages are to be output in cyan.
GREEN	The messages are to be output in green.
MAGENTA	The messages are to be output in magenta.
RED	The messages are to be output in red.
WHITE	The messages are to be output in white.
YELLOW	The messages are to be output in yellow.
COMMENT	is a comment about the partner, up to 70 characters long.
CONNECT	specifies when the connection to the partner is to be set up.
LOGON	The connection to the partner is to be set up automatically by OMNIS as soon as the connection to the terminal exists.
START	The connection to the partner is to be set up immediately after OMNIS is started or as soon as the connection to the terminal exists.

OPNCON	The connection is to be set up explicitly by the user by means of an OPNCON command.
CPASS	defines a password which is to be used when setting up a connection to the partner. Use of this password depends on the type of partner; for TIAM, for example, no connection password is needed. If both <i>CPASS</i> and <i>LPASS</i> are specified in one statement, the value entered last applies.
password	Connection password (1..4 characters).
EXIT	defines an exit group. Note that OMNISKD cannot check whether an exit routine actually exists.
NO	The partner's message traffic is not to pass via an exit group.
eac	The partner's message traffic is to pass via the exit group <i>eac</i> .
STD	The passing of messages to an exit group is defined in the <i>OPTION</i> command or <i>DECLARE-TERMINAL</i> statement.
HCOPI	defines message output on the terminal printer assigned to the terminal.
YES	The partner's messages are to be displayed on the data display terminal and printed simultaneously on the terminal printer.
NO	The partner's messages are to be sent to the data display terminal only. Output on the terminal printer is suppressed, even if it is explicitly requested by the partner.
hac	The partner's messages are to be output on the directly connected printer with the address code <i>hac</i> .
STD	Terminal printer output is defined by the <i>OPTION</i> command or <i>DECLARE-TERMINAL</i> statement.
HOLD	defines the behavior of OMNIS after a loss of connection to the terminal. If <i>SECURITY LEVEL≠LOW</i> , <i>HOLD=NO</i> always applies.
NO	The connection to the partner is to be cleared down if the connection to the terminal is lost (implicit <i>CLSCON</i>).
YES	The connection to the partner is to be retained if the connection to the terminal is lost. The <i>OPASS</i> operand must be specified for <i>HOLD=YES</i> . Connections to partners who have read in chip cards are always cleared down if the connection to the terminal is lost.
STD	The behavior on loss of connection to the terminal is defined by the <i>OPTION</i> command or <i>DECLARE-TERMINAL</i> statement.

KPAC	defines a K key to be assigned to the partner as a partner address code and to be used to switch between partners in formatted mode.
num	is the number of this K key in the range 1 through 14. For UTM partners without a multiplex connection, this key triggers a <i>KDCLAST</i> , for UTM partners with a multiplex connection it triggers a <i>REPEAT</i> , and for other partners it initiates simulation of a loss of connection to the terminal.
NO	No K key is defined.
LCASE	specifies whether lowercase letters are to be passed on to the partner or not. This operand is interpreted only for DCAM partners. Does not have any effect on the connection message during connection set-up.
YES	Lowercase letters are passed on to the partner.
NO	Lowercase letters are not passed on to the partner; they are converted into uppercase letters.
LINE25	controls the use of the 25th line on the screen for the partner information (PAC, PID, partner/host name, partner type, route, K-key, user ID).
STD	The use of the 25th line on the screen is defined on a terminal-specific basis (<i>OPTION, DECLARE-TERMINAL</i>).
NO	The 25th line on the screen is not used for partner information. If you have switched to <i>LINE25=NO</i> , OMNIS can reset the 25th line on the screen at the next change of partner, provided you are using a VTSU version as of V13.0.
YES	Partner information is written on the 25th line on the screen. OMNIS generates the 25th line on the screen only on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will no longer be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, e.g. from formatted to line mode and, for the 9763 terminal, as a result of a status query, e.g. on <i>LOGON</i> or <i>START-EDT</i> .
OMNIS	Partner information is written on the 25th line on the screen. OMNIS generates the 25th line on the screen on changing partners and whenever this line is modified by the set partner (at the time of the next output).
LMSG	See CMSG.

LOGGING	controls the logging of messages to and from the partner in the logging file.
NO	Messages are not to be logged in the logging file.
YES	Messages are to be logged in the logging file.
STD	The logging of messages is defined for each terminal separately (<i>OPTION, DECLARE-TERMINAL</i>).
LPASS	see <i>CPASS</i> .
MTAB	defines a message table with which the partner's output is to be processed. <i>MTAB</i> is relevant only for UCON partners. If a message table is assigned to the partner, this table must be defined in the OMNIS startup file.
NO	The partner's output is not to be processed by any message table.
name	The partner's output is to be processed by the message table <i>name</i> .
STD	Processing of the partner's output with a message table is defined for each terminal separately (<i>OPTION, DECLARE-TERMINAL</i>).
OPASS	defines an <i>OCCUPY</i> password. This password is needed when a partner is taken over with the <i>OCCUPY</i> command.
password	<i>OCCUPY</i> password (1..4 characters).
PAC	defines output of the partner address code for line mode.
PREFIX	The partner address code is to be displayed in the same line as the partner's message (as a prefix).
LINE	The partner address code is to be displayed in a separate line (together with the partner's name and type) before the message itself.
NO	The partner address code is not to be displayed.
STD	Partner address code output is defined for each terminal separately (<i>OPTION, DECLARE-TERMINAL</i>).
PRONAME	specifies (together with <i>PTNNAME</i>) the partner's symbolic network address. If <i>PRONAME</i> is not specified, OMNIS assumes that the partner is located on the processor on which OMNIS is loaded.
*ANY	* <i>ANY</i> causes the required processing load to be distributed over multiple BS2000 hosts: OMNIS determines the two hosts with the lowest loads in the default job class and tries to set up a connection to the host with the lowest load.

This "load balancing" on multiple hosts by OMNIS is offered for the partner types TIAM and DCAM. The load balancing option is not available for indirect partners.

PROTOCOL	defines the behavior of OMNIS with respect to a DCAM application. The default is <i>OMNIS</i> . The <i>PROTOCOL</i> operand is not visible at the exit interface.
OMNIS	OMNIS behaves, from the viewpoint of a DCAM application, like a program.
DSSIM	OMNIS behaves, from the viewpoint of a DCAM application, like a real terminal. For DCAM partners with <i>PROTOCOL=DSSIM</i> , only <i>CONNECT=LOGON</i> or <i>OPNCON</i> is meaningful, since only these settings ensure correct processing of messages.
PTNNAME	specifies (together with <i>PRONAME</i>) the partner's symbolic network address. The default is \$DIALOG for <i>TYP=TIAM</i> and \$CONSOLE for <i>TYP=UCON</i> . <i>PTNNAME</i> is a mandatory operand for all other partner types.
ROUTE	specifies whether the connection to the partner is to be set up directly or indirectly, or whether a multiplex connection is to be set up.
DIRECT	The connection to the partner is to be set up directly.
INDIRECT	The connection to the partner is to be set up indirectly (i.e. via an OMNIS present in the target computer). With <i>ROUTE=INDIRECT</i> , the <i>PRONAME</i> operand must be specified.
MUX	A multiplex connection to the partner is to be set up. The <i>PRONAME</i> operand must be specified.
SAVE	specifies whether screen contents and/or P keys are saved on quitting the application.
STD	Saving is defined individually for each terminal.
ALL	Both the screen contents and the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
PKEY	The P keys are saved individually for each application.
SCREEN	The screen contents are saved and output on the terminal when the application is selected again. (In line mode, the screen is cleared following the first input.)
SNMP	This operand enables the partner connection to be monitored via SNMP.
YES	The partner connection is monitored via SNMP and OMNIS.

NO	The partner connection is not monitored via SNMP.
START-SEQUENCE	<p><i>number</i> is an integer which specifies the first n digits ($4 \leq n \leq 7$) in the key of the start sequence in the text file to be processed following connection setup. n is defined using <i>LENGTH TEXTKEY=n</i> (see page 54). When specifying <i>START-SEQUENCE</i>, leading zeros can be omitted. If $n=7$, values 1 through 9999999 are permissible for <i>START-SEQUENCE</i>; if $n=4$, values between 1 and 9999 are permitted. OMNISKD cannot check whether <i>TEXTKEY</i> matches the configuration during OMNIS operation (with the <i>OPNCON</i> command, the value specified for <i>START-SEQUENCE</i> is compared with <i>TEXTKEY</i>). The OMNISTC utility enables text files to be converted to various key lengths (see page 247).</p>
TYP	defines the type of partner.
TIAM	The partner is the system application \$DIALOG (timesharing).
UCON	The partner is the system application \$CONSOLE (operating).
DCAM	The partner is a DCAM or PDN application.
UTM	The partner is a UTM application.
CMSG	(alternative: <i>LMSG</i>) defines a connection message to be sent to the partner by OMNIS upon establishing the connection. This message depends on the type of partner; in the case of UCON partners it may, for example, contain the prefix "\$CONSOLE", the authorization name and the password. The operand <i>CMSG/LMSG</i> is not a genuine keyword operand and must always be the last operand in the operand list. In return the connection message may contain any desired characters (including commas).

Dependency of operand specifications on partner type

Operand	TIAM	UTM	DCAM	UCON
PRONAME	*	*	*	*
PTNNAME	o	x	x	o
CPASS/LPASS	-	o	o	o
CMSG/LMSG	-	-	o	o
CODE	-	-	o	-
MTAB	-	-	o	o
BERID	-	-	-	-

- Operand may not be specified
- x Operand must be specified
- o Operand may be specified
- * Operand must be specified for indirect connections and may be specified for direct connections

See [page 207](#) for an example.

DECLARE-PKEYSET - Declare a P key set

The *DECLARE-PKEYSET* statement is used to declare a P key set. On the basis of the statement OMNISKD generates a PLUS input file.

OMNISKD creates a P key set in the PLUS input file for each declaration, generating the control statement *DEL* before each P key set and the control statement *SAV* after each P key set. The PLUS input file begins with the *TY55* statement and ends with *STOP*. You use PLUS to generate the P key library from the PLUS input file.

Before calling OMNISKD you assign the PLUS input file using the following statement:

```
/SET-FILE-LINK LINK-NAME=OMMPLUS, FILE-NAME= <full-filename 1 .. 54>
```

If the *SET-FILE-LINK* statement is omitted, OMNISKD attempts to open the file *omm.plus.in*. The PLUS input file must be processed with a PLUS procedure.

Operation	Operands
D[ECLARE]-PK[EYSET]	PK[EYLIB]=library ,PT[AST]=p-key-name [,COM[MENT] = 'max-string70' [,M[SG]='message'] [,P1='contents of key P1'] ... [,P20='contents of key P20']

- PKEYLIB** specifies the name of the library (no more than 54 characters) containing the P key sets generated by PLUS.
- PTAST** specifies the name of the P key set (no more than 8 characters). See the "TIAM" manual [14] for more detailed information.
- COMMENT** is a comment up to 70 characters long. Since PLUS cannot assign a comment to a P key set, OMNISKD does not evaluate this operand.
- MSG** defines the message (no more than 255 characters) displayed on the terminal when the P key set is loaded.
- P1, ..., P20** determines the contents of key P_n (1 ≤ n ≤ 20, no more than 255 characters). See the "TIAM" manual [14] for more detailed information. In order to prevent a P key assignment from being deleted, you must allocate the value * to the corresponding key.

See [page 207](#) for an example.

DECLARE-START-SEQUENCE - Declare a start sequence

The *DECLARE-START-SEQUENCE* statement is used to declare a start sequence for the text file.

Before calling OMNISKD you assign the text file (ISAM file *KEYLEN=8*) as follows:

```
/CREATE-FILE FILE-NAME=<full-filename 1..54>,SUPPRESS-ERRORS=*FILE-EXISTING
/MODIFY-FILE-ATTRIBUTES FILE-NAME=<full-filename 1..54>
/SET-JOB-STEP
/SET-FILE-LINK LINK-NAME=SOMSTEXT,FILE-NAME=<full-filename 1..54>, -
/ ACCESS-METHOD=ISAM(KEY-LEN=8)
```

Operation	Operands
D[ECLARE]-S[TART]-S[EQUENCE]	S[TART]-S[EQUENCE]-N[UMBER]=number ,R[ECORDS]=('record',...) [,COM[MENT] = 'max-string70']

START-SEQUENCE-NUMBER

Here you specify a start sequence number with between 4 and 7 digits. OMNIS/OMNIS-MENU accepts leading zeros.

number is an integer specifying the first n ($4 \leq n \leq 7$) digits in the key of the start sequence to be processed following connection setup, as given in the text file. The number of digits is defined by the *TEXTKEY-LENGTH* operand of the *DECLARE-PARAMETER* statement; see [page 133](#). If $n=7$, the permitted values for *START-SEQUENCE-NUMBER* are between 1 and 9999999.

An entry of *START-SEQUENCE-NUMBER=0* is allowed, but cannot be assigned. Such records are therefore treated as comments. OMNISKD cannot check whether *TEXTKEY* matches the configuration during OMNIS operation (in the *OPNCON* command, the value specified for *START-SEQUENCE* is compared with *TEXTKEY*). The OMNISTC program allows you to convert text files to different key lengths (see [page 247](#)).

RECORDS is a list of start sequence statements (no more than 255 characters each; single quotes, if any, must be duplicated).

COMMENT is a comment about the start sequence, with up to 70 characters.

See [page 207](#) for an example.

DECLARE-TAC - Declare an entry in the TAC table

The *DECLARE-TAC* is used to declare an entry in the TAC allocation table.

A TAC allocation table has to be present in order for the "implicit application switch" function to be used. On generating the TAC allocation table refer to the "Functions and Commands" manual [1].

Before calling OMNISKD you assign the TAC allocation table with the following command:
/SET-FILE-LINK LINK-NAME=OMMTAC, FILE-NAME= <full-filename 1..54>

Operation	Operands
D[ECLARE]-TA[C]	T[ACNAME] = EBCDIC-character - ' ','" ,PR[ONAME] = proname ,PTN[NAME] = ptnname [,COM[MENT] = 'max-string70' [,M[ENUNAME] = menu_entry [,S[WITCH] = $\left. \begin{array}{l} \text{SE[ND]} \\ \text{SW[ITCHONLY]} \end{array} \right\}]$

Meaning of the operands

TACNAME	defines the TAC (transaction code) or group TAC. A TAC is up to 8 characters long (EBCDIC characters, no blanks or commas). TACs with the same initial character can be combined to form TAC groups. The character string identifying a group is between 1 and 4 characters long and is always followed by an asterisk '*'. Use of TAC groups may make it easier to create the allocation table. The TAC (or group TAC) must be unique in the TAC allocation table in relation to <i>MENUNAME</i> .
PRONAME	specifies (together with <i>PTNNAME</i>) the application's network address.
PTNNAME	specifies (together with <i>PRONAME</i>) the application's network address.
COMMENT	is an all-purpose comment up to 70 characters long.

- MENUNAME** defines the scope of the TAC.
- If a menu name is specified, the entry applies only to that menu. Otherwise the specified TAC applies to all menus.
- If the same TAC exists once with and once without a menu name, only the entry with the menu name applies.
- SWITCH** defines the behavior in the event of an implicit application switch.
- SWITCHONLY** means "switch only". The screen contents are restored.
- SEND** means "switch and send (the user input)".
- With **SWITCH=SEND** you should use the OMNIS "Save screen" function (if the application referenced by the TAC does not send the entire screen).
- Note that the new application also needs to be able to understand the associated TAC.

See [page 207](#) for an example.

DECLARE-TERMINAL - Declare a terminal

The *DECLARE-TERMINAL* statement is used to declare a terminal.

Operation	Operands
D[ECLARE]-TE[RMINAL]	PR[ONAME]=proname ,PT[NNAME]=ptnname [,ACK= $\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ L[INE] \end{array} \right\}$]
	[,ADM= $\left. \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}$] [,A[UTO]-L[OGOFF]= $\left. \begin{array}{l} \underline{S[TD]} \\ Y[ES] \\ N[O] \end{array} \right\}$] [,B[REAK]-K[EY]= $\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{num} \end{array} \right\}$] [,C[ALL]-I[NF]= $\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array} \right\}$] [,C[ALL]-K[EY]= $\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{num} \end{array} \right\}$] [,CH[ANGE]= $\left. \begin{array}{l} \underline{Y[ES]} \\ N[O] \end{array} \right\}$]

Operation	Operands
	<p> $[,\text{COL}[\text{OR}] = \left. \begin{array}{l} \text{N}[\text{O}] \\ \text{B}[\text{LUE}] \\ \text{C}[\text{YAN}] \\ \text{G}[\text{REEN}] \\ \text{M}[\text{AGENTA}] \\ \text{R}[\text{ED}] \\ \text{W}[\text{HITE}] \\ \text{Y}[\text{ELLOW}] \end{array} \right\}]$ </p> <p> $[,\text{COM}[\text{MENT}] = \text{'max-string70'}$ </p> <p> $[,\text{CON}[\text{NECT}] = \left. \begin{array}{l} \text{L}[\text{OGON}] \\ \text{S}[\text{TART}] \end{array} \right\}]$ </p> <p> $[,\text{D}[\text{ISPLAY}] - \text{M}[\text{ODE}] = \left. \begin{array}{l} \text{S}[\text{Y}[\text{STEM}]] \\ \text{M}[\text{IXED}] \\ \text{O}[\text{MNIS}] \\ \text{S}[\text{T}[\text{D}]] \end{array} \right\}]$ </p> <p> $[,\text{E}[\text{XCLUSIVE}] - [\text{PARTNER}] = \left. \begin{array}{l} \text{N}[\text{O}] \\ \text{Y}[\text{ES}] \\ \text{S}[\text{TD}] \end{array} \right\}]$ </p> <p> $[,\text{EX}[\text{IT}] = \left. \begin{array}{l} \text{S}[\text{TD}] \\ \text{N}[\text{O}] \\ \text{eac} \end{array} \right\}]$ </p> <p> $[,\text{HC}[\text{OPY}] = \left. \begin{array}{l} \text{S}[\text{TD}] \\ \text{N}[\text{O}] \\ \text{Y}[\text{ES}] \\ \text{hac} \\ (\text{D}[\text{EVICE}] = \text{devname}, \text{KK}[\text{EY}] = \text{num}) \\ \quad [\text{PK}[\text{EY}] = \text{num}] \\ \quad [\text{FO}[\text{RM}] = \text{STD} \text{form-name}] \end{array} \right\}]$ </p>

Operation	Operands
	$[,HO[LD]=\left\{\begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ A[UTO] \end{array}\right\}]$
	$[,I[INPUT]-L[OGGING]=\left\{\begin{array}{l} \underline{S[TD]} \\ R[EC] \\ SE[ND] \\ B[OTH] \end{array}\right\}]$
	$[,I[INPUT]-S[AVE]=\left\{\begin{array}{l} ([K]num1..14 \\ Fnum 1..24 \\ ([K]num1..14[,qty1..16]) \\ (Fnum1..24[,qty1..16]) \\ \underline{S[TD]} \\ N[O] \end{array}\right\}]$
	$[,KP[AC]=\left\{\begin{array}{l} \underline{S[TD]} \\ num \\ N[O] \end{array}\right\}]$
	$[,LIN[E25]=\left\{\begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ O[MNIS] \end{array}\right\}]$
	$[,LIS[TENING]=\left\{\begin{array}{l} N[O] \\ hac \end{array}\right\}]$
	$[,LO[GGING]=\left\{\begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array}\right\}]$

Operation	Operands
	<p data-bbox="568 215 817 310">[,MT[AB]=$\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{name} \end{array} \right\}$]</p> <p data-bbox="568 352 840 386">[,OP[ASS]='password']</p> <p data-bbox="568 428 947 529">[,OP[NCON]=$\left\{ \begin{array}{l} \underline{F[REE]} \\ D[ECLARED] \end{array} \right\}$]</p> <p data-bbox="568 571 1014 730">[,O[UTPUT]-L[OGGING]=$\left\{ \begin{array}{l} \underline{S[TD]} \\ R[EC] \\ SE[ND] \\ B[OTH] \end{array} \right\}$]</p> <p data-bbox="568 806 826 966">[,PAC=$\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ L[INE] \\ P[REFIX] \end{array} \right\}$]</p> <p data-bbox="568 1008 940 1100">[,R[EPLY]=$\left\{ \begin{array}{l} A[LL] \\ R[ESTRICTED] \\ S[TD] \end{array} \right\}$]</p> <p data-bbox="568 1150 887 1310">[,SA[VE] = $\left\{ \begin{array}{l} \underline{S[TD]} \\ A[LL] \\ N[O] \\ P[KEY] \\ SC[REEN] \end{array} \right\}$]</p> <p data-bbox="568 1386 819 1478">[,SN[MP]=$\left\{ \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}$]</p>

Operation	Operands
	$[\text{TE}[\text{RMTYP}]=\left. \begin{array}{l} \underline{8160} \\ 8161 \\ 8162 \\ 9749 \\ 9750 \\ 9751 \\ 9752 \\ 9753 \\ 9754 \\ 9755 \\ 9756 \\ 9763 \\ \text{APPL} \\ \text{SKP} \end{array} \right\}]$ $[\text{T}[\text{RANSPO}]\text{-P}[\text{ROTOCOL}]=\left. \begin{array}{l} \underline{\text{N}[\text{EA}]} \\ \text{I}[\text{SO}] \end{array} \right\}]$ $[\text{U}[\text{SER}]\text{-P}[\text{ROTOCOL}]=\left. \begin{array}{l} \underline{\text{O}[\text{MNIS}]} \\ \text{N}[\text{O}] \\ \text{V}[\text{TSUCB}] \end{array} \right\}]$

- PRONAME** defines (together with *PTNNAME*) the terminal's network address. The network address specified with *PRONAME/PTNNAME* must be unique within the network. A terminal cannot be declared more than once.
- PTNNAME** defines (together with *PRONAME*) the terminal's network address. The network address specified with *PRONAME/PTNNAME* must be unique within the network. A terminal cannot be declared more than once.
- ACK** defines the screen overflow control.
- NO** Screen overflow control is switched off (hence asynchronous message output at the terminal).
 - YES** Screen overflow control is switched on (messages appear a screen at a time at the terminal).

STD	Screen overflow control is defined by the <i>SET</i> command.
LINE	The behavior of screen overflow control is brought into line with the behavior of TIAM. In contrast to TIAM, the <i>ACK=YES</i> setting results in @:ACK if a switchover to format mode is to take place after outputs in line mode. The user therefore has the option of taking note of the line mode outputs before the format output clears the screen. However, if the line outputs are to be ignored, like, for example, the messages BLS0500 and BLS0552 after the startup of the EDT program which immediately switches to format mode, the <i>ACK=LINE</i> setting enables @:ACK to be restricted to the overflow in line mode.
ADM	defines command authorization. With <i>SECURITY LEVEL=HIGH</i> , <i>ADM=NO</i> always applies.
YES	All commands (including administration commands) may be entered at the terminal.
NO	No administration commands may be entered at the terminal without <i>APASS</i> .
AUTO-LOGOFF	specifies whether <i>LOGOFF</i> is to be sent to the terminal's TIAM partners once the relevant start sequence assigned with <i>DECLARE-TERMINAL</i> or <i>OPNCON</i> has been processed. The <i>AUTO-LOGOFF</i> operand is evaluated only for partners for whom <i>AUTO-LOGOFF=STD</i> is set in the <i>DECLARE-PARTNER</i> statement.
YES	<i>LOGOFF</i> is sent to the TIAM partner(s) of the terminal after the relevant start sequence has been processed.
NO	The connection is maintained after the relevant start sequence has been processed.
STD	The default setting is <i>NO</i> .
BREAK-KEY	defines a K key which is to generate a <i>BREAK</i> for \$DIALOG partners.
num	The value range for <i>num</i> is 1..14.
NO	No <i>BREAK-KEY</i> is defined.
STD	The value set globally for <i>BREAK-KEY</i> with the <i>SET</i> command applies.
CALL-INF	specifies whether or not the CALL state is to be indicated by a system message.
YES	CALL state is to be indicated by a message.
NO	CALL state is not to be indicated by a message.
STD	The value set globally for <i>CALL-INF</i> with the <i>SET</i> command applies.


CALL-KEY	defines a K key which is to result in a CALL.
num	is the K key which is to result in a CALL. The value range for num is 1..14.
NO	No CALL-KEY is defined.
STD	The value set globally for <i>CALL-KEY</i> with the <i>SET</i> command applies.
CHANGE	defines the authorization for the <i>OPTION</i> command.
YES	The user can use the <i>OPTION</i> command to alter the terminal characteristics defined with <i>DECLARE-TERMINAL</i> .
NO	The user cannot alter the terminal characteristics defined with <i>DECLARE-TERMINAL</i> ; the <i>OPTION</i> command is rejected.
COLOR	specifies the color in which OMNIS messages are to be output, as follows:
NO	OMNIS does not insert color control characters in the messages.
BLUE	The messages are to be output in blue.
CYAN	The messages are to be output in cyan.
GREEN	The messages are to be output in green.
MAGENTA	The messages are to be output in magenta.
RED	The messages are to be output in red.
WHITE	The messages are to be output in white.
YELLOW	The messages are to be output in yellow.
COMMENT	is a comment about the terminal, up to 70 characters long.
CONNECT	specifies when the connection to the terminal is to be set up.
LOGON	The connection is to be set up from the terminal.
START	The connection is to be set up immediately after OMNIS is started. If the connection setup is unsuccessful with <i>CONNECT=START</i> , the terminal automatically switches to the <i>HOLD</i> state. With <i>HOLD=YES/AUTO</i> , active partners continue to remain active.
DISPLAY-MODE	defines the colors for UCON outputs.
SYSTEM	The entire message is output in accordance with system conventions. Messages beginning with % are output in yellow, those beginning with ? are output in red and all others are output in white.
MIXED	The message prefix is output in the partner's color, and the rest in accordance with system conventions.

OMNIS	The entire message is output in the partner's color.
STD	The color for UCON outputs is defined globally (<i>SET</i>).
EXCLUSIVE-PARTNER	controls the message flow.
NO	All messages relating to all partners are output on the terminal.
STD	The SET value for OMNIS applies.
YES	<p>While you are engaged in dialog with one partner, only that partner's messages are output. All messages from all other partners are buffered and not output on the terminal.</p> <p>All messages requesting inputs are announced, by means of system message <code>OMS0215</code> in the 25th screen line.</p> <p>Only the first output message is announced in each case, by means of system message <code>OMS0214</code> in the 25th screen line.</p> <p>In the case of groups, only the messages relating to the first partner in the group are output, all messages from all other partners are buffered.</p> <p>When <i>EXCLUSIVE-PARTNER=YES</i> is specified, all partners behave as if <i>CLASS=OUTPUT</i>, regardless of their <i>CLASS</i> value.</p>
EXIT	<p>defines an exit group. If an exit group is assigned to the terminal, this exit group must have been declared previously (with the <i>DECLARE-EXIT</i> statement). <i>EXIT</i> is only effective for system messages, commands and messages to and from partner(s) which have been declared with <i>EXIT=STD (DECLARE-PARTNER)</i>.</p>
NO	The terminal's message traffic is not to proceed via an exit group.
eac	The terminal's message traffic is to be passed through the exit group <i>eac</i> .
STD	The transfer of messages to an exit group is defined in the <i>SET</i> command.
HCOPY	<p>defines message output on the terminal printer assigned to the terminal. The <i>HCOPY</i> entry is evaluated only for OMNIS messages and for messages from partners declared with <i>HCOPY=STD</i>.</p>
YES	All messages are to be displayed on the terminal and printed simultaneously on the terminal printer.
NO	All messages are to be displayed on the terminal only. Printer output is suppressed, even if explicitly requested by the partner.
hac	The terminal's messages are to be output on the directly connected printer with the address code <i>hac</i> .

STD	Terminal printer output is defined by the <i>SET</i> command.
DEVICE	Using <i>devname</i> you specify a printer on which the screen dump is to be printed. Instead of <i>devname</i> , you may also specify <i>CENTRAL</i> . OMNIS also accepts the command if the printer name is not known in the system. The name of the file used is as follows: $S. \left. \begin{array}{l} \textit{user} \\ \textit{devname} \end{array} \right\} .\textit{date.time.tid}$ <p>where the user name <i>user</i> is needed when using OMNIS-MENU, otherwise the terminal name. <i>date</i> and <i>time</i> refer to the time of file opening.</p>
KEY	By specifying <i>num</i> , you define a K key which you can use to initiate a screen dump and printout on the printer <i>devname</i> .
FORM	This operand sets the form for the printer. By specifying <i>STD</i> or <i>form-name</i> (max. 6 characters), you can define the format of the print form to be used on the printer defined with <i>DEVICE</i> . If the form name consists only of blanks, the <i>PRINT</i> command is issued without a form name. If a switchover is made from the specified printer to a directly connected printer with <i>HCOPY=hac</i> , or the print output is suppressed with <i>HCOPY=STD</i> , the form name is automatically set to the standard form <i>STD</i> .
HOLD	defines the behavior of OMNIS after a loss of connection to the terminal. The <i>HOLD</i> entry is evaluated only for partners declared with <i>HOLD=STD (DECLARE-PARTNER)</i> .
NO	All connections to the terminal's partners are to be cleared down on loss of connection to the terminal (implicit <i>CLSCON</i>).
YES	The connections to the terminal's partners are to be retained on loss of connection to the terminal. The <i>OPASS</i> operand must be specified for <i>HOLD=YES</i> . <i>HOLD=YES</i> is evaluated only if <i>SECURITY LEVEL=LOW</i> . Connections to Kerberos-authenticated partners (before: chip card) are always cleared down if the connection to the terminal is lost.
AUTO	The partners are automatically taken over again as soon as the connection is reestablished. Connections to Kerberos-authenticated partners (before: chip card) are always cleared down if the connection to the terminal is lost. The effect of <i>HOLD=AUTO</i> depends on the <i>PRIORITY</i> setting. <i>HOLD=AUTO</i> is effective only for <i>SECURITY LEVEL=LOW</i> .
STD	The behavior when a connection is lost is defined by the <i>SET</i> command.

INPUT-LOGGING	Together with the <i>LOGGING</i> operand, <i>INPUT-LOGGING</i> controls the logging of messages entered at the terminal. <i>INPUT-LOGGING</i> is evaluated only if <i>LOGGING=YES</i> is set. With <i>LOGGING=NO</i> , no logging is executed. With <i>LOGGING=STD</i> , the <i>LOGGING</i> setting specified in the <i>SET</i> command applies.
REC	The messages are written to the logging file after being entered at the terminal.
SEND	The messages are written to the logging file before being sent to the partner.
BOTH	The messages are written to the logging file both when they are entered at the terminal and when they are sent to the partner.
STD	The value specified for <i>INPUT-LOGGING</i> in the <i>SET</i> command applies.
INPUT-SAVE	<p>Input made in line mode on terminals is saved in a buffer and can be output again with the specified K or F key. The output is in reverse order of the input; i.e. the LIFO principle applies. When the buffer reaches its maximum capacity (1000 characters), the oldest input is deleted.</p> <p>If no quantity is specified, the default value 1 applies. If a key is assigned but no output quantity is specified, the default value 1 also applies. If the K or F key is merely reassigned without the quantity being specified, the quantity that was previously set continues to be valid.</p> <p>If the default value <i>STD</i> is set or the functionality is switched off with <i>NO</i>, the quantity that was previously specified is reset to the default value of 1.</p> <p>LINE entries that are displayed in blanking mode are not buffered for security reasons. If the K key is pressed for screen dump after a request for a LINE entry that is displayed in blanking mode, the request is not processed. If the partner is changed following a LINE entry request and <i>SAVE=SCREEN</i> is set for the terminal, the saved screen cannot be restored if you return to the original partner. The screen is deleted.</p>
[K] <i>num1..14</i>	is the number of the K key in the range 1..14.
F <i>num1..24</i>	is the number of the F key in the range 1..24.
<i>qty1..16</i>	is the quantity of entered commands that are to be output.
NO	No K or F key is used to output the input in line mode.
STD	The K or F key defined by the <i>SET</i> command is used.

KPAC	defines a K key which can be used to force line mode message output when a format is displayed on the screen.
num	is the number of this K key in the range 1..14.
NO	No K key is used to switch to OMNIS command mode.
STD	The K Key defined by the <i>SET</i> command is used.
LINE25	controls the use of the 25th line of the screen for partner information (PAC, PID, partner/processor name, partner type, route, K key, user ID). <i>LINE25</i> is not evaluated for terminals of type 8160.
STD	The way in which the 25th line on the screen is used is defined globally (<i>SET</i>).
NO	The 25th line is not used for partner information.
YES	Partner information is written on the 25th line on the screen. OMNIS generates the 25th line on the screen only on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will no longer be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, e.g. from formatted to line mode and, for the 9763 terminal, as a result of a status query, e.g. on <i>LOGON</i> or <i>START-EDT</i> .
OMNIS	Partner information is written on the 25th line on the screen. OMNIS generates the 25th line on the screen on changing partners and whenever this line is modified by the set partner (at the time of the next output).
LISTENING	controls whether the dialog is output on another "listening" terminal; see also the "Functions and Commands" manual [1].
NO	Output is restricted to the current terminal.
hac	Dialog is sent both to the current terminal and to the terminal identified by <i>hac</i> .
LOGGING	controls the logging of messages to and from the partner(s) in the logging file. If <i>SECURITYLEVEL</i> ≠ <i>LOW</i> , all OMNIS commands and system messages are logged in the logging file, regardless of any other settings. The <i>LOGGING</i> entry is evaluated only for partners declared with <i>LOGGING=STD (DECLARE-PARTNER)</i> .
NO	Messages are not to be logged in the logging file.

YES	Messages are to be logged in the logging file.
STD	The logging of messages not specified by terminal.
	If the <i>PAR-OPT-SET</i> logging priority is set (see the "Functions and commands" manual [1]), the behavior is globally defined (see <i>SET</i> command). If the <i>SET-PAR-OPT</i> logging priority is set, the default value <i>NO</i> is valid.
MTAB	defines a message table with which the output of all the terminal's UCON partners is to be processed. <i>MTAB</i> is only effective for partners declared with <i>MTAB=STD (DECLARE-PARTNER)</i> . If a message table is assigned to the terminal, this table must be defined in the OMNIS startup file.
NO	Outputs from UCON partners are not to be processed by a message table.
name	Outputs from UCON partners are to be processed by the message table <i>name</i> .
STD	Processing of the outputs from UCON partners with a message table is defined in the <i>SET</i> command.
OPASS	defines an <i>OCCUPY</i> password. <i>OPASS</i> must be assigned to allow a connection takeover with the <i>OCCUPY</i> command.
OPNCON	defines the freedom of connection.
DECLARED	Connections may be set up from the terminal only to partners assigned to it by declaration (<i>DECLARE-PARTNER</i>). <i>OPNCON</i> commands for other partners are rejected.
FREE	Connections may also be set up from the terminal to undeclared partners.
OUTPUT-LOGGING	Together with the <i>LOGGING</i> operand, <i>OUTPUT-LOGGING</i> controls the logging of messages arriving from the partner. <i>OUTPUT-LOGGING</i> is evaluated only if <i>LOGGING=YES</i> is set. With <i>LOGGING=NO</i> , no logging is executed. With <i>LOGGING=STD</i> , the <i>LOGGING</i> setting specified in the <i>SET</i> command applies.
REC	The messages are written to the logging file after they have been received from the partner.
SEND	The messages are written to the logging file before being sent to the terminal.
BOTH	The messages are written to the logging file both when they are received from the partner and when they are sent to the terminal.

STD	The value specified for <i>OUTPUT-LOGGING</i> in the <i>SET</i> command applies.
PAC	defines output of the partner address code for line mode. The <i>PAC</i> entry is evaluated only for partners declared with <i>PAC=STD</i> (<i>DECLARE-PARTNER</i>). OMNIS messages are always output if <i>PAC=PREFIX</i> .
PREFIX	The partner address code is to be displayed in the same line as the partner's message (as a prefix).
LINE	The partner address code is to be displayed in a separate line (together with the partner's name and type) before the message itself.
NO	The partner address code is not to be displayed.
STD	Partner address code output is defined in the <i>SET</i> command.
REPLY	controls replies to UCON applications.
ALL	OMNIS passes all input through to UCON.
RESTRICTED	In the event of a reply to UCON, OMNIS checks that the associated question was output on the relevant terminal. If so, it passes the reply through to UCON. If not, it rejects the reply and issues message OMS0236.
STD	The value specified in the <i>SET</i> command applies.
SAVE	specifies whether screen contents and/or P keys are saved on quitting the application.
STD	The setting defined in the <i>SET</i> command applies.
ALL	Both the screen contents and also the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
PKEY	The P keys are saved individually for each application.
SCREEN	The screen contents are saved and output on the terminal when the application is selected again. (In line mode, the screen is cleared following the first input.)
SNMP	This operand enables the terminal connection to be monitored via SNMP.
YES	The terminal connection is monitored via SNMP and OMNIS.
NO	The terminal connection is not monitored via SNMP.

TERMTYP	specifies the type of the declared terminal. <i>TERMTYP</i> should match the actual terminal type. In principle, OMNIS can obtain the terminal type dynamically. Note, however, that <i>TERMTYP</i> needs to be specified if the terminal is defined with <i>CONNECT=LOGON</i> and if a partner is assigned to the terminal using <i>CONNECT=START</i> . OMNIS does not know the actual terminal type on setting up the connection and thus assumes the specified <i>TERMTYP</i> . If this entry does not subsequently match the actual terminal type, OMNIS issues the message OMS0084, since problems could otherwise occur for the partner with the incorrectly set type.
8160-9763	Terminal hardware
APPL	DCAM
SKP	SKP
TRANSPORT-PROTOCOL	specifies the type of the transport protocol. The operand is accepted only with <i>TERMTYP=APPL</i> .
NEA	The NEA transport protocol is used.
ISO	The ISO transport protocol is used.
USER-PROTOCOL	specifies the protocol defined for use between OMNIS and programs as terminals. This operand is evaluated only if <i>TERMTYP=APPL</i> .
OMNIS	The OMNIS-specific protocol for programs as terminals is used.
NO	No protocol is used, i.e. only the net message is sent. OMNIS always uses DUE as the key code. The application must determine the output type (<i>LINE/FORM/PHYS</i>) from the message header.
VTSUCB	The OMNIS-APPL protocol is supported. This allows applications as terminals access to the VTSUCB, which is supported as of Version 6.2.

See [page 207](#) for an example.

DECLARE-USER - Declare a user

The *DECLARE-USER* statement is used to declare a user.

Operation	Operands
D[ECLARE]-U[SER]	N[AME] = user_entry [,C[OMMENT]='max-string70'] [,E[XCLUSIVE]-P[ARTNER]= $\left. \begin{matrix} \text{N[O]} \\ \text{Y[ES]} \\ \text{S[TD]} \end{matrix} \right\}]$ [,H[ARDCOPY]-D[EVICE] = devname] [,H[ARDCOPY]-F[ORM] = $\left. \begin{matrix} \text{form-name} \\ \text{S[TD]} \end{matrix} \right\}]$ [,I[NFO] = $\left. \begin{matrix} \text{'c'cinfo'} \\ \text{'x'xinfo'} \end{matrix} \right\}]$ [,L[ENGTH] = 0..120] [L[OGOUT]-H[OLD]-A[VAIABILITY]= $\left. \begin{matrix} \text{Y[ES]} \\ \text{N[O]} \end{matrix} \right\}]$ [,M[ENU] = menu_entry] [,O[MNIS]-C[OMMAND]-M[ODE] = $\left. \begin{matrix} \text{N[O]} \\ \text{Y[ES]} \end{matrix} \right\}]$ [,PA[SSWORD] = $\left. \begin{matrix} \text{'password'} \\ \text{*INITIAL} \\ \text{*UNCHANGED} \\ \text{'*NONE'} \\ \text{*REQUIRED} \end{matrix} \right\}]$ [,PO[SITION] = 1..120] [PR[INCIPAL]= $\left. \begin{matrix} \text{*NO} \\ \text{*ALL} \\ \text{'<user_principal 1..120>'} \end{matrix} \right\}]$ [,T[IMEOUT] = 0..1440] [,U[SER]-N[UMBER] = 0..9999]

NAME	specifies the name of the user. <i>user_entry</i> must not exceed 8 character in length. The user must use this name to identify himself to OMNIS-MENU.
COMMENT	is a comment on the menu; up to 70 characters in length.
EXCLUSIVE-PARTNER	controls the message flow.
NO	All messages of all the partners are displayed at the terminal.
STD	The OMNIS setting SET applies.
YES	You are conducting a dialog with an application. This application's messages are displayed. All messages of all other applications are buffered and not displayed at the terminal. All messages requesting input are announced by message OMS0215 in the 25th line on the screen. Only the first output message is announced: this is done with message OMS0214 in the 25th line on the screen.
HARDCOPY-DEVICE	defines a user-specific printer name (RSO) for the hardcopy function. The name specified here has priority over the printer name entered in the menu.
HARDCOPY-FORM	This operand sets the form for the printer. By specifying <i>STD</i> or <i>form-name</i> (max. 6 characters), you can define the format of the print form to be used on the printer defined with <i>HARDCOPY-DEVICE</i> . If the form name consists only of blanks, the <i>PRINT</i> command is issued without a form name. If a switchover is made from the specified printer to a directly connected printer with <i>HCOPY=hac</i> , or the print output is suppressed with <i>HCOPY=STD</i> , the form name is automatically set to the standard form <i>STD</i> .
INFO	The ID card information is specified here. During the authorization check OMNIS-MENU interprets only as much of this information as is specified in the <i>LENGTH</i> entry. Thus the maximum length of the input field is 120 characters.
c'cinfo'	The ID card information is entered as a character string. Example: <i>INFO=c'definitionfollows'</i>
x'xinfo'	The ID card information is entered in hexadecimal code. Example: <i>INFO=x'AD24B7'</i>

LENGTH	defines the length of the relevant ID card information. The information may not exceed 120 characters in length. <i>LENGTH=0</i> (default) means that no ID card is required for identification purposes.
LOGOUT-HOLD-AVAILABILITY	Controls the availability of the K2 option for the user when logging out from the OMNIS-MENU.
YES	The K2 option is available (default value).
NO	The K2 option is not available.
MENU	specifies the name of the menu assigned to the user. <i>menu_entry</i> may not exceed 8 characters in length. It serves as a key to the <i>omm.menutab</i> file.
OMNIS-COMMAND-MODE	specifies whether the user can enter the OMNIS command mode from the OMNIS-MENU <i>Application Menu</i> by means of the <i>*OMNIS</i> command or by pressing the appropriate K or P key (<i>OMNIS-COMMAND-MODE=YES</i>). Default= <i>NO</i> .
PASSWORD	defines the user password.
password	<i>password</i> must not exceed 8 EBCDIC characters in length. Since only printable characters can be entered in <i>PASSWORD</i> input fields of the login mask, only such characters are meaningful for generation purposes as well. You can only use the character ',' if <i>PASSWORD</i> is not the last operand. A hyphen included in the eight characters is interpreted as a continuation character. If no assignment is made, i.e. the <i>PASSWORD</i> operand is missing, the password is filled with blanks.
*INITIAL	If <i>*INITIAL</i> is specified, the user is forced to define a new password the next time he logs on. This saves the administrator from having to define a password for each user, which the user may not then change. The password defined by the user must meet the conditions defined with <i>PROTECT-PW</i> (see page 136).
*UNCHANGED	If <i>*UNCHANGED</i> is specified, any password which is already stored in the assigned user table is retained. If no password yet exists, the operand is overwritten with blanks.
* <i>NONE</i>	If <i>*NONE</i> is specified, no password is assigned to the user. subsequently no password can be assigned to this user. This user is accessed either without specifying a password or by using the password <i>*NONE</i> .

***REQUIRED** Has the same effect as *PASSWORD=*UNCHANGED*, if the user already exists in the user table.
 Has the same effect as *PASSWORD=*INITIAL*, if the user does not yet exist in the user table.

POSITION Position on the ID card at which the relevant information starts.
POSITION may not exceed 120. Default=1.

PRINCIPAL controls the protection provided by Kerberos authentication.
 You will find basic information on the Kerberos support via OMNIS-MENU in the "Functions and Commands" manual [1].

***NO** If ***NO** is entered, no Kerberos authentication is provided for this user.
 The client is not requested to present a Kerberos ticket.
 Access protection is not negotiated by means of a Kerberos ticket. This is the default value.

***ALL** If ***ALL** is entered, no Kerberos authentication is provided for this user.
 However the client is asked to present a Kerberos ticket.
 If the client does not support Kerberos authentication, access to OMNIS-MENU will be denied.

'<user_principal 1..120>'

If a principal is indicated, Kerberos authentication is executed for this user. The client has access to OMNIS-MENU if he has a Kerberos ticket and the supplied principal complies with the principal configured for this user. If the client does not support Kerberos authentication, access to OMNIS-MENU will be denied.

The name of the principal of a user comprises three components: User, instance and realm.

Typical format: Benutzer/Instanz@REALM,
 e.g. MCHAUser@TS.FUJITSU.COM, where:

User is the user name. In Windows this is the Domain ID of the user

Instance is generally NIL and is not indicated.
 (also / is not indicated)

REALM Name of the Kerberos-domain, presented in upper case.

At DECLARE-USER the user name and the REALM have to be stated for the principal. OMNIS-MENU converts the user entry into upper case letters. Upon checking of the requested ticket the supplied principal is converted into upper case letters as well and thus can be checked for compliance.

TIMEOUT	Specifies the inactivity time (in minutes) before the connection to a terminal is cleared down. If <i>TIMEOUT=0</i> (=default), there is no time limit. This operand takes priority over the <i>TIMEOUT</i> operand of the <i>DECLARE-MENU</i> statement. A user who requires particular protection can therefore be assigned a shorter time than actually provided for in a menu.
USER-NUMBER	Freely selectable four-digit number. Together with the application number it serves to define the start sequence. The default= <i>0000</i> means "no user number".

9 OMNISMD: Generation program

OMNISMD (**OMNIS-MENU** Definitions) is a program which enables administrators to generate the data for OMNIS-MENU.

It makes it easy to create and modify the files used by OMNIS-MENU for various functions. The names of these files can be reassigned by means of link names (*SET-FILE-LINK* command).

The administrator can define an individual working menu for each user. Each menu can contain up to 900 different entries of any type (*APPL*, *ADMI* or *TEXT*).

9.1 Starting OMNISMD

You can call OMNISMD not only with the *START-PROGRAM* command, but also with *START-OMNISMD*. However, in order to do this, the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* must be enabled.

Example

OMNISMD can be started with the following procedure, for example:

```
/BEGIN-PROCEDURE PARAMETERS=YES(PROC-PARAM=( -
/      &VEROMS=085, -
/      &VEROMM=035, -
/      &LANG=D, -
/      &UID=00, -
/      &OMSUID=OMNIS, -
/      &OMMUID=OMNIS, -
/      &PRE=MAN), -
/      ESC-CHAR='&'),LOG=A
/ASSIGN-SYSLST TO-FILE=&VRS..OMNISMD.LIST
/SET-FILE-LINK FILE-NAM=$&UID..&PRE..&VEROMM..GENTAB, LINK-NAME=OMMGEN
/SET-FILE-LINK FILE-NAM=$&UID..&PRE..&VEROMM..USERTAB, LINK-NAME=OMMUSER
/SET-FILE-LINK FILE-NAM=$&UID..&PRE..&VEROMM..APPLTAB, LINK-NAME=OMMAPPL
/CREATE-FILE FILE-NAME=$&UID..&PRE..&VEROMM..MENUTAB,-
/      SUPPORT=*PUBLIC-DISK(-
/      SPACE=*RELATIVE(PRIMARY-ALLOCATION=48,-
/      SECONDARY-ALLOCATION=16))
```

```

/SET-JOB-STEP
/SET-FILE-LINK FILE-NAME=$&UID..&PRE..&VEROMM..MENUTAB, -
/          LINK-NAME=OMMMENU, -
/          BUFFER-LENGTH=STD(SIZE=16)
/MODIFY-MSG-FILE-ASSIGNMENT ADD=$&OMMUID..SYSMSA.OMNIS-MENU.&VEROMM
/MODIFY-MSG-ATTRIBUTES SEARCH=*ALL,TASK-LANGUAGE='&LANG'
/SET-FILE-LINK FILE-NAME=$&OMMUID..SYSFHS.OMNIS-MENU.&VEROMM..&LANG, -
/          LINK-NAME=MAPLIB
/SET-TASKLIB LIBRARY=$TSOS.MFHSROUT
/SET-JOB-STEP
/START-OMNISMD
/SET-JOB-STEP
/MODIFY-FILE-ATTRIBUTES FILE-NAME=$&UID..&PRE..&VEROMM..MENUTAB,-
/          SUPPORT=PUBLIC-DISK(SPACE=RELEASE(-
/          NUMBER-OF-PAGES=ALL-RELEASABLE))
/MODIFY-MSG-FILE-ASSIGNMENT REMOVE-FILE=$&OMMUID..SYSMSA.OMNIS-MENU.&VEROMM
/REMOVE-FILE-LINK LINK-NAME=OMMGEN
/REMOVE-FILE-LINK LINK-NAME=OMMUSER
/REMOVE-FILE-LINK LINK-NAME=OMMMENU
/REMOVE-FILE-LINK LINK-NAME=OMMAPPL
/ASSIGN-SYSLST TO-FILE=*PRIMARY
/END-PROCEDURE

```

In the *list* file, the tables are to be prepared for printing.

filename1..filename4 are the fully qualified file names assigned with *SET-FILE-LINK*.

format-library is the name of the OMNIS-MENU format library. If it is not assigned with *SET-FILE-LINK*, OMNIS-MENU uses the file *SYSFHS.OMNIS-MENU.035.E*.

library-name contains the FHS formatting modules. If they are in the *\$TSOS.TASKLIB* system library, they do not need to be assigned with *SET-TASKLIB*.

If files are not assigned with *SET-FILE-LINK* commands, OMNISMD or OMNISBD creates files with the default file names *omm.gentab*, *omm.usertab*, *omm.menutab* and *omm.appltab* under the current user account.

The block size of *omm.menutab* must be a multiple of 16.

If an error occurs while OMNISMD is being initialized, this is acknowledged with the message OMM0601. The procedure is resumed at the next */SET-JOB-STEP*. The cause of the error is indicated in the error message.

9.2 Working with OMNISMD

OMNISMD offers the following functions:

- creation, modification and deletion of user entries in the user table
- creation, modification and deletion of menu entries in the menu table
- creation, modification and deletion of application entries in the application table
- creation and modification of the parameters
- output of file contents on SYSLSST
- editing a start-sequence with EDT
- call of the HELP functions
- termination of OMNISMD

When OMNISMD is started, the *Function Menu* appears. The administrator can then switch to the various menus by selecting the required functions.

OMNISMD can be terminated from any mask with the F3 key. With function 9 or the K1 key, you arrive back at the *Function Menu*. If you then choose function 9 again, OMNISMD is terminated.

By pressing the F1 key, you can call up help with the current mask.

The diagram below shows how the various functions are interrelated.

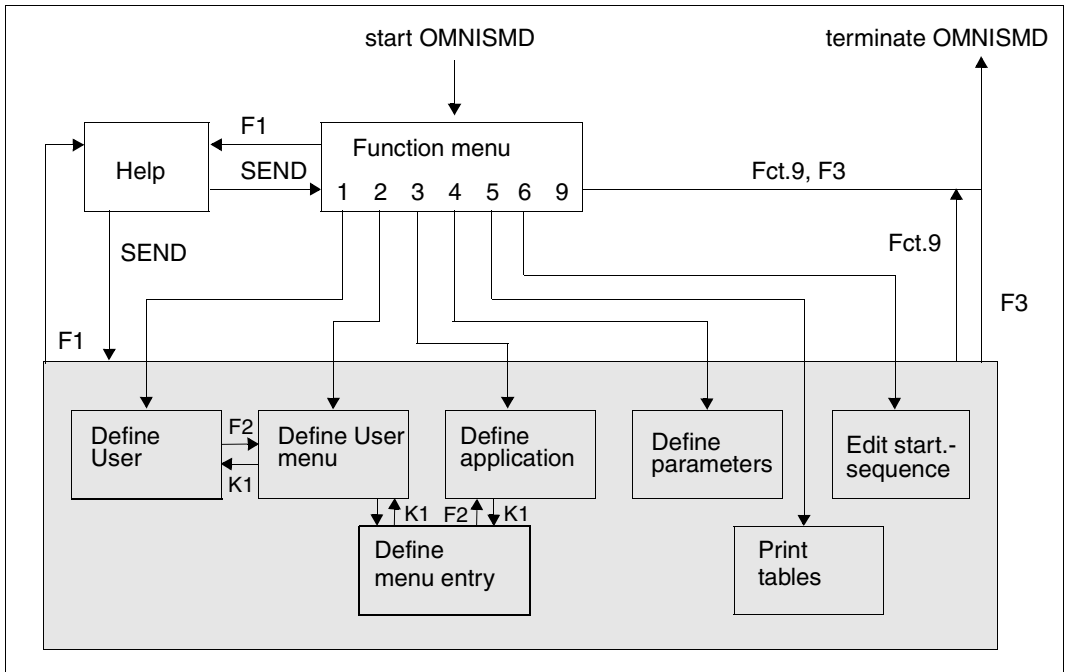


Figure 15: OMNISMD functions

Order of the entries

A new user is entered in the *Define User* mask.

With the F2 key, you can switch to the *Define User Menu* mask. The menu entries appear as they will later be seen by the user.

If there is already a user menu, it is shown by OMNISMD. Now, you can edit the menu entries. Mark the menu entries with an *X* in the first column.

The *Define Menu Entry* mask is then displayed. In this mask, you can define the type of menu entry in more detail (type *APPL*, *ADMI* or *TEXT*).

With menu entries of the type *APPL*, you can switch directly to the *Define Application* mask with the F2 key. If the application does not yet exist, you must enter the PAC and the domain (local or global) and then switch to the *Define Application* mask with the F2 key.

When you return to the *Define Menu Entry* mask, OMNISMD checks the entered data and writes the *APPL*-type menu entry into the menu.

From any mask, you can return to the previous mask using K1.

Workflows in OMNISMD

Starting with mask *Function Menu* (OMMGMAI) the following workflows are foreseen in OMNISMD:

Mask	Next mask	Object	Action	Next mask
OMMGMAI (Fct. 1)	OMMGUSR	user	F2	OMMGMEN
OMMGMAI (Fct. 2)	OMMGMEN	menu	mark, SEND	OMMGMEE
	OMMGMEE	menu-entry	F2	OMMGAP1/2
OMMGMAI (Fct. 3)	OMMGAP1/2	application	F4	OMMGTXT
OMMGMAI (Fct. 6)	OMMGTXT	start-sequence	SEND	EDT

Concurrent access to the start-sequence is prevented by a lock-mechanism. 3 cases have to be distinguished:

1. More than one OMNISMD wants to edit a certain start-sequence.
Once the first OMNISMD edits the start-sequence, all successors get the error message `OMM0666`.
2. OMNIS just processes the start-sequence.
While OMNIS processes the start-sequence, it is locked against modifying. The start-sequence can be edited in EDT, but the write back waits up to 10 seconds for the lock to be released. If the lock isn't released and timeout occurs EDT gets back the control and displays the error message `OMM0666`. The delete of a start-sequence also waits up to 10 seconds for the lock to be released, before the attempt is canceled with the error message `OMM0666`.
3. OMNISMD writes the start-sequence into the text file.
In this more or less short time period the start-sequence is locked against OMNIS. If there is indeed a collision, OMNIS skips the start-sequence processing and displays the error message `OMS0276`.

Problems during writing back the start-sequence.

- The length of the records must not exceed 255 characters.
If some records are longer, OMNISMD asks during write back how to deal with that. The user has the following choices:
 - add: the too long records are truncated. OMNISMD confirms with warning message `OMM0661`
 - discard: the start-sequence is not written back and thrown away. OMNISMD confirms with the warning message `OMM0660`
 - return to EDT.

- The number of records depends on the key length of the start-sequence number. If the start-sequence consist of too many records, OMNISMD asks during write back with message OMS0665 how to deal with that.
The user has the following choices:
 - add: the maximum number of records are added, the rest is thrown away. OMNISMD confirms with the message OMM0662
 - discard: the start-sequence is not written back and thrown away. OMNISMD confirms with the message OMM0660
 - return to EDT
- The EDT is not exit with RETURN but with HALT or K1. OMNISMD asks during write back with message OMS0663 how to deal with that. The user has the following choices:
 - add: the start-sequence is written back. OMNISMD confirms with the message OMM0611
 - discard: the start-sequence is not written back and thrown away. OMNISMD confirms with the warning message OMM0660
 - return to EDT

9.3 Masks

The various BS2000 masks of OMNISMD have the following features in common.

Screen layout

The name of the format being displayed is shown at the top left of the screen after *Function:*. At the center of the line below, there is a brief description of the function of the mask.

In the area for entering data, there are either a prompt for the next function and the necessary information for the various functions, or the information and write fields for entering data in the specific table.

The third section of the screen contains a reference to the function keys for terminating OMNISMD and for help. This is also the section in which the active function is selected (except in the *Function Menu* mask). It is also possible in the various masks to load the next data record to be processed immediately after executing the previously entered action (*Next entry* field; the data records are sorted alphabetically).

Below the third screen section, there are two lines for displaying OMNISMD messages.

Data input

Only the *Text in menu* field in the *Define Menu Entry* mask is case-sensitive.

You can move the cursor from one input field to another with the tab keys.

All the fields required for the specific data record are filled in, and the values are then submitted with the SEND key. If one of your entries is incorrect, the input field remains clear (the other fields go darker) until you correct the values and submit them with the SEND key.

Help function

With the F1 key, you can call up help with the current mask.

With the SEND key, you can then return to the current mask.

Function Menu

When you start OMNISMD, the *Function Menu* mask appears:

```

Function: OMMGMAI          O M N I S - M E N U          11:45:25/2015-03-05
                        Function Menu
-----
      Please select a function   1

      Manage user entries       : 1
      Manage menu entries       : 2
      Manage application entries : 3

      Manage global parameters  : 4

      Print generation data     : 5

      Manage start sequences    : 6

      Terminate program         : 9

-----
      F1=Help F3=Quit SEND=Send
  
```

Figure 16: Function menu

Enter the appropriate number for the desired function next to the prompt *Please select a function* and submit it with the SEND key. Default=1.

Meaning of the fields

- 1 Selects the **Define User** mask (see [page 181](#))
- 2 Selects the **Define User Menu** mask (see [page 186](#))
- 3 Selects the **Define Application** mask (see [page 192](#))
- 4 Selects the **Define Parameters** mask (see [page 200](#))
- 5 Selects the **Print Tables** mask (see [page 203](#))
- 6 Selects the **Edit start sequence** mask (see [page 205](#))
- 9 **Ends the generation** and closes the open files

Define User

You can call up the *Define User* mask by entering *1* in the *Function Menu* mask (see [page 180](#)).

```

Function: OMMGUSR          O M N I S - M E N U          12:01:19/2015-03-05
                        Define User
-----
Read/write/delete entry (R/W/D)      R

User name, Password, user number     TESTUSER / ZZZZZZZZ / 0000
Comm. Authentifizierung mit Kerberos
User menu name                        TESTMENU
Hardcopy device name / Form          / STD
OMNIS command mode (Y/N)             Y
Exclusive partner (Y/N/' '=STD)      N
LOGOUT HOLD in logout mask (Y/N)     Y
TIMEOUT (max. 1440 min)              0000
Principal or ID card reader:
  Position of principal or ID card    001
  Length of principal or ID card      026 Length = '0': no specification
  Principal (P) / ID-code (X/C) P     TESTUSER@G02.FUJITSU.LOCAL

Next entry [ ]                      Next function 1 (9: Function Menu)
-----
F1=Help F2=Define Menu F3=Quit K1=Cancel

```

Figure 17: Define User

All the data needed to identify the user is entered in this mask. This mask also specifies the name of the menu assigned to the user.

Meaning of the fields

Read/write/delete entry (R/W/D)

Here you specify whether the entered data is to be read (*R*), written (*W*) or deleted (*D*).
Default=*R*.

An entry in the user table can be written only if there is a corresponding menu entry in the menu table.

User name, Password, user number

First, the name of the user for the entry in the *omm.usertab* file is entered here. The user must use this name to log on at OMNIS-MENU (max. 8 characters in length).

If *R* is the value entered in the *Read/write/delete entry* field, you can carry out the following functions with the following input:

- * outputs a mask with a list of all the users.
You then see the diagram on [page 185](#).
You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + loads the next data record.
- loads the previous data record.
- ++ loads the last data record.
- loads the first data record.

The data records are sorted by name in ascending alphabetical order.

After the user name, you enter the password (max. 8 characters in length). The password is overwritten with *Zs* for reading purposes.

You must fill in the *Password* field again if you want to change the password.

The password assigned by the administrator must comply with the globally generated password complexity rules.

The user number is a freely selectable four-digit number, used together with the application number in start sequence definitions. The default of *0000* means that no user number is defined.

It is also possible to deliberately assign no password to a user (see [page 167](#)).

Comm.

Here you enter a comment about the user with a maximum of 70 characters.

User menu name

The name of the menu offered to the user is entered here (max. 8 characters in length). OMNISMD checks whether there is already a corresponding entry in the menu table. If you do not specify a name here, the user gets an empty menu after logging on and can work only in OMNIS command mode (and this only with appropriate authorization).

Hardcopy device name

Here you can specify a user-specific printer name for the hardcopy function.

If you do specify a name here, it takes priority over the printer name in the menu.

OMNIS command mode (Y/N)

If you set *Y*, the user can enter OMNIS command mode by entering **O[MNIS]* in the *Application Menu* mask or by pressing the corresponding K/P key.

Default=*N*.

Exclusive partner (Y/N/'=STD)

With this field, you can control the message flow.

If *Y* is specified, only the messages of the application with which you are in a dialog are output. All messages from other applications are buffered and not displayed on the terminal.

The first output message is announced on the 25th line of the screen by the OMNIS message OMS0214.

Messages requiring input are announced with the message OMS0215.

If *N* is specified, all messages of all applications are output to the terminal (no message flow control).

If no specification is made for *Exclusive partner*, the OMNIS *SET* setting is used.

LOGOUT HOLD in logout mask (Y/N)

Controls the availability of the K2 option for the user when logging out from the OMNIS-MENU.

TIMEOUT (max. 1440 minutes)

Specifies the inactivity time (in minutes) before the connection to a terminal is cleared down. If *TIMEOUT=0* (=default), there is no time limit.

Principal or ID card reader

See [page 170](#) for details on the specification of the principal.

Position of principal or ID card

Position of the principal or position on the ID card where the relevant information starts (max. 120 characters in length). Default=*001*.

Length of principal or ID card

Length of the principal or of the relevant ID card information. The maximum length of the information is 120 characters. If the value is left as *000*, this means that no ID card is needed for identification. If the ID card information is input in hexadecimal code, the number of characters entered must be even. Default=*000*.

Principal (P) / ID-code (X/C)

If *P* is entered, Kerberos-authentication is performed for this user. The principal is input next to this.

Code (X/C) specifies whether the ID card information is input as a character string (*C*) or in hexadecimal code (*X*). The ID card information is input next to this.

OMNIS-MENU evaluates this information only in relation to what is specified for *Length of ID card information*. The maximum length of the input field is therefore 120 characters.

Default=*C*.

Next entry

The user name of the next data record awaiting processing can be entered here (max. 8 characters in length). If the previous action was "Write" (*W*) or "Delete" (*D*), this data record is loaded as soon as the specified action has been executed.

When you input:

- * a mask is output with a list of all the users.
You then see the diagram on [page 185](#).
You can page in the list by entering +,- or ++,--. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + - the next data record is loaded.
- - the previous data record is loaded.
- ++ - the last data record is loaded.
- - the first data record is loaded.

The data records are sorted by name in ascending alphabetical order.

Next function

By inputting the appropriate number, you can select another mask. With function 9 you return to the *Function Menu* mask. Actions entered in the *Read/write/delete entry* field are no longer executed. Default=1.

List of all defined users

From the *Define User* mask you can arrive at the following mask (see [page 182](#)) in two ways:

- by entering * in the *User name* field, if R is entered as the value in the *Read/write/delete entry* field
- by entering * in the *Next entry* field

```
Function: OMMGLST          O M N I S - M E N U          11:53:46/2002-01-22
                          List of USERTAB
-----
X IDEFIX
█ IDEFIXA
 IDEFIXAP
 IDEFIXC
 IDEFIXFO
 IDEFIXII
 IDEFIXIN
 IDEFIXNN
 IDEFIXNO
 IDEFIXP
 IDEFIXTI
 IDEFIXU
 IDEFIXZ
 IDEFIX00
 IDEFIX1
 IDEFIX2
Command :  (++,--,+,-)
-----
K1=Cancel F3=Quit SEND=List navigation/with 'x' get marked record
```

Figure 18: USERTAB list

Define User Menu

You can call up the *Define User Menu* mask in two ways:

- by entering 2 in the *Function Menu* mask (see [page 180](#))
- by pressing the F2 key in the *Define User* mask (see [page 181](#))

```

Function: OMMGMEN          O M N I S - M E N U          11:55:41/2002-01-22
                        Define User Menu
-----
Read/write/delete entry      (R/W/D)                      Page 01/01
Name of menu and user number IDEFIX / 0000 Timeout (max.1440) 0000
Spooler (Device, K and P key) DRPC9200 K10 P10 Form RS12ST
Comm. IDEFIX
PAC Text                    K/F P          PAC Text                    K/F P
OE                          K05 05          OME OMSME - 7/133          K06 00
OW                          K07 00          OMW OMSMW - 7/133          K08 00
OE28                        K00 00          VM4 OMSW - 14/214 - VM4   K00 00
OW28                        K00 00          VM5 OMSW - 29/214 - VM5   K00 00
SKV SKV - 7/133            K00 00          12 OMNIS111 - 4/16        K00 00
DOC DOC - 5/16             K00 00          PULS PULSV3 - 5/16        K00 00
OM11 OMLUTM11 - 7/133     FD6 06          00 00 - 7/133            K00 00
RZO RZOMNIS - 7/133       K00 00          DSIM DSSIM - 7/133        K00 00
MAIL                         K00 00          14 OMSEKT2 - 7/133        K00 00
ADM1 *UPDATE               K00 00          ADM2 *UNLOCK               K00 00
Forward/Backward            (+,-,++,--,+,n,-n)
Next entry                  Next function 2 (9: Function Menu)
-----
F1=Help F3=Quit K1=Return/Cancel

```

Figure 19: Define User Menu

This mask is for entering the name of the menu assigned to the user. It is also for entering all the applications which are to be available to the user in this menu. These names (PACs) must be unique.

You can change the order of the menu entries as you wish, even moving them from page to page. Mark the application(s) you want to move with *M* in the marking column. Then enter *B* (before) or *A* (after) in the marking column of the menu entry before or after which you want place the marked application(s).

Meaning of the fields

Read/write/delete entry (R/W/D/_)

Specify here whether the input data is to be read (*R*), written (*W*) or deleted (*D*). Specify *_* if you want to rearrange, page or mark entries. Default=*R*.

An entry is made in the menu table only if all the assigned applications are already in the application table.

Page n/N

You are currently viewing page *n* of a total of *N* pages in this menu.

Name of menu and user number

Name of menu specifies the name of the menu defined in the *omm.menutab* file (max. 8 characters in length).

If *R* is the value entered in the *Read/write/delete entry* field, you can carry out the following functions with the following input:

- * outputs a mask with a list of all the user menus.
 You then see a list comparable to the list of all defined users, which is shown in the diagram on [page 185](#).
 You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + - loads the next data record.
- - loads the previous data record.
- ++ - loads the last data record.
- - loads the first data record.

The user number is a freely selectable four-digit number, used together with the application number in start sequence definitions. The default of *0000* means that no start sequence is defined.

Spooler (Device, K and P key)

Here you can enter the name of the printer for hardcopy output. This name is used only if no printer name is assigned to the user.

Specify the K key which is to initiate a printout of the current screen.

Specify the P key which is to initiate a printout of the current screen.

Comm.

Here you enter a comment about the menu in a maximum of 70 characters.

Operations in the marking column (left of the PAC):

By entering *x* (change), you can output the *Define Menu Entry* mask, in which you can then define the type of menu entry more precisely or modify it.

By entering *d* (delete), you can delete the menu entry from the table. In other words, you delete the reference to the actual menu entry. The menu entry in the menu table is replaced with a blank line.

By entering *m* (move) and *b* (before) or *a* (after), you can change the order of menu entries, even moving them from screen to screen.

PAC

The PAC of the application is entered here. The PAC must be globally unique for global applications, and unique within the menu for local applications. The maximum length of *pac* is four (alphanumeric) characters.

Text

Text describes the menu entry (max. 20 characters).

K/F

K or F key to be used for switching to this application. The default of *K00* means that no K or F key is assigned.

P

P key to which the specified K/F key is loaded automatically. The default of *00* means that no P key is assigned.

K and F keys can be assigned without being associated to P keys. P keys can be assigned only in conjunction with K or F keys.

Paging commands (+, -, ++, --, +n, -n)

By entering the appropriate commands, you can page in the user menu. ++ brings you to the last defined page of the user menu.

Next entry

The menu name of the next data record awaiting processing can be entered here (max. 8 characters in length). If the previous action was "Write" (*W*) or "Delete" (*D*), this data record is loaded as soon as the specified action has been executed.

When you input:

- * outputs a mask with a list of all the user menus.
You then see a list comparable to the list of all defined users, which is shown in the diagram on [page 185](#).
You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + - loads the next data record.
- - loads the previous data record.

- ++ - loads the last data record.
- - loads the first data record.

Next function

By inputting the appropriate number, you can select another mask. With function 9 you return to the *Function Menu* mask. Actions entered in the *Read/write/delete entry* field are no longer executed. Default=2.

Define Menu Entry

You can call up the *Define Menu Entry* mask by marking an entry with *X* in the *Define User Menu* mask (see [page 186](#)).

```

Function: OMIMGEE          O M N I S - M E N U          11:56:55/2002-01-22
                          Define Menu Entry
-----
Name of Menu      : IDEFIX
Menu entry type  : APPL

PAC of menu entry   OE          (only of type APPL or ADMI)
Text in menu
Administrator function  (*UNLOCK,*UPDATE)
Domain              L          (G=global,L=local)

Here you define a menu entry of a menu. A menu entry is of type TEXT,
APPLICATION or ADMINISTRATION. Only Applications or Administrator functions
have a PAC. For Applications the field 'Administrator function' must be
Blank. An application is either LOCAL (only valid for menu IDEFIX ) or
GLOBAL (used by other menus).

-----
F2=Define Application F3=Quit K1=Return/Cancel DUE=Write entry

```

Figure 20: Define Menu Entry

In the *Define Menu Entry* mask you can define either an application or an OMNIS-MENU administrator function.

With F2 you arrive at the first of the two *Define Application* masks.

With K1 you can exit the mask.

When you send the entry with SEND, a plausibility check is run, and the entry is accepted if this check is successful.

Meaning of the fields

Name of Menu

Here you see the name of the current user menu.

Menu entry type

Here you see the type of the menu entry:

- *APPL* for an application
- *ADMI* for an OMNIS-MENU administrator function
- *TEXT* for a text entry

PAC of menu entry

The PAC of the application is entered here. The PAC must be globally unique for global applications, and unique within the menu for local applications. The maximum length of *PAC* is four (alphanumeric) characters.

Text in menu

Here you can modify the text of the menu entry again.

Administrator function

Only for entries of the type *ADMI*. You specify which OMNIS-MENU administrator function (**UNLOCK*, **UPDATE*) you want to link to *PAC*.

Domain

Only for entries of the type *APPL*. You can specify whether the application is to be locally (*L*) or globally (*G*) valid.

Define Application

You can call up the *Define Application* mask (*OMMGAP1* function) in two ways:

- by entering 3 in the *Function Menu* mask (see [page 180](#))
- by pressing the F2 key in the *Define Menu Entry* mask (see [page 190](#))

Function: OMMGAP1	OMNIS - MENU	12:03:45/2015-03-05
Define Application		

Read/write/delete entry (R/W/D)		R
Menuname and Partner address code (PAC)	* [] /	PRE1
Comm.	COMMENT: TYPE=*PRE,PRONAME=*PREDIALOG,PTNNAME=*PREDIALOG	
User and application number	0000 /	00
Network address of application (PTN/PRO)	*PREDIAL /	*PREDIAL
Application type (TIAM,UTM,DCAM,etc.)	*PRE	
Route (I=INDIRECT, D=DIRECT, M=MUX)	D	
Suppress LINE message (Y/N)	N	
Name of substitute application (APP)	[]	
Name of P key set	[]	
Connection password (CPASS)	[]	
Connection message (CMSG)	[]	
Next entry [] / []	Next function 3 (9: Function Menu)	

F1=Help F2=Second page F3=Quit F4=start sequence K1=Return/Cancel		

Figure 21: Define Application

To go to the next mask, press the F2 key:

Function: OMMGAP2	OMNIS - MENU	12:04:36/2015-03-05
Define Application		
Read/write/delete entry (R/W/D)		R
Application switch (S=SCREEN,P=PKEY,A=ALL,N=NO,' '=STD)		
Automatic disconnection (Y/N)		N
SNMP (Y/N)		N
User number priority (U/A)		U
Restart message (X/C)		C
Length of the restart message		00
Contents	[REDACTED]	
UTM: SIGNON data (B=BOTH,U=USER,N=NONE,S=SUPPRESS)		B
TIAM: LOGOFF BUT not allowed (Y/N)		N
DCAM: Lower case letters (Y/N)		N
Code (S=SYSCODE,B=BINARY)		S
Protocol (O=OMNIS,D=DSSIM)		O
SKP: Authorization and password		[REDACTED] / [REDACTED]
Next entry	[REDACTED] / [REDACTED]	Next function 3 (9: Function Menu)
F1=Help F2=First page F3=Quit F4=start sequence K1=Return/Cancel		

Figure 22: Define Application

To return to the initial mask, press the F2 key.

An application entry contains all the information for issuing an OMNIS *OPNCON* command.



The general OMNIS conventions apply to operands *PAC*, *TYP*, *ROUTE*, *PTN*, *PRO*, *APP*, *CMSG*, *CPASS*, *LCASE* and *CODE* (see the "Functions and Commands" manual [1]).

Depending on the partner type, some specifications are mandatory; see also [page 199](#). OMNISMD checks the specifications for completeness. An entry can be made in the application table only if all the required parameters are specified correctly.

Meaning of the fields

Read/write/delete entry (R/W/D)

Here you specify whether the entered data is to be read (*R*), written (*W*) or deleted (*D*).
Default=*R*.

Menu name and Partner address code (PAC)

Here you specify the menu name of the application; for local applications this is the name of the user menu, and for global applications it is '*'.

Global applications are not written to a menu with the *W* (Write) action.
For local applications a menu entry is automatically appended to the end of the relevant menu, if it does not already exist. The first 20 characters of the comment field are copied into the *Text in menu* field of the menu entry.

If *R* is the value entered in the *Read/write/delete entry* field, you can carry out the following functions with the following input:

* outputs a mask with a list of all the applications.
 You then see a list comparable to the list of all defined users, which is shown in the diagram on [page 185](#).
 You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.

+ - loads the next data record.
- - loads the previous data record.
++ - loads the last data record.
— - loads the first data record.

The PAC of the application must be globally unique for global applications, and unique within the menu for local applications. The maximum length of *PAC* is four (alphanumerical) characters.

Comm.
Here you enter a comment about the application in a maximum of 70 characters.

User and application number

Here you enter a comment about the application in a maximum of 70 characters.

User and application number

The user number is a freely selectable four-digit number. It is used together with the two-digit application number to define the index key of the start sequence. The default of *0000 / 00* means that no start sequence is defined.

Network address of application

The network address is made up of the name of the application (*PTNNAME*) to which the connection is to be set up (max. length 8 characters) and the name of the processor (*PRONAME*) on which the application is running (max. length 8 characters). For partner type UCON the default value is \$CONSOLE and for TIAM it is \$DIALOG.

When *PREDIALOG is entered, the network address will be determined by the user in a dialog (OMMFPRE mask) mode before the connection is opened.

Application type (TIAM,UTM,DCAM,etc.)

The type of the application is entered here. Default=TIAM. All types known to OMNIS are permitted: TIAM, DCAM, UTM, UCON

When *PREDIALOG is entered, the application type will be determined by the user in a dialog (OMMFPRE mask) mode before the connection is opened.

Route (I=INDIRECT, D=DIRECT, M=MUX)

The options you can choose are "indirect" (I), "direct" (D) and "multiplex" (M).

ROUTE=M is possible only for UTM applications. Default=D.

Suppress LINE message (Y/N)

If you enter Y, all line messages arriving from the applications are suppressed. In other words, you will not receive responses to commands you input.

If you enter N, all line messages arriving from the applications are output. Default=N.

Name of substitute application (APP)

defines the name of the "simulated terminal" used by OMNIS to set up the connection to the application. The application knows OMNIS by this name.

appname may also be specified as a partially-qualified entry in the form of a three-character prefix, followed by *. In such cases, OMNIS independently creates a unique name by using the prefix and thus enables multiple instances of applications to be opened.

If ONE-USER-ONE-CONNECTION=NO has been set, even a local application may be opened concurrently more than once.

If no *appname* is specified, OMNIS generates a unique name.

Name of P key set

Here you can enter the name of the P key set created with PLUS for the application (max. length 8 characters); see also [page 220](#) and the "TIAM" manual [14].

Connection password (CPASS)

Here you enter the password required by the application during the connection setup (max. length 4 characters; see also the CPASS operand in the OPNCON command).

Example: C'xxxx' or X'yyyyyyyy'

Because of the input conventions for a password, the length of the input field is 11 characters.

Connection message (CMMSG)

Message sent to the application during the connection setup (max. length 30 characters; see also the CMMSG parameter in the OPNCON command).

Next entry

The menu name and the PAC of the next data record awaiting processing can be entered here (max. 8 characters in length). If the previous action was "Write" (*W*) or "Delete" (*D*), this data record is loaded as soon as the specified action has been executed.

When you input:

- * outputs a mask with a list of all the applications.
You then see a list comparable to the list of all defined users, which is shown in the diagram on [page 185](#).
You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + - loads the next data record.
- - loads the previous data record.
- ++ - loads the last data record.
- - loads the first data record.

Next function

By inputting the appropriate number, you can select another mask. With function 9 you return to the *Function Menu* mask. Actions entered in the *Read/write/delete entry* field are no longer executed. Default=3.

Meaning of the fields of the follow-up mask

Read/write/delete entry

Here you specify whether the entered data is to be read (*R*), written (*W*) or deleted (*D*).
Default=*R*.

Application switch (S=SCREEN, P=PKEY, A=ALL, N=NO, '='=STD)

Here you specify whether the screen contents and/or the P keys are saved when the application is exited.

If you specify *SCREEN* (*S*), the screen contents are saved and then output on the terminal the next time the application is selected.

Since the saved screen contents are sent in physical form, the screen is deleted after the next input in line mode.

If a screen could not be saved, the "Restart message" function is used when the application is selected again (see below).

If you specify *PKEY* (*P*), the P keys are saved for the specific terminal.

If you specify *ALL* (*A*), both the screen contents and the P keys are saved.

If you specify *NO* (*N*), neither the screen contents, not the P keys are saved.

STD: The *OPTION* or *SET* setting of OMNIS applies.

Automatic disconnection (Y/N)

If you specify *Y*, the connection to the application is cleared down when the application is exited. The release of the connection is not displayed. If you specify *N* (=default), the connection remains up. The screen dump function takes preference over this function.

Restart message (X/C)

Here you specify whether the restart message is to be in hexadecimal code (*X*) or a character string (*C*=default).

The screen dump function takes preference over the *Restart message* function.

Length of the restart message

In format or phys mode the default message is sent if the length is 0 (=default); if the length is >0, the restart message specified under *Contents* is sent to the application.

The maximum length of the restart message is 60 characters for a character string, or 30 characters for hexadecimal code.

Contents

The message specified here is sent to the application. Blanks can also be specified at the end of the character string.

UTM: SIGNON data (B=BOTH, U=USER, N=NONE, S=SUPPRESS)

Default=*B*. This field is evaluated only for multiplex connections (*ROUTE=M*). If you enter *B* (BOTH), the user name (USER) and password are transferred to UTM. The user name and password must be generated in UTM.

If you enter *U* (USER), only the user name (USER) is transferred to UTM. This is the best specification if the user has been generated in UTM without a password (i.e. the password consists of blanks).

If you enter *N* (NONE), no BIND data is transferred to UTM. In this case, automatic logon works only if the UTM application has been generated without a user (USER).

If you enter *S* (SUPPRESS), no SIGNON data is sent to UTM, even if specified in the *Application Menu*.

In the "Functions and Commands" manual [1], you will find a section on the "Special aspects of using OMNIS-MENU", which describes what data is transferred by OMNIS-MENU to UTM in OPEN SESSION, depending on the various entries in the *Login* and *Application Menu* masks and the generation with OMNISMD (*B/U/N/S*) and also how UTM reacts with different generations.

TIAM: LOGOFF BUT not allowed (Y/N)

This parameter applies only to TIAM applications. If you specify *Y*, the connection to the partner is cleared down after the TIAM command *LOGOFF BUT*, i.e. the *CLSCON pac* command is sent to OMNIS.

If you specify *N*, *LOGOFF BUT* remains effective, and the subsequent *LOGON* can be executed. Default=*N*.

DCAM: Lower case letters (Y/N)

This parameter applies only to DCAM applications. Lower case letters are either sent on to DCAM applications (*Y*) or not sent on (*N*=default).

Code (S=SYSCODE/B=BINARY)

The data is sent in EBCDIC (*S*) or transparently (*B*) to DCAM applications.

Protocol (O=OMNIS, D=DSSIM)

If you enter *D* (DSSIM), OMNIS behaves like a real terminal for a DCAM application. If you enter *O* (OMNIS), OMNIS behaves like a program for a DCAM application. An exact description of this behavior is provided in the [section "Prerequisites for DCAM programming" on page 99](#).

Next entry

The menu name and the PAC of the next data record awaiting processing can be entered here (max. 8 characters in length). If the previous action was "Write" (*W*) or "Delete" (*D*), this data record is loaded as soon as the specified action has been executed.

When you input:

- * outputs a mask with a list of all the applications.
 You then see a list comparable to the list of all defined users, which is shown in the diagram on [page 185](#).
 You can page in the list by entering +,- or ++, --. You can also take an entry from the list by marking it with *x* and switch back to the definition mask.
- + - loads the next data record.
- - loads the previous data record.
- ++ - loads the last data record.
- - loads the first data record.

Next function

By inputting the appropriate number, you can select another mask. With function 9 you return to the *Function Menu* mask. Actions entered in the *Read/write/delete entry* field are no longer executed. Default=3.

The table below provides an overview of the fields to which particular attention should be paid:

	TIAM	UTM	DCAM	UCON
PTN(NAME)	k	o	o	k
PRO(NAME)	x	x	x	x
PAC	o	o	o	o
CPASS	n	k	k	k
CMSG	n	n	k	k
SIGNON data	n	k only for MUX	n	n
LOGOFF BUT	k	n	n	n
LCASE	n	n	k	n
CODE	n	n	k	n
Protocol	n	n	k	n

Legend:

- o This field is mandatory
- k This field is optional
- n This field is not evaluated
- x This field is mandatory for indirect connections, and optional for direct connections.

Define Parameters

You can call up the *Define Parameters* mask by entering *4* in the *Function Menu* mask (see [page 180](#)). The data is stored in *omm.gentab*.

```

Function: OMMGGEN          O M N I S - M E N U          14:58:35/2009-12-10
                        Define Parameters
-----
Read/write generation data (R/W)          R

Comm. GLOBALE PARAMETER VON OMNIS-MENU UND OMNIS
One user - one connection (Y/N)          Y
Display Logout mask (Y/N)                Y
Modify Login mask (Y/N)                  Y
Delete P keys (Y/N)                      Y
Timeout (up to 1440 minutes)              0015
Retain connections (Y/N)                  Y
Maximum permissible number of Login attempts 03 (0: unlimited)
Switch character                           @
Password complexity (Length,Level,Duration) 4 NONE 002
Textkey length                             6
Go-back and logout key                     K12          P12
OMNIS command key                          K11          P11
Implicit application switching               K13          P13 Dialog Y
Next function 4 (9: Function menu)

-----
F1=Help F3=Quit K1=Cancel
OMM0618 RECORD READ FROM FILE

```

Figure 23: Define Parameters

In this mask you must fill in all fields which are not defaulted.

Meaning of the fields

Read/write generation data (R/W)

Here you specify whether the entered data is to be read (*R*) or written (*W*).
Default=*R*.

One user - one connection (Y/N)

If you enter *Y*, a user can be logged on at OMNIS-MENU only once at a time.
If you enter *N*, several users can work with OMNIS-MENU under the same user name at the same time. If *N* is specified, the connections cannot therefore be retained when the user logs off from OMNIS-MENU.

Display logout mask (Y/N)

If you enter *Y*, the logout mask is output during the logout process.

Modify login mask (Y/N)

If you enter *Y*, the *Login* mask in OMNIS-MENU can be modified. The information is read from the file assigned in the start procedure with *LINK-NAME=OMMAUTH*. The file can be modified during operation and is effective the next time a user logs on. If you enter *N*, the default mask is output.

Delete P keys (Y/N)

Here you define whether the contents of the P keys are deleted (*Y*) or retained (*N*) when logout is performed from OMNIS-MENU.

Timeout

If there is no activity at the terminal for the time specified here, the connection to the terminal is cleared down. If you specify *0000* (=default), there is no time limit.

Retain connections (Y/N)

The connections of a terminal logged off by timeout are kept up (*Y*) or cleared down (*N*=default). The open connections can only be used at the same terminal.

Maximum possible number of login attempts

Once the number specified here of failed attempts to log on to OMNIS-MENU has been reached (*0*=infinite), the terminal is locked. All active connections are cleared down, in accordance with *HALT=NO*.

The lock can be unlocked by the OMNIS-MENU administrator. When OMNIS-MENU is terminated, all terminal locks are unlocked.

Switch character

You specify here which printable character must be used before the PAC to switch applications.

_: No characters apply.

Password complexity (Length,Level,Duration)

You specify here which conditions the user passwords must meet.

Length Specify the minimum length of the password. You can specify a value between *0* (no password necessary) and *8*.

Level You can specify the following values here:

NONE any string with the length specified in *Length* is a valid password.

MIN a maximum of two consecutive characters in the password can be the same. The password consists of at least one character.

MED a maximum of two consecutive characters in the password can be the same, and the password must contain at least one letter and one digit. The password consists of at least 2 characters.

MAX a maximum of two consecutive characters in the password can be the same, and the password must contain at least one letter, one digit and a special character. Special characters are any characters other than a-z, A-Z and 0-9.
The password consists of at least 3 characters.

Duration A value between 0 and 180 specifies the period in days before a password must be changed. 2 weeks before the end of the period, the user is requested after logging on to change the password.
At the end of the period, the user can no longer log on.
The administrator must then define a new password for the user with OMNISMD.

Go-back and logout key

One of the K1-K13 keys must be defined as a go-back and logout key. The P1-P20 keys can also be defined as a go-back and logout key. The K key must be defined. The default is *P:00*, meaning that the K key defined as the go-back key is not loaded to a P key.

OMNIS command key

The K/P key assigned here enables authorized users to switch to OMNIS command mode, both from the *Application Menu* and from an application.
The K/P key is used in a menu only if it is not already assigned to an application. Without a K key, no P key can be assigned.

Implicit application switching

The K/P key assigned here enables implicit application switching for authorized users, both from the *Application Menu* and from an application.
The uniqueness of the P key must be guaranteed. The same P key must not be specified in the operands OMNIS-PKEY and RESET-PKEY.
When Dialog Y is specified, the functionality of implicit application switching applies also in the dialog with partner applications.

Next function

By inputting the appropriate number, you can select another mask. With function 9 you return to the *Function Menu*. Actions entered in the *Read/write/delete entry* field are no longer executed. Default=4.

Print Tables

For maintenance reasons, it is helpful to be able to print out the created files in a readable form. For this purpose, OMNISMD provides the *Print Tables* mask.

You can call up the *Print Tables* mask by entering 5 in the *Function Menu* (see [page 180](#)).

```

Function: OMMPRT          O M N I S - M E N U          12:00:34/2002-01-22
                          Print Tables
-----
The tables listed below can be prepared for printing
via SYSLST.
Please select the appropriate table  1

User table           : 1
Menu table           : 2
Application table    : 3
All tables           : 9

Next function  5 (9: Function Menu)
-----
F1=Help F3=Quit K1=Cancel SEND=Send

```

Figure 24: Print Tables

Enter the number of the table you want next to the *Please select the appropriate table* prompt, and submit it with the SEND key. Default=1.

You can output each table individually to SYSLST, or all tables at once. You can output them to a printer with the following command:

```
/PRINT-FILE file,LAYOUT-CONTR=PAR(CONTR-CHAR=EBCDIC)
```

Using the OMNISFC conversion program, you can also present the current configuration in simplified form, and in readable form for OMNISKD (see [page 225](#)).

Meaning of the fields

Please select the appropriate table

- 1 User table
All entries of the *omm.usertab* file are output ready for printing to SYSLST.
- 2 Menu table
All entries of the *omm.menutab* file are output ready for printing to SYSLST.
- 3 Application table
All entries of the *omm.appltab* file are output ready for printing to SYSLST.
- 9 All tables
All files are output ready for printing to SYSLST.

Next function

You can select another mask by entering the appropriate number.
Function 9 returns you to the *Function Menu*. Default=5.

Edit start sequence

OMNISMD supports the editing of individual start-sequences in the text file with the help of the software product EDT. The start-sequence numbers and the text key length can be chosen freely for pure OMNIS-applications or determined by the user-, menu- and application structure for OMNIS-MENU.

OMNISMD offers the *Edit start sequence* mask to do so.

You reach this mask (function: *OMMGTXT*) in two ways:

- choose function 6 in the *Function Menu* mask (see [page 180](#))
- press key F4 in the *Define Application* mask (see [page 192](#))

```

Function: OMMGTXT          O M N I S - M E N U          11:58:57/2015-03-05
                          Edit start sequence
-----
Edit/Delete (E/D)        E
TEXT-FILE                = OMNIS.TEXT
TEXTKEY-LENGTH           = 6
START-SEQUENCE-NUMBER   = 000308
-----
USER-NUMBER-PRIORITY    = USER-MENU-APPLICATION
USER                     NAME = TESTUSER          USER-NUMBER = 0000
MENU                    NAME = TESTMENU          USER-NUMBER = 0000
APPLICATION              PATH = *                ,R416  USER-NUMBER = 0000
                                                                APPL-NUMBER = 00
-----
F1=Help DUE=Send F3=Quit K1=Cancel

```

Figure 25: Edit start sequence

Meaning of the fields

Edit/Delete = E/D

Specifies if the start-sequence should be edited or deleted.

TEXT-FILE = <filename 1..54>

Name of the text file that contains the start-sequence.

The name can be specified prior the call of OMNISMD using the command

```
/ADD-FILE-LINK LINK-NAME=SOMSTEXT.
```

In this case the field TEXT-FILE is protected.

TEXTKEY-LENGTH = <integer 4..7>

Specification of the text key length.

If mask OMMGMAL is the entry-point the text key length can be chosen freely, in case of OMMGAP1/2 it is fixed to 6 and protected.

START-SEQUENCE-NUMBER = <char-integer 4..7>

Specification the start-sequence number.

OMNISMD proposes one for OMNIS-MENU based on the following passed masks:

- OMMGUSR (Define User)
- OMMGMEN (Define User Menu)
- OMMGMEE (Define Menu Entry)
- OMMGAP1/2 (Define Application)

The specification USER-NUMBER-PRIORITY of the application definition chooses between the multiple user numbers of the 3 objects, which builds the start-sequence together with the application number.

9.4 Terminating OMNISMD

OMNISMD can be terminated with the F3 key. In the *Function Menu*, OMNISMD can also be terminated by selecting function 9.

An OMNISFC run is recommended after each OMNISMD run to ensure that the input files are uniform and up-to-date (e.g. for OMNISKD).

10 Configuration examples

10.1 Example 1: OMNIS configuration

The example used here (a network with three processors) is illustrated on the next page.

ENTER file for OMNIS on processor HST1:

```
/LOGON
/SET-DCAM-APPLICATION-LINK LINK-NAME=OMNIS,APPLICATION-NAME=OMNISADM -
    ,PROTECTION=PARAMETERS(CONNECTION-PASSWORD=C'QUIX'
/START-OMNIS
FILE CONFIG=OMNIS.CONF.HST1
END
/LOGOFF
```

ENTER file for OMNIS on processor HST2:

```
/LOGON
/SET-DCAM-APPLICATION-LINK LINK-NAME=OMNIS,APPLICATION-NAME=OMNISADM -
    ,PROTECTION=PARAMETERS(CONNECTION-PASSWORD=C'QUAX'
/START-OMNIS
FILE CONFIG=OMNIS.CONF.HST2
END
/LOGOFF
```

ENTER file for OMNIS on processor VAR3:

```
/LOGON
/SET-DCAM-APPLICATION-LINK LINK-NAME=OMNIS,APPLICATION-NAME=OMNISADM -
    ,PROTECTION=PARAMETERS(CONNECTION-PASSWORD=C'QUEX'
/START-OMNIS
FILE CONFIG=OMNIS.CONF.HST3
END
/LOGOFF
```

Configuration of the network

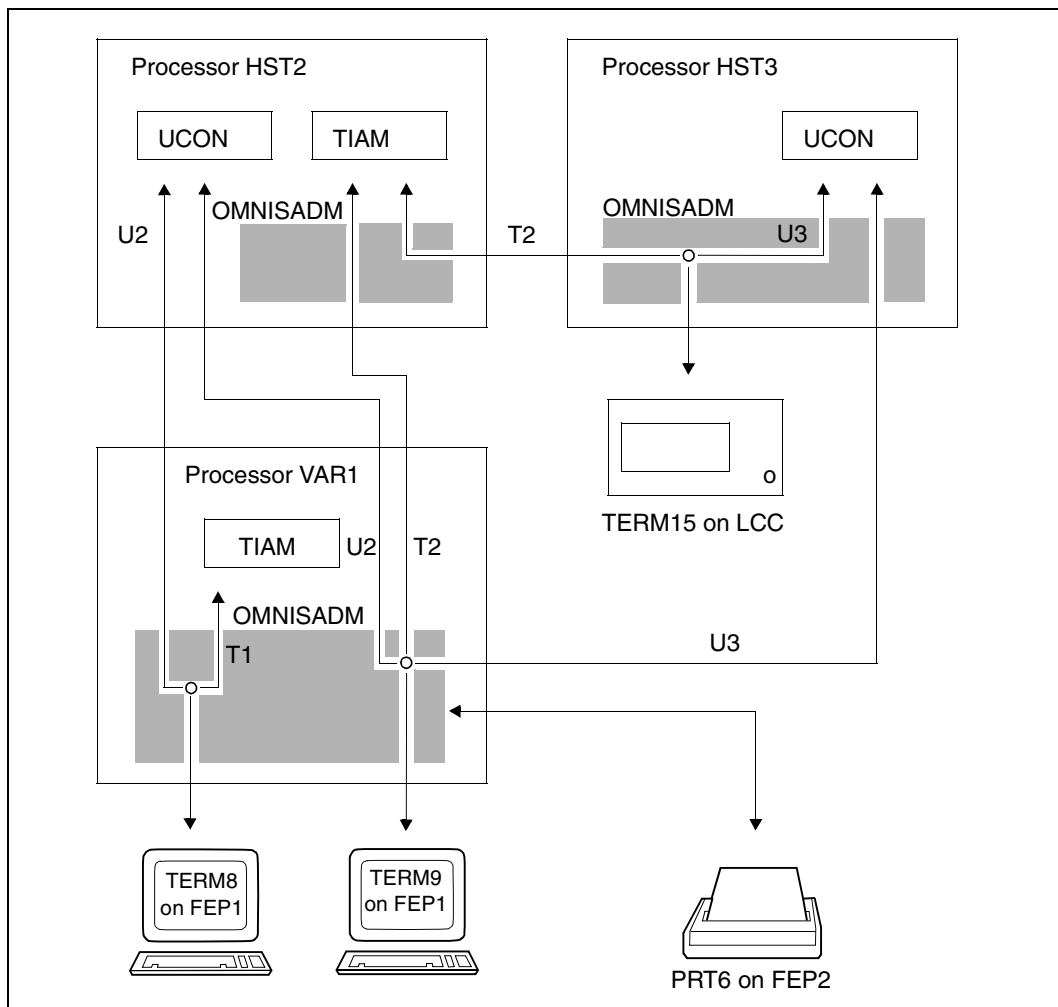


Figure 26: Configuration of a network with three processors

Configuration input file OMNIS.CONF.HST1.IN

```

DECLARE-OMNIS @2, PRONAME=HST2, CPASS=C'QUAX', PTNNAME=OMNIS
DECLARE-OMNIS @3, PRONAME=HST3, CPASS=C'QUEX', PTNNAME=OMNIS
DECLARE-HARDCOPY PRT6, PTNNAME=PRT6, PRONAME=FEP2
DECLARE-TERMINAL PTNNAME=TERM8, PRONAME=FEP1, TERMTYP=9750
    DECLARE-PARTNER U2, TYP=UCON, PRONAME=HST2, MSG=UID2, C'PASS2'
    DECLARE-PARTNER T1
DECLARE-TERMINAL PTNNAME=TERM9, PRONAME=FEP1, TERMTYP=8160
    D-P U2, TYP=UCON, PRONAME=HST2, MSG=UID1, C'PASS1'
    D-P U3, TYP=UCON, PRONAME=HST3, ROUTE=INDIRECT, MSG=UID3, C'PASS3'
    D-P T2, TYP=TIAM, PRONAME=HST2, ROUTE=INDIRECT, HCOPI=PRT6
    D-G U, ADD=(U2,U3)

```

Configuration input file OMNIS.CONF.HST2.IN

```

DECLARE-OMNIS @1, PRONAME=HST1, PTNNAME=OMNISADM, CPASS=C'QUIX'
DECLARE-OMNIS @3, PRONAME=HST3, CPASS=C'QUEX', PTNNAME=OMNIS

```

Configuration input file OMNIS.CONF.HST3.IN

```

DECLARE-OMNIS @1, PRONAME=HST1, PTNNAME=OMNISADM, CPASS=C'QUIX'
DECLARE-OMNIS @2, PRONAME=HST2, CPASS=C'QUAX', PTNNAME=OMNIS
DECLARE-TERMINAL PTRNAME=TERM15, PRONAME=HST3, TERMTYP=8161
    DECLARE-PARTNER T2, TYP=TIAM, PRONAME=HST2, ROUTE=INDIRECT
    DECLARE-PARTNER U3, TYP=UCON, MSG=UID6, C'PASS6'

```

Generating the configuration file in HST1

```

/ASSIGN-SYSDTA TO-FILE=OMNIS.CONF.HST1.IN
/SET-FILE-LINK LINK-NAME=OMNIS.CONF.HST1, FILE-NAME=SOMSKD
/START-OMNISKD

```

Generating the configuration file in HST2

```

/ASSIGN-SYSDTA TO-FILE=OMNIS.CONF.HST2.IN
/SET-FILE-LINK LINK-NAME=OMNIS.CONF.HST2, FILE-NAME=SOMSKD
/START-OMNISKD

```

Generating the configuration file in HST3

```

/ASSIGN-SYSDTA TO-FILE=OMNIS.CONF.HST3.IN
/SET-FILE-LINK LINK-NAME=OMNIS.CONF.HST3, FILE-NAME=SOMSKD
/START-OMNISKD

```

10.2 Example 2: OMNISKD / OMNIS-MENU

The preparations required for using **OMNISKD** and **OMNIS-MENU** are described for a predefined configuration (see [figure 27 on page 211](#)). These preparations include

- creating entries in the OMNIS and OMNIS-MENU files
- procedures for starting up OMNIS-MENU and OMNISKD (see [page 212](#)).

Applications of types TIAM, DCAM, UTM, UCON and ADMI and the information for the corresponding *OPNCON* commands required for the given configuration are then entered with the aid of OMNISKD and assigned to four user menus.

These four menus are assigned to six users.

An example of the generation in UTM required for the configuration is included (see [page 220](#)), followed by the special case of parallel multiplex connections between OMNIS and a UTM application (see [page 224](#)).

Configuration for the communication of eight terminals with seven partners in three different computers via OMNIS

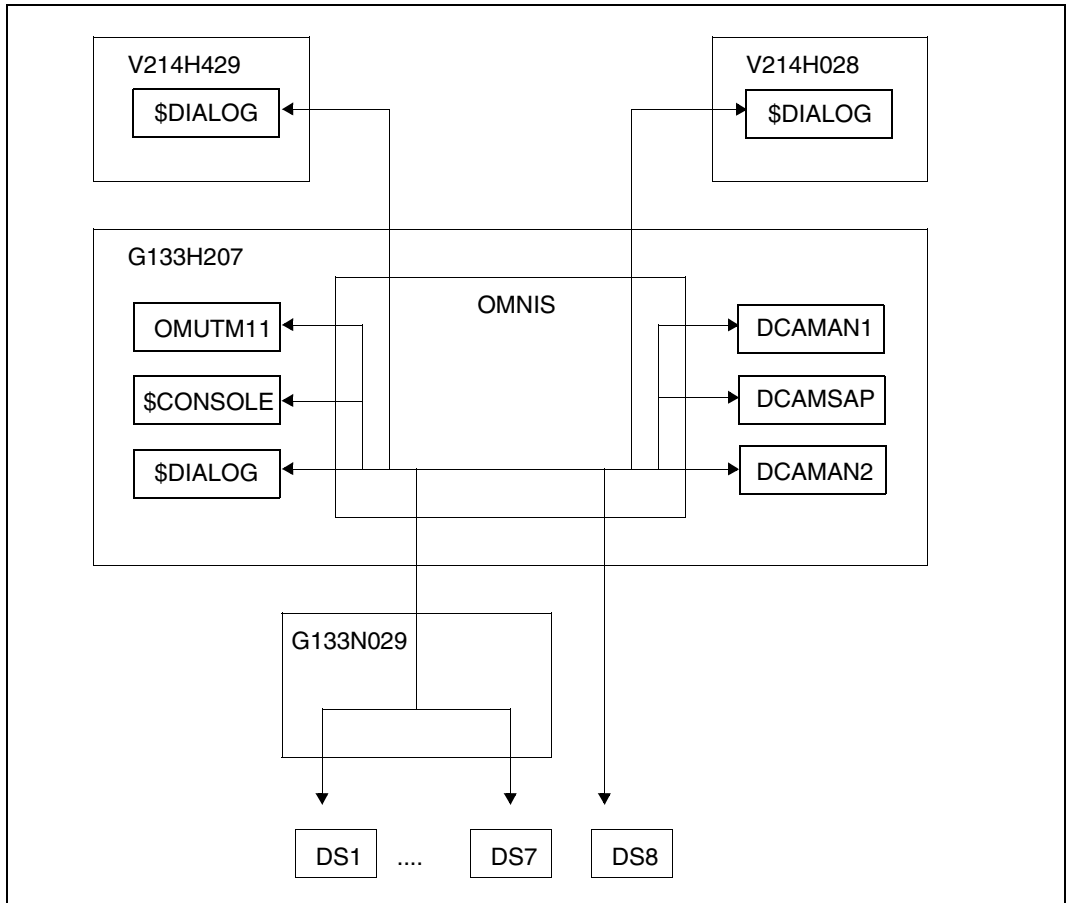


Figure 27: Communication of terminals via OMNIS

Entries in the startup file man.startup for declaring and calling the OMNIS-MENU exit routine

```
EXIT OMM,ACTION=CREATE,MODULES=OMM
SET EXIT=OMM,ACK=NO
SET CALL-K=3,CALL-I=YES
SET OPNSTART=YES
SET NPASS='OMM'
```

Procedure for starting up OMNIS-MENU

```

/BEGIN-PROCEDURE PARAMETERS=YES(PROCEDURE-PARAMETERS=( -
/      &VRS=035, -
/      &LANG=D, -
/      &OMSUID=OMSE, -
/      &OMMUID=OMSME, -
/      &PRE=MAN.OMM), -
/      ESCAPE-CHARACTER='&'), LOGGING=ALL
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/MODIFY-SDF-OPTIONS SYNTAX-FILE=$&OMSUID..SYSSDF.OMNIS.085.USER
/MODIFY-MSG-FILE-ASSIGN ADD=$&OMSUID..SYSMSA.OMNIS.085
/MODIFY-MSG-FILE-ASSIGN ADD=$&OMMUID..SYSMSA.OMNIS-MENU.035
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..GEN, LINK-NAME=OMMGEN
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..USER, LINK-NAME=OMMUSER
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..MENU, LINK-NAME=OMMMENU, -
/      BUFFER-LENGTH=STD(SIZE=16)
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..APPL, LINK-NAME=OMMAPPL
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..PKEY, LINK-NAME=OMMPKEY
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..TAC, LINK-NAME=OMMTAC
/SET-FILE-LINK FILE-NAME=$&OMMUID..&VRS..SYSFHS.&LANG, LINK-NAME=MAPLIB
/SET-FILE-LINK FILE-NAME=&PRE..BULLETIN, LINK-NAME=OMMAUTH
/SET-TASKLIB LIBRARY=$&OMMUID..MFHSROUT
/SET-DCAM-APPLICATION-LINK APPL-NAME=MAN, LINK-NAME=OMNIS
/DELETE-FILE FILE-NAME=&PRE..LOGGING.001
/SKIP-COMMANDS ERASERR
/.ERASERR SET-JOB-STEP
/START-OMNIS
  APPL PREFIX=MAN
  FILE CONFIG=&PRE..KON.AUS
  FILE LOGGING=&PRE..LOGGING
  FILE MODLIB=$&OMMUID..SYSLNK.OMNIS-MENU.035
  LENGTH TWORK=250
  LENGTH TEXTKEY=6
  FILE STARTUP=&PRE(STARTUP)
  FILE TEXT=&PRE..TEXT
/SET-JOB-STEP
/MODIFY-MSG-FILE-ASSIGN REM=$&OMMUID..SYSMSA.OMNIS-MENU.035
/MODIFY-MSG-FILE-ASSIGN REM=$&OMSUID..SYSMSA.OMNIS.085
/MODIFY-SDF-OPTIONS SYNTAX-FILE=*NONE
/      END-PROCEDURE

```



The minimum length that should be specified for the OMNIS-DST work area (*LENGTH TWORK*) is 250.

The length selected for the message receive buffer (*LENGTH IOAREA*) must not be smaller than the default value (4096).

OMNISKD input file KON.EIN

The order shown below is not mandatory. The menu-related applications are defined directly after the associated menu. For rules governing the order of declarations see [page 106](#).

```

*****
* GLOBAL PARAMETERS *
*****
DECLARE-PARAMETER ONE-USER-ONE-CONNECTION=YES -
                  ,MODIFY-AUTHORIZATION-MASK=YES -
                  ,DISCONNECTION-MASK=YES -
                  ,RESET-KKEY=12,RESET-PKEY=12 -
                  ,OMNIS-KKEY=11,OMNIS-PKEY=11 -
                  ,DELETE-PKEYS=YES -
                  ,TIMEOUT=0020 -
                  ,CALL-HOLD=AUTO -
                  ,CONNECTION-ATTEMPTS=03 -
                  ,SWITCH-CHARACTER='@' -
                  ,TEXTKEY-LENGTH=6 -
                  ,PROTECT-PW=(0,NONE,000)
*****
* START SEQUENCES *
*****
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=101 -
  ,COMMENT='START-SEQUENCE 101 FOR JIMBEAM' -
  ,RECORDS=( '.JIMBEAM LOGON OMSE,1' -
    , 'SHOW-JOB-STATUS' )
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=102 -
  ,COMMENT='START-SEQUENCE 102 FOR JIMBEAM' -
  ,RECORDS=( '.JIMBEAM LOGON OMSW,1' -
    , 'SHOW-USER-STATUS' )
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=201 -
  ,COMMENT='START-SEQUENCE 201 FOR JEAMBEAM' -
  ,RECORDS=( '.MINERAL LOGON OMSE,1' )
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=202 -
  ,COMMENT='START-SEQUENCE 202 FOR MINERAL' -
  ,RECORDS=( '.MINERAL LOGON OMSW,1' )
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=301 -
  ,COMMENT='START-SEQUENCE 301 FOR IDEFIX' -
  ,RECORDS=( '.IDEFIX LOGON OMSE,1' )
DECLARE-START-SEQUENCE START-SEQUENCE-NUMBER=302 -
  ,COMMENT='START-SEQUENCE 302 FOR IDEFIX' -
  ,RECORDS=( '.IDEFIX LOGON OMSW,1' )
*****
* MUX CONNECTIONS *
*****
DECLARE-MUX PTNNAME=OMUTM11,PRONAME=G133H207 -

```

Configuration examples

```
        ,CONNECT=START,AVAILABILITY=NO
*****
* P KEY SETS *
*****
DECLARE-PKEYSET COMMENT=' ' -
  ,MSG='''PTAST1''' -
  ,P1='''START-PROGRAMM FROM-FILE=$EDT''',EM,DUE' -
  ,P2='''STA L''',EM,DUE' -
  ,PTAST=PTAST1 -
  ,PKEYLIB=MAN.OMM.035.PKEY
DECLARE-PKEYSET COMMENT=' ' -
  ,MSG='''PTAST2''' -
  ,P1='''START-PROGRAMM FROM-FILE=$LMS''',EM,DUE' -
  ,PTAST=PTAST2 -
  ,PKEYLIB=MAN.OMM.035.PKEY
*****
* GLOBAL APPLICATIONS *
*****
DECLARE-APPLICATION PATH=( * ,CON1 ) -
  ,COM=' CONSOLE-G133H207 ' -
  ,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
  ,PRONAME=G133H207 -
  ,AUTOMATIC-CLSCON=Y,SAVE=NO -
  ,TYPE=UCON,ROUTE=D,LINE=N -
  ,CMMSG=@CONSOLE
DECLARE-APPLICATION PATH=( * ,D31 ) -
  ,COM=' DCAM31-D31-DCAMAN1 ' -
  ,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
  ,PTNNAME=DCAMAN1 ,PRONAME=G133H207 -
  ,LCASE=Y,CODE=S,PROTOCOL=O,AUTOMATIC-CLSCON=N,SAVE=NO -
  ,TYPE=DCAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,D32 ) -
  ,COM=' DCAM32-D32-DCAMAN2 ' -
  ,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
  ,PTNNAME=DCAMAN2 ,PRONAME=G133H207 -
  ,LCASE=N,CODE=S,PROTOCOL=D,AUTOMATIC-CLSCON=N,SAVE=NO -
  ,TYPE=DCAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,M31 ) -
  ,COM=' MUX31-M31-OMUTM11 ' -
  ,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
  ,PTNNAME=OMUTM11 ,PRONAME=G133H207,APPNAME=DS521028 -
  ,SIGNON=B,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
  ,TYPE=UTM ,ROUTE=M,LINE=N
DECLARE-APPLICATION PATH=( * ,M32 ) -
  ,COM=' MUX32-M32-OMUTM11 ' -
  ,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
  ,PTNNAME=OMUTM11 ,PRONAME=G133H207,APPNAME=DS531028 -
  ,SIGNON=N,AUTOMATIC-CLSCON=N,SAVE=NO -
```

```

,TYPE=UTM ,ROUTE=M,LINE=N
DECLARE-APPLICATION PATH=( * ,T11 ) -
,COM=' TIAM11-T11-V214H429 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=01 -
,PRONAME=V214H429 -
,AUTOLOGOFF-BUT=N,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=TIAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,T21 ) -
,COM=' TIAM21-T21-V214H028 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=01 -
,PRONAME=V214H028 -
,AUTOLOGOFF-BUT=N,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=TIAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,T31 ) -
,COM=' TIAM31-T31-G133H207 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=01 -
,PRONAME=G133H207,PTAST=PTAST1 -
,AUTOLOGOFF-BUT=N,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=TIAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,T32 ) -
,COM=' TIAM32-T32-G133H207 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=01 -
,PRONAME=G133H207,PTAST=PTAST2 -
,AUTOLOGOFF-BUT=N,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=TIAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=( * ,U31 ) -
,COM=' UTM31-U31-OMUTM11 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=02 -
,PTNNAME=OMUTM11 ,PRONAME=G133H207,APPNAME=OX511013 -
,SIGNON=B,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=UTM ,ROUTE=D,LINE=Y
DECLARE-APPLICATION PATH=( * ,U32 ) -
,COM=' UTM32-U32-OMUTM11 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=02 -
,PTNNAME=OMUTM11 ,PRONAME=G133H207,APPNAME=OX521013 -
,SIGNON=B,AUTOMATIC-CLSCON=N,SAVE=NO -
,TYPE=UTM ,ROUTE=D,LINE=N
*****
* DECLARATION OF ALL MENUS *
*****
DECLARE-MENU NAME=MENUADMI,USER-NUMBER=0000 -
,COM=' ' -
,ENTRY-LIST= -
((TYPE=ADMI,PATH=( *UNLOCK ,XXXX),TEXT=' *** UNLOCK ***',FKEY=03,PKEY=03)-
,(TYPE=APPL,PATH=( * ,CON1),TEXT=' CONSOLE-G133H207',KKEY=05,PKEY=05)-
,(TYPE=ADMI,PATH=( *UPDATE ,YYYY),TEXT=' *** UPDATE ***',KKEY=07,PKEY=07)-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-

```

Configuration examples

```
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=APPL,PATH=(MENUADMI,T11 ),TEXT='TIAM11-T11-
G133H207',KKEY=04,PKEY=04)-
,(TYPE=APPL,PATH=(MENUADMI,SAP ),TEXT='DCAMSAP-SAP-
SAPAPPL',KKEY=06,PKEY=06)-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
*****
* DECLARATION OF ALL LOCAL APPLICATIONS IN THE MENU *
*****
DECLARE-APPLICATION PATH=(MENUADMI,SAP ) -
,COM='DCAMSAP-SAP-SAPAPPL ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=00 -
,PTNNAME=SAPAPPL ,PRONAME=G133H207 -
,LCASE=N,CODE=S,PROTOCOL=0,AUTOMATIC-CLSCON=N,SAVE=NO -
,RESTART-MESSAGE=X'F2CF50414030204041660000' -
,TYPE=DCAM,ROUTE=D,LINE=N
DECLARE-APPLICATION PATH=(MENUADMI,T11 ) -
,COM='TIAM11-T11-G133H207 ' -
,USER-NUMBER=0000,APPLICATION-NUMBER=01 -
,PRONAME=G133H207-
,AUTOLOGOFF-BUT=N,AUTOMATIC-CLSCON=N,SAVE=SCREEN -
,TYPE=TIAM,ROUTE=D,LINE=N
DECLARE-MENU NAME=MENU1 ,USER-NUMBER=0000 -
,COM=' ' -
,ENTRY-LIST= -
((TYPE=APPL,PATH=( * ,T31 ),TEXT='TIAM31-T31-G133H207',FKEY=04,PKEY=04)-
,(TYPE=APPL,PATH=( * ,U31 ),TEXT='UTM31-U31-OMUTM11',KKEY=06,PKEY=06)-
,(TYPE=APPL,PATH=( * ,M31 ),TEXT='MUX31-M31-OMUTM11',KKEY=08,PKEY=08)-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=TEXT ,TEXT=' ' )-
,(TYPE=APPL,PATH=( * ,D31 ),TEXT='DCAM31-D31-DCAMAN1',FKEY=05,PKEY=05)-
,(TYPE=APPL,PATH=( * ,U32 ),TEXT='UTM32-U32-OMUTM11',KKEY=07,PKEY=07)-
,(TYPE=APPL,PATH=( * ,M32 ),TEXT='MUX32-M32-OMUTM11',KKEY=09,PKEY=09)-
```



```

, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' ))
DECLARE-MENU NAME=MENU2 ,USER-NUMBER=0000 -
,COM=' '-
,ENTRY-LIST= -
((TYPE=APPL,PATH=( * ,D31 ),TEXT='DCAM31-D31-DCAMAN1',FKEY=04,PKEY=04)-
, (TYPE=APPL,PATH=( * ,T31 ),TEXT='TIAM31-T31-G133H207',KKEY=06,PKEY=06)-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' ))
DECLARE-MENU NAME=MENU3 ,USER-NUMBER=0000 -
,COM=' '-
,ENTRY-LIST= -
((TYPE=APPL,PATH=( * ,T11 ),TEXT='TIAM11-T11-
V214H429',KKEY=04,PKEY=04)-
, (TYPE=APPL,PATH=( * ,T31 ),TEXT='TIAM31-T31-
G133H207',KKEY=06,PKEY=06)-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=APPL,PATH=( * ,T21 ),TEXT='TIAM21-T21-V214H028
',KKEY=05,PKEY=05)-
, (TYPE=APPL,PATH=( * ,T32 ),TEXT='TIAM32-T32-G133H207
',KKEY=07,PKEY=07)-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-
, (TYPE=TEXT ,TEXT=' ' )-

```

Configuration examples

```
      ,(TYPE=TEXT                ,TEXT='                '                )-
      ,(TYPE=TEXT                ,TEXT='                '                )-
      ,(TYPE=TEXT                ,TEXT='                '                )-
      ,(TYPE=TEXT                ,TEXT='                '                ))
*****
* DECLARATION OF ALL USERS                                     *
*****
DECLARE-USER  NAME=ADMIN    ,PASSWORD=*UNCHANGED,USER-NUMBER=0000  -
      ,COM='                '                '                '                '
      ,MENU=MENUADMI ,OMNIS-COMMAND-MODE=Y,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=000,POSITION=001  -
      ,INFO=C ' '
DECLARE-USER  NAME=USER1    ,PASSWORD=*UNCHANGED,USER-NUMBER=0001  -
      ,COM='                '                '                '                '
      ,MENU=MENU1    ,OMNIS-COMMAND-MODE=N,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=000,POSITION=001  -
      ,INFO=C ' '
DECLARE-USER  NAME=USER2    ,PASSWORD=*UNCHANGED,USER-NUMBER=0002  -
      ,COM='                '                '                '                '
      ,MENU=MENU2    ,OMNIS-COMMAND-MODE=N,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=000,POSITION=001  -
      ,INFO=C ' '
DECLARE-USER  NAME=USER3    ,PASSWORD=*UNCHANGED,USER-NUMBER=0003  -
      ,COM='                '                '                '                '
      ,MENU=MENU3    ,OMNIS-COMMAND-MODE=Y,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=000,POSITION=001  -
      ,INFO=C ' '
DECLARE-USER  NAME=USER4    ,PASSWORD=*UNCHANGED,USER-NUMBER=0004  -
      ,COM='                '                '                '                '
      ,MENU=MENU2    ,OMNIS-COMMAND-MODE=N,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=017,POSITION=001  -
      ,INFO=C 'PSE4WAGE901234567 '
DECLARE-USER  NAME=USER5    ,PASSWORD=*UNCHANGED,USER-NUMBER=0005  -
      ,COM='                '                '                '                '
      ,MENU=MENU1    ,OMNIS-COMMAND-MODE=N,EXCLUSIVE-PARTNER=STD  -
      ,LENGTH=000,POSITION=001  -
      ,INFO=C ' '

```

Procedure for starting up OMNISKD

```

/BEGIN-PROCEDURE PARAMETERS=YES(PROCEDURE-PARAMETERS=( -
/      &VRS=035, -
/      &LANG=D, -
/      &OMSUID=OMSE, -
/      &OMMUID=OMSME, -
/      &PRE=MAN.OMM), -
/      ESCAPE-CHARACTER='&'), LOGGING=ALL
/MODIFY-SDF-OPTIONS $&OMSUID..SYSSDF.OMNIS.085.USER
/MODIFY-MSG-FILE-ASSIGNMENT ADD=$&OMSUID..SYSMSA.OMNIS.085
/MODIFY-MSG-ATTRIBUTES TASK-LANGUAGE='&LONG'
/SET-JOB-STEP
/ASSIGN-SYSLST TO-FILE=&PRE..KON.LST
/ASSIGN-SYSDTA TO-FILE=&PRE(KON.EIN)
/SET-FILE-LINK FILE-NAME=&PRE..KON.AUS, LINK-NAME=SOMSKD
/SET-FILE-LINK FILE-NAME=&PRE..TEXT, LINK-NAME=SOMSTEXT
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..PKEY.IN, LINK-NAME=OMMPLUS
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..TAC, LINK-NAME=OMMTAC
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..GEN, LINK-NAME=OMMGEN
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..USER, LINK-NAME=OMMUSER
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..APPL, LINK-NAME=OMMAPPL
/CREATE-FILE FILE-NAME=&PRE..&VRS..MENU, -
/      SUPPORT=PUBLIC-DISK(-
/      SPACE=RELATIVE(PRIMARY-ALLOCATION=48, - /
SECONDARY-ALLOCATION=16))
/SET-JOB-STEP
/SET-FILE-LINK FILE-NAME=&PRE..&VRS..MENU, LINK-NAME=OMMMENU, - /
BUFFER-LENGTH=STD(SIZE=16)
/START-OMNISKD
/SET-JOB-STEP
/MODIFY-FILE-ATTRIBUTES FILE-NAME=&PRE..&VRS..MENU, -
/      SUPPORT=PUBLIC-DISK(SPACE=RELEASE(- /
NUMBER-OF-PAGES=ALL-RELEASABLE))
/REMOVE-FILE-LINK LINK-NAME=OMMMENU
/REMOVE-FILE-LINK LINK-NAME=OMMAPPL
/REMOVE-FILE-LINK LINK-NAME=OMMUSER
/REMOVE-FILE-LINK LINK-NAME=OMMGEN
/REMOVE-FILE-LINK LINK-NAME=OMMTAC
/REMOVE-FILE-LINK LINK-NAME=OMMPLUS
/REMOVE-FILE-LINK LINK-NAME=SOMSTEXT
/REMOVE-FILE-LINK LINK-NAME=SOMSKD
/ASSIGN-SYSLST TO-FILE=*PRIMARY
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/MODIFY-MSG-ATTRIBUTES TASK-LANGUAGE=*STD
/MODIFY-MSG-FILE-ASSIGNMENT REM-FILE=$&OMSUID..SYSMSA.OMNIS.085
/MODIFY-SDF-OPTIONS SYNTAX-FILE=*NONE
/END-PROCEDURE

```

Procedure for creating the P key set PKEYS1

```
/BEGIN-PROCEDURE PARAMETERS=YES(PROCEDURE-PARAMETERS=( -
/      &VRS=035, -
/      &PRE=MAN.OMM), -
/      ESCAPE-CHARACTER='&'), LOGGING=ALL
/MODIFY-JOB-SWITCHES OFF=(4,5)
/ASSIGN-SYSDTA TO-FILE=&PRE..&VRS..PKEY.IN
/START-PROG FROM-FILE=$PLUS
/SET-JOB-STEP
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROCEDURE
```

A special generation is also required on the UTM side for the example above.

Generation in UTM

```
/BEGIN-PROCEDURE PAR=YES(PROC-PAR=( &GEN=), ESC-CHAR='&' ), LOG=N
/ REMARK **
/ REMARK * CREATE KDCFILE FOR UTM APPLICATION *
/ REMARK **
/ ASSIGN-SYSLST TO-FILE=UTM.LIST.KDCDEF
/ ASSIGN-SYSDTA TO-FILE=*SYSCMD
/ START-KDCDEF
OPTION GEN=&GEN
*****
* APPLICATION-SPECIFIC INFORMATION *
*****
MAX APPLNAME=OMUTM11
MAX KB=32767, SPAB=8192, TRMSGLTH=32764, NB=32760
MAX KDCFILE=(OMUTM11, S)
MAX TASKS=5
MAX ASYNTASKS=4
MAX TERMWAIT=120
MAX LOGACKWAIT=120
*****
* U S E R S *
*****
USER ADMIN, PASS=ADMIN, PERMIT=ADMIN
USER USER1, PASS=USER1
USER USER2, PASS=USER2
USER USER3, PASS=USER3
USER USER4, PASS=USER3
USER USER5, PASS=USER5
```

```

*****
* ADMINISTRATION TACS *
*****
DEFAULT TAC PROGRAM=KDCADM
DEFAULT TAC ADMIN=YES
DEFAULT TAC TIME=300000
TAC KDCAPPL
TAC KDCDIAG
TAC KDCHELP
TAC KDCINF
TAC KDCLOG
TAC KDCLTERM
TAC KDCMUX
TAC KDCPTERM
TAC KDCSEND
TAC KDCSHUT
TAC KDCSWTCH
TAC KDCTAC
TAC KDCTCL
TAC KDCUSER
*****
* TACS FOR PROGRAM UNITS *
*****
DEFAULT TAC TYPE=D, CALL=BOTH, ADMIN=N
*
TAC TAC1A, PROGRAM=TAC1A
TAC TAC1B, PROGRAM=TAC1B
TAC TAC1C, PROGRAM=TAC1C, CALL=NEXT
TAC TAC1D, PROGRAM=TAC1D
TAC TAC2A, PROGRAM=TAC2A
TAC TAC2B, PROGRAM=TAC2B
TAC TAC2C, PROGRAM=TAC2C, CALL=NEXT
TAC TAC2D, PROGRAM=TAC2D
*****
* FUNKTION KEYS *
*****
*
SFUNC K1, TAC=TAC1A
SFUNC K3, TAC=TAC1B
*****
* MUX CONNECTION *
*****
*
MUX MAN, PRONAM=G133H207, CONNECT=Y, STATUS=ON, MAXSES=10

```

Configuration examples

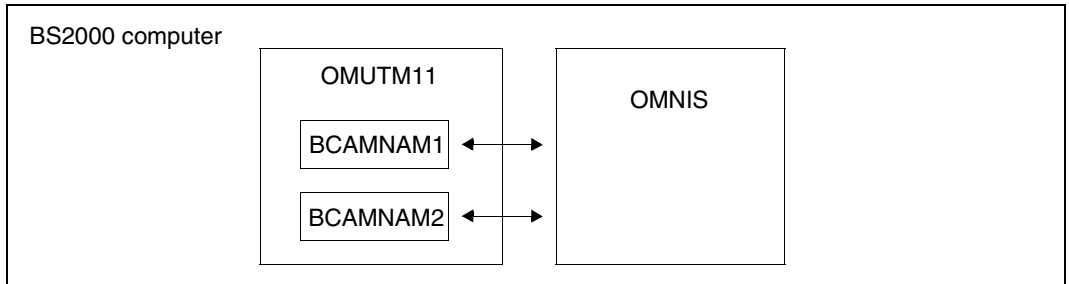
```
*****
* PTERMS / LTERMS WITHOUT MUX WITHOUT OMNIS *
*****
DEFAULT LTERM QAMSG=N
DEFAULT LTERM ANNOAMSG=Y
*
PTERM DS510025,PTYPE=T9750,LTERM=MX510025,PRONAM=G133N029
LTERM MX510025
*
PTERM DS510073,PTYPE=T9750,LTERM=MX510073,PRONAM=G133N029
LTERM MX510073
*
*****
* PTERM'S / LTERM'S WITH MUX *
*****
*
PTERM DS521028,PTYPE=T9750,LTERM=MX521028,PRONAM=G133N029,CONNECT=N
LTERM MX521028
*
PTERM DS531028,PTYPE=T9750,LTERM=MX531028,PRONAM=G133N029,CONNECT=N
LTERM MX531028
*
*****
* PTERM'S / LTERM'S WITHOUT MUX FOR OMNIS *
*****
*
PTERM
OX511013,PTYPE=T9750,LTERM=OX511013,PRONAM=G133H207,PROTOCOL=NO,CONNECT=N
LTERM OX511013
*
PTERM
OX521013,PTYPE=T9750,LTERM=OX521013,PRONAM=G133H207,PROTOCOL=NO,CONNECT=N
LTERM OX521013
*
PTERM
OX511029,PTYPE=T9750,LTERM=OM511029,PRONAM=G133H207,PROTOCOL=NO,CONNECT=N
LTERM OM511029
*
*****
* DESCRIPTION OF MUXROOT *
*****
ROOT MUXROOT
*****
* DESCRIPTION OF FORMAT HANDLING SYSTEM *
*****
FORMSYS TYPE=FHS
```

```
*****
* PROGRAM FOR ADMINISTRATION                                     *
*****
PROGRAM KDCADM,COMP=ILCS
*****
* PROGRAM FOR PROGRAM UNITS                                     *
*****
PROGRAM TAC1A,COMP=ILCS
PROGRAM TAC1B,COMP=ILCS
PROGRAM TAC1C,COMP=ILCS
PROGRAM TAC1D,COMP=ILCS
PROGRAM TAC2A,COMP=ILCS
PROGRAM TAC2B,COMP=ILCS
PROGRAM TAC2C,COMP=ILCS
PROGRAM TAC2D,COMP=ILCS
END
/END-PROCEDURE
```

Special case: Parallel multiplex connections between OMNIS and a UTM application

For the generation in UTM, a number of application names (corresponding to the number of connections) are to be assigned to a UTM application and the corresponding MUX statements issued. In the OMNIS generation, *DECLARE-MUX* statements must be entered in the OMNIS configuration file for all these application names.

In the given example, two parallel connections are to be maintained between OMNIS and the partner *OMUTM11*, as shown in the following diagram:



Generation in UTM

The following statements must be inserted in the file *KDCFILE*:

```
BCAMAPPL BCAMNAM1
BCAMAPPL BCAMNAM2
```

In this way the UTM application *OMUTM11* can be accessed via three names:

- *OMUTM11* (corresponding to the MAX statement)
- *BCAMNAM1* (corresponding to the BCAMAPPL statement)
- *BCAMNAM2* (corresponding to the BCAMAPPL statement)

The MUX statements in the example should be supplemented by the following:

```
MUX MAN, PRONAM=G133H207, CONNECT=Y, STATUS=ON, MAXSES=10, BCAMAPPL=BCAMNAM1
MUX MAN, PRONAM=G133H207, CONNECT=Y, STATUS=ON, MAXSES=10, BCAMAPPL=BCAMNAM2
```

Generation in OMNIS

In the configuration file, the application names *BCAMNAM1* and *BCAMNAM2* must be declared as separate multiplex partners:

```
DECLARE-MUX PTNNAME=BCAMNAM1, PRO=G133H207
DECLARE-MUX PTNNAME=BCAMNAM2, PRO=G133H207
```

OMUTM11, *BCAMNAM1* or *BCAMNAM2* can be specified as the partner name in the OMNIS command *OPNCON* or on entering the application in the application table. One connection is then set up for each name used.

11 OMNISFC: Converting configuration files

OMNISFC (OMNIS/OMNIS-MENU File Converter) is a utility routine with which the administrator converts OMNIS/OMNIS-MENU configuration files into readable OMNISKD input files.

It also allows the conversion of configuration files from earlier OMNIS and OMNIS-MENU versions; see the [section “Changing versions” on page 229](#) for details.

OMNISFC can process OMNIS configuration files as of version V6.0 and OMNIS-MENU configuration files as of version V2.0.

11.1 Working with OMNISFC

The following figure shows the relationship between the utility routines and the files for the configuration of OMNIS.

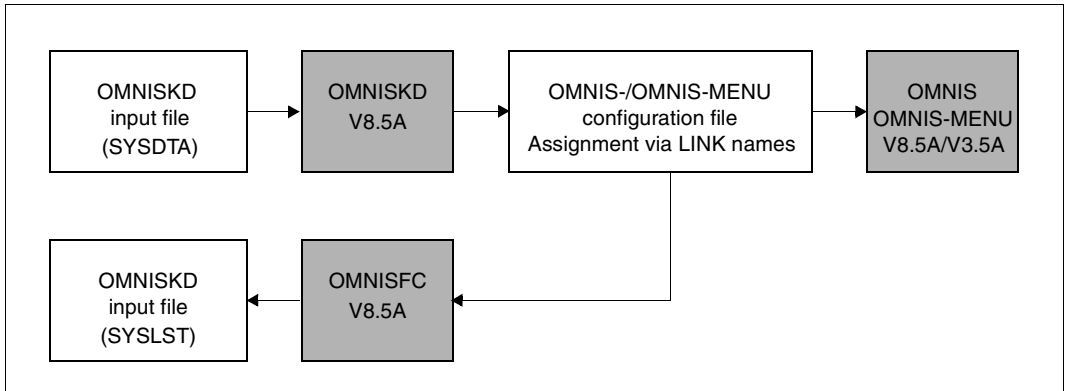


Figure 28: Relationship between utility routines and files for the configuration of OMNIS/OMNIS-MENU

OMNISKD reads the input file from SYSDTA.

The assignment of the file names for the configuration files occurs via link names (i.e., LINK names with the *SET-FILE-LINK* command in the OMNISKD or OMNISFC start procedure).

OMNISFC writes the generated OMNISKD input file to SYSLST.

OMNISFC can be called via the *START-PROGRAM* command or via *START-OMNISFC*. In order to do this, the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* must be enabled.

Example of an OMNISFC startup procedure

```

/BEGIN-PROCEDURE PARAMETERS=NO, LOGGING=NO
/WRITE-TEXT '***** OMNISFC ***** : 035'
/REMARK OMNISFC READS IN THE FOUR MENU FILES OMMGEN, OMMUSER
/REMARK OMMAPPL, OMMMENU (OLD + NEW FORMAT) AND BUILDS AN
/REMARK OMNISKD DEFINITION FILE FROM THEM
/SET-FILE-LINK FILE-NAM=KR.TAB.TACTAB, LINK-NAME=OMMTAC
/SET-FILE-LINK FILE-NAM=KR.TAB.GEN, BUFFER-LENGTH=STD(SIZE=2), -
LINK-NAME=OMMGEN
/SET-FILE-LINK FILE-NAM=KR.TAB.USER, BUFFER-LENGTH=STD(SIZE=2), -
LINK-NAME=OMMUSER
/SET-FILE-LINK FILE-NAM=KR.TAB.MENU, BUFFER-LENGTH=STD(SIZE=16), -
LINK-NAME=OMMMENU
/SET-FILE-LINK FILE-NAM=KR.TAB.APPL, BUFFER-LENGTH=STD(SIZE=2), -
LINK-NAME=OMMAPPL
/SET-FILE-LINK LINK-NAME=SOMSKD, -
/ FILE-NAME=KR.KON.AUS
/SET-FILE-LINK LINK-NAME=SOMSTEXT, ACCESS-METHOD=ISAM(KEY-LEN=8), - /
FILE-NAME=KR.TEXT
/SET-FILE-LINK LINK-NAME=OMMPLUS, FILE-NAME=KR.PKEY.IN, -
/ ACCESS-METHOD=SAM
/ MODIFY-MSG-FILE-ASSIGNMENT ADD-FILE=SYSMSA.OMNIS.085
/ SET-JOB-STEP
/ ASSIGN-SYSLST TO=*LIBRARY-ELEMENT(LIBRARY=KR.B, - /
ELEMENT=GEN.TXT, TYPE=S)
/ ASSIGN-SYSDTA TO=*SYSCMD
/REMARK MODIFY-SDF-OPTIONS SYNTAX-FILE=$OMNIS.SYSSDF.OMNIS.085.USER
/START-PROGRAM *M($OMNIS.SYSLNK.OMNIS.085, OMNISFC, RUN-MOD=ADV)
/ ASSIGN-SYSLST TO=*PRIMARY
/ REMOVE-FILE-LINK LINK-NAME=OMMGEN
/ REMOVE-FILE-LINK LINK-NAME=OMMUSER
/ REMOVE-FILE-LINK LINK-NAME=OMMMENU
/ REMOVE-FILE-LINK LINK-NAME=OMMAPPL
/ REMOVE-FILE-LINK LINK-NAME=OMMPLUS
/ REMOVE-FILE-LINK LINK-NAME=OMMTAC
/ REMOVE-FILE-LINK LINK-NAME=SOMSKD
/ REMOVE-FILE-LINK LINK-NAME=SOMSTEXT
/ REMOVE-FILE-LINK LINK-NAME=BLSLIB01
/SET-JOB-STEP
/EXIT-PROCEDURE

```

If passwords are to be transferred, the OMNIS-MENU files (USERTAB) can also be used as a direct input for OMNISKD.

OMNISFC creates an OMNISKD input file (output to SYSLST) from the contents of the OMNIS-MENU tables. OMNISFC always generates at least 20 menu entries (1 page), even if less than 20 entries are defined. The additional menu entries are of type *TEXT* with *TEXT=''*.

You can still create TAC tables and text files using EDT (or by using PLUS to create the P key library) and then assign them via the corresponding link names.

Converting the text file

If a *GENTAB* of OMNIS-MENU V3.1A or higher exists, the keys of the text file are converted in accordance with the *TEXTKEY-LENGTH* operand (see [page 137](#)). With older *GENTAB* versions, the 8-byte key of the text file is converted to a 6-byte key. In all other cases, OMNISFC prompts you to enter the key length you require; the default value for the key length is 4.

OMNISFC always adds a *DECLARE-PARAMETER* statement to the OMNISKD input file, giving the *TEXTKEY-LENGTH* operand the value appropriate to the OMNISFC setting. The *RESET-KKEY* operand is not specified in the *DECLARE-PARAMETER* statement unless it has been defined in the *GENTAB* table.

The following examples show a number of different options for converting the 8-byte key of a text file using OMNISFC:

Text file:

```
00100101Text1
00100102Text2
00200101Text3
00200102Text4
00200201Text5
03000101Text6
03000102Text7
```

A key length of 7 yields:

```
Start sequence 10010 with Text1 and Text2
Start sequence 20010 with Text3 and Text4
Start sequence 20020 with Text5
Start sequence 300010 with Text6 and Text7
```

A key length of 6 yields:

```
Start sequence 1001 with Text1 and Text2
Start sequence 2001 with Text3 and Text4
Start sequence 2002 with Text5
Start sequence 30001 with Text6 and Text7
```

A key length of 4 yields:

```
Start sequence 10 with Text1 and Text2
Start sequence 20 with Text3, Text4 and Text5
Start sequence 300 with Text6 and Text7
```

Converting P key sets

OMNISFC cannot convert P key sets generated by PLUS (R modules) from the P key library; it can only operate on a PLUS input file.

Before calling OMNISFC, you must first assign the PLUS input file (SAM file):

```
/SET-FILE-LINK LINK-NAME=OMMPLUS,FILE-NAME=<full-filename1 .. 54>
```

OMNISFC processes the following statements in the PLUS input file:

- Pi (define P key)
- MSG (output message)
- SAV (save P key set)
- DEL (delete P key set)

All other statements in a PLUS input file are skipped by OMNISFC.

OMNISFC reads all statements, beginning with the *DEL* statement up to the *SAV* statement, and generates a *DECLARE-PKEYSET* definition on SYSLST. The *DEL* and *SAV* statements themselves are not transferred to the *DECLARE-PKEYSET* statement.

OMNISFC transfers all *Pi* statements and makes the *MSG* operand from an *MSG* statement. A *SAV* statement is converted to the *PTAST* or *PKEYLIB* operand of the *DECLARE-PKEYSET* statement. A *DEL* statement starts a *DECLARE-PKEYSET* statement.

If the PLUS input file is not created with OMNISKD or is modified later, the *DEL* statements could be missing. These *DEL* statements are no longer required by OMNISFC. If they are missing, they are implicitly assumed at the beginning of the file and after every *SAV* statement. The *DEL* statements, however, can still exist explicitly in the input file. Like all other statements with the exception of *Pi*, *SAV* and *MSG*, they are ignored by OMNISFC.

In addition, the name of the P key set no longer has to be specified in the *SAV* statement. If this is missing, the default name assumed by PLUS, *PLUS.MAPLIB*, is assumed. This also corresponds to the general PLUS syntax for the *SAV* statement.

If the *SAV* statement is missing at the end of the file, the line ' ,PKEYLIB=PLUS.MAPLIB' is generated. This returns an error in the OMNISKD run, since the P key module name specification is missing. The *DECLARE-PKEYSET* statement, however, is then terminated and the subsequent entries are generated correctly.

11.2 Changing versions

11.2.1 Changing the OMNIS version

As OMNIS V8.5A and OMNISFC V8.5A can process OMNIS configuration files as of OMNIS V6.0A, the OMNIS configuration files (OMNISKD output files) need not always be recreated when changing the version of OMNIS.

On starting OMNIS, the configuration file is read, converted and processed for the current run, and a corresponding message is output.

Note, however, that if you are planning to use the new version frequently, it is advisable to upgrade the configuration file for the new version to avoid the need for a fresh conversion whenever you start OMNIS.

The individual steps involved in upgrading are illustrated in the figure below:

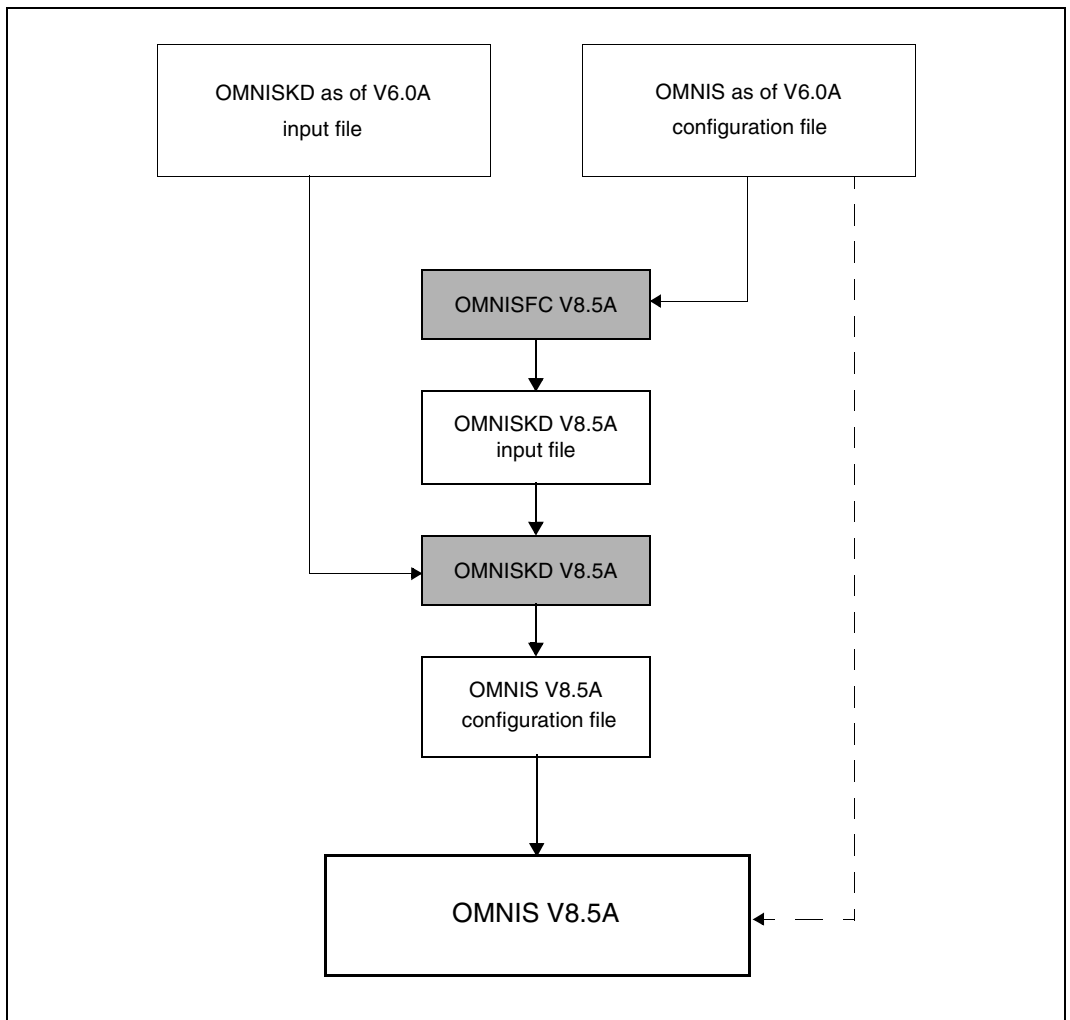


Figure 29: Changing from OMNIS Version as of V6.0 to OMNIS V8.5A

11.2.2 Changing the OMNIS-MENU version

Changing from OMNIS-MENU V3.0A through V3.4A to V3.5A

The OMNIS-MENU configuration files of V3.0A through V3.4A must be taken over by means of an OMNISFC V8.5A run and an OMNISKD V8.5A run. The old configuration files cannot be automatically taken over by OMNIS-MENU V3.5A, since the user table is also updated (i.e., written to).

The OMNIS configuration file should also be upgraded here at the same time.

The individual steps involved in upgrading from OMNIS-MENU V3.0A through V3.4A to OMNIS-MENU V3.5A are illustrated in the figure below.

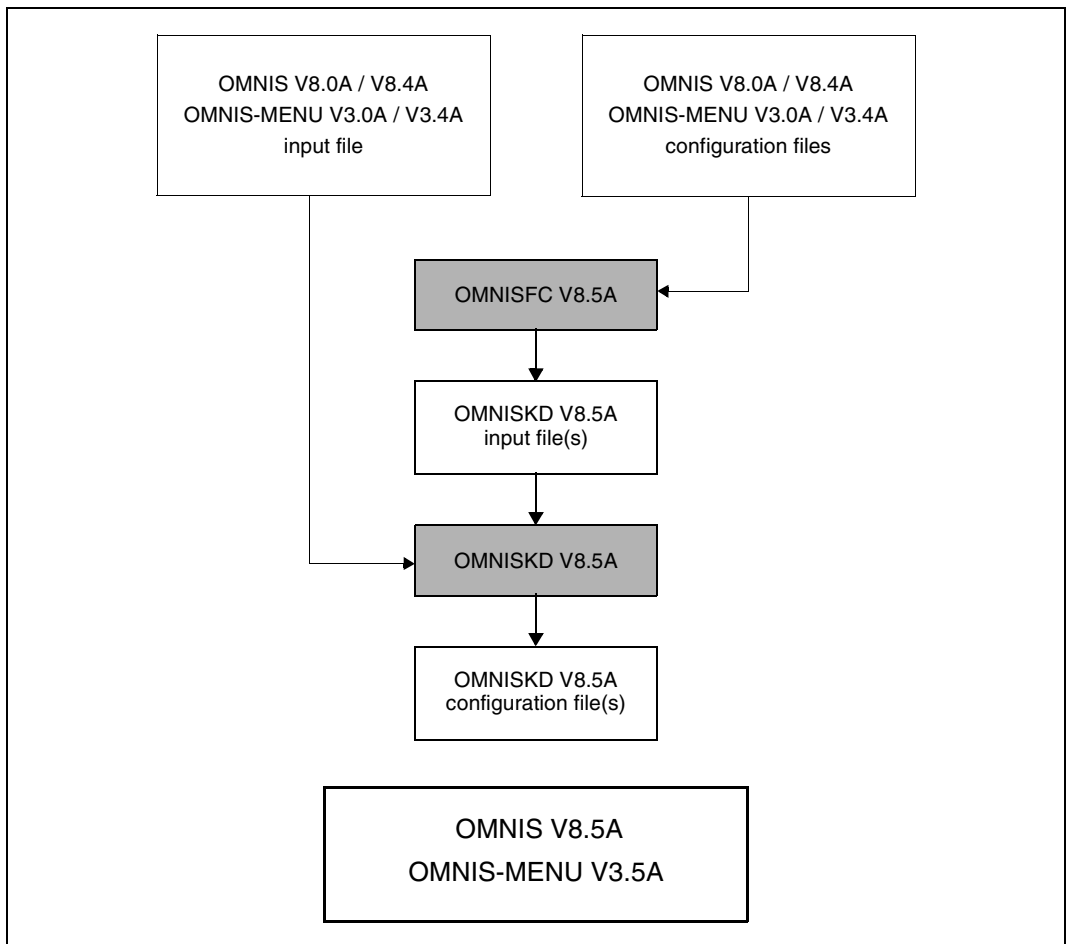


Figure 30: Changing from OMNIS-MENU V3.0A through V3.4A to OMNIS-MENU V 3.5A

Changing from OMNIS-MENU V2.0A to V3.5A or from OMNIS-MENU V2.1A to V3.5A

The OMNIS-MENU configuration files of V2.0A or V2.1A must be taken over by means of a OMNISFC V8.5A run and a OMNISKD V8.5A run. The old configuration files cannot be automatically taken over by OMNIS-MENU V3.5A, since the user table is also updated (i.e., written to). Existing OMNIS configuration files must also be upgraded accordingly.

The individual steps involved in upgrading from OMNIS-MENU V2.0A or V2.1A to OMNIS-MENU V3.3A are illustrated in the figure below.

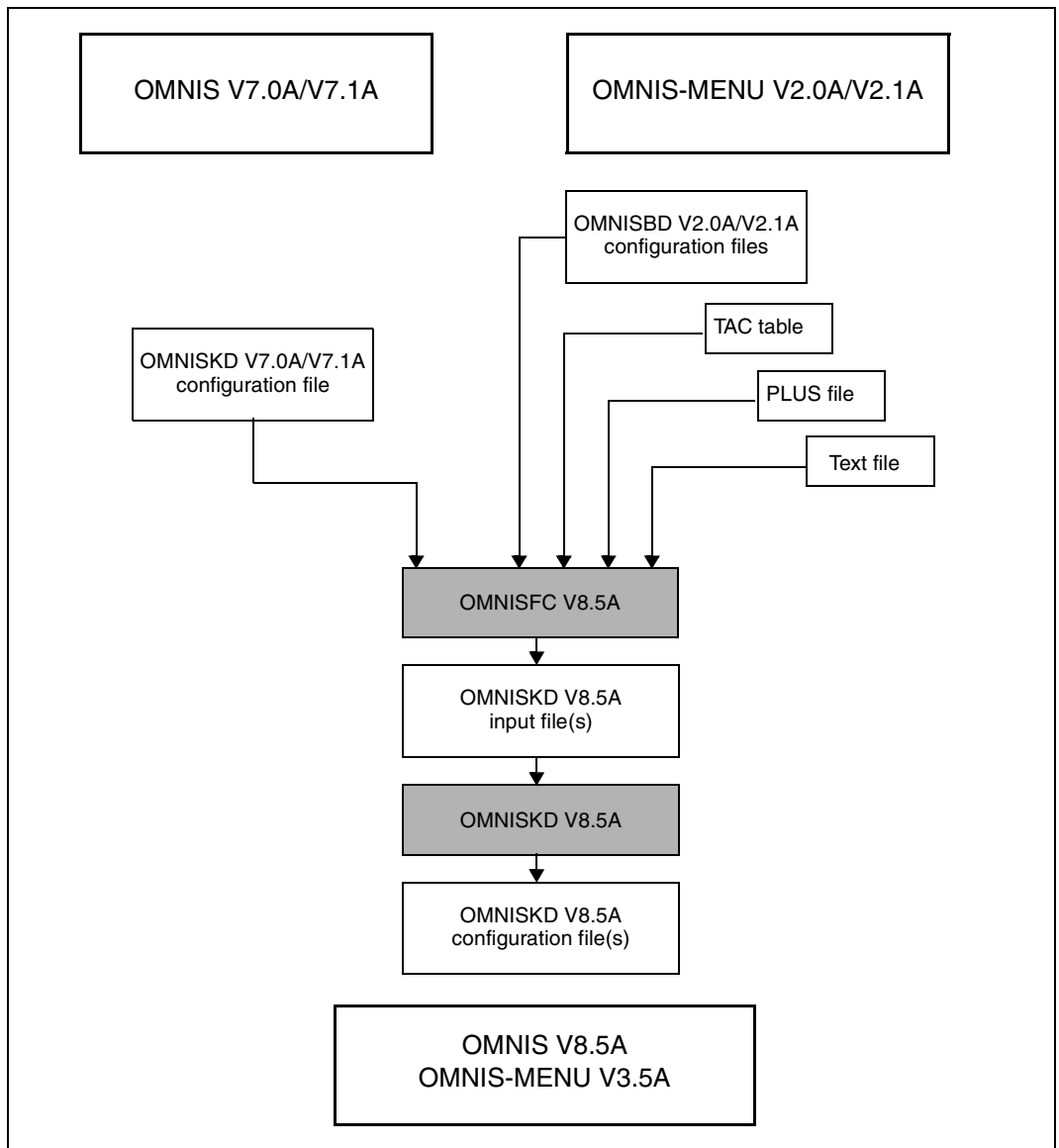


Figure 31: Diagram of the version change from (as of) V7.0A / V2.0A to V8.5A / V 3.5A

Version change from OMNIS-MENU ≤ V1.3A to V3.5A

Since OMNISFC V8.5A can only process OMNIS-MENU configuration files as of version V2.0A, the configuration files of any version ≤ V1.3A must first be converted to configuration files of Version V2.1A in an intermediate step.

The *SYSLNK.OMNIS.085* library still contains *OMNISBD.021* and *OMNISFC.021* to allow you to convert OMNIS-MENU configuration files from Version ≤ 1.3 to Version 2.1. You can start these tools as follows:

```
/START-PROGRAM FROM FILE=*M(SYSLNK.OMNIS.085,OMNISFC.021,RUN-MOD=ADV)
/START-PROGRAM FROM FILE=*M(SYSLNK.OMNIS.085,OMNISBD.021,RUN-MOD=ADV)
```

The rest of the procedure is exactly the same as for changing versions from V2.1A to V3.5A (see above).

MIGRATE procedure

The *MIGRATE* procedure, which can be found in the *SYSPRC.OMNIS-MENU.035* library, is available as an aid for changing versions.

12 OMNISLC: Editing the logging file

The utility routine OMNISLC can be used for offline editing of a logging file.

OMNISLC is called via the *START-PROGRAM* command or via *START-OMNISLC*.

The SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* must be enabled for this purpose.

12.1 Using OMNISLC

The logging file in which OMNIS logs all terminal inputs and outputs (when the logging function is active) is a PAM file with a relatively complicated internal data structure.

The OMNISLC (**OMNIS** Logging **C**onversion) utility has two functions:

1. It converts the logging file (or parts of it) to an easily readable form.
2. It converts the PAM file into an ISAM file, which is considerably easier to process by means of private analysis programs than the original PAM file.

The following figure gives an overview of the relationships:

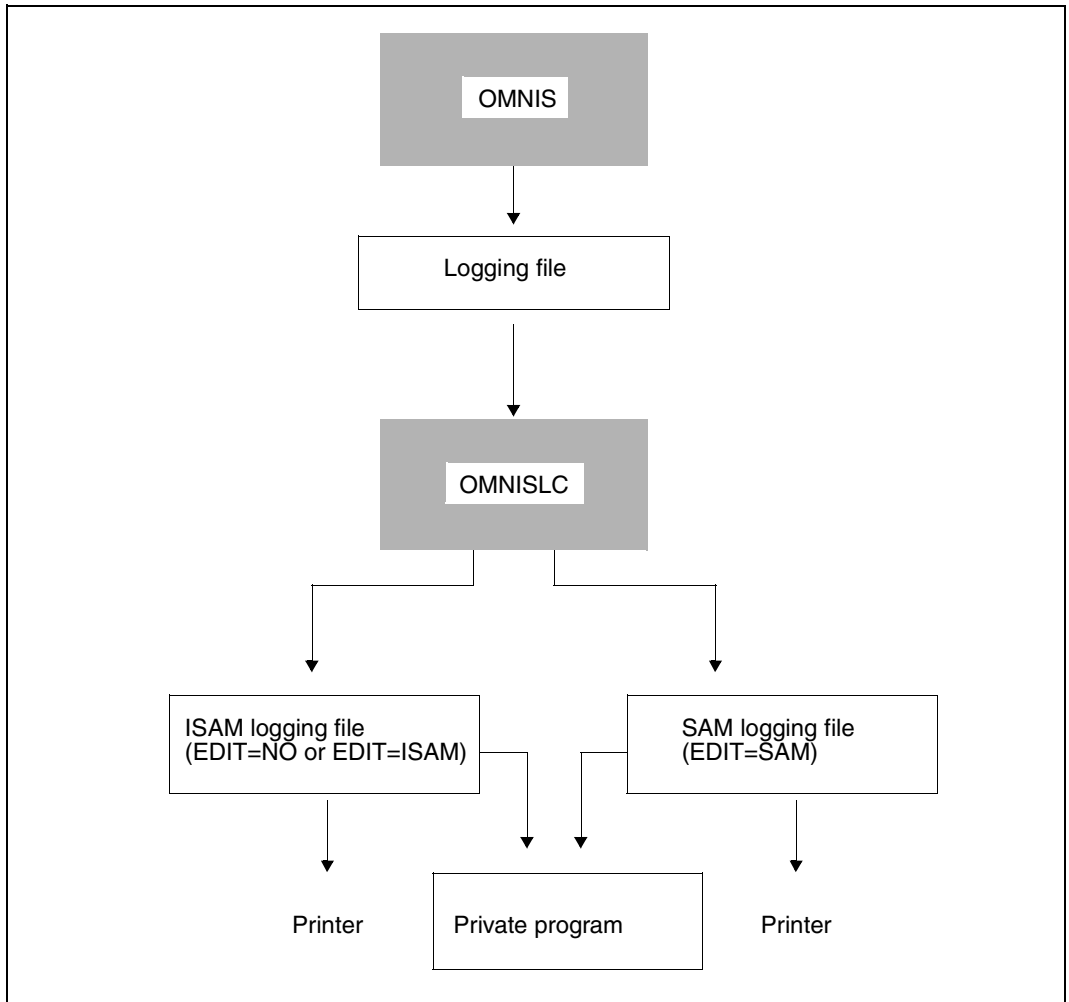


Figure 32: Editing the logging file with the aid of OMNISLC

Compatible versions of OMNIS and OMNISLC must be used.

Overview of statements

INPUT	defines the input file
OUTPUT	defines the output file
SELECT	selects the terminals, partners or users whose records are to be transferred from the input file to the output file and the logging parameters (<i>INPUT-LOGGING</i> , <i>OUTPUT-LOGGING</i>) that are to apply to the selected records
RUN	initiates conversion of the input file to the output file without terminating OMNISLC.
HELP	displays the syntax of OMNISLC commands at the terminal.
END	terminates the OMNISLC routine.

Order of statements

The *INPUT*, *OUTPUT* and *SELECT* statements must be entered before the *RUN* statement. *HELP* can be entered at any point.

Several conversions (each started with a *RUN*) can be carried out in the course of one session.

12.2 OMNISLC control statements

General syntax, command format

The notation used for description of the control statements is the same as that used for the OMNISKD statements. The statement format is also the same as that of the OMNISKD statements (see [page 109](#)).

It should be noted in particular that the statements can be indented as desired; this makes the input file easier to read.

END - Terminate OMNISLC

The *END* statement terminates an OMNISLC program run.

Operation	Operands
E[ND]	

HELP - Display command syntax

The *HELP* statement displays brief information about the syntax of OMNISLC statements on the terminal.

Operation	Operands
H[ELP]	

INPUT - Define input file

The *INPUT* statement defines the name of the logging file to be edited with OMNISLC.

Operation	Operands
I[INPUT]	input-filename

input-filename is the name of the logging file produced by OMNIS which is to be edited.

If no input file is defined using *INPUT*, then OMNISLC uses the file *OMNIS.LOGGING.001* as its input file.

Example

```
INPUT OMNIS.LOGGING.2014-03-18.001
```

OUTPUT - Defining the output file

The *OUTPUT* statement defines the name of the output file.

Operation	Operands
O[OUTPUT]	output-filename

output-filename specifies the name of a file to be used by OMNISLC for output.

If no output file is defined using *OUTPUT*, OMNISLC uses the file *input-filename.edit* as its output file, where *input-filename* refers to the file specified in the *INPUT* statement and *edit* stands for *NO*, *SAM* or *ISAM*, depending on what was specified in the *RUN* statement.

Example

```
OUTPUT OMNIS.LOGGING.2014-03-18.001.SAM
```

RUN - Starting conversion

The *RUN* statement initiates the editing of the logging file.

Operation	Operands
R[UN]	$[E[DIT]=\left\{\begin{array}{l} S[SAM] \\ I[SAM] \\ N[O] \end{array}\right\}]$ $[,L[CASE]=\left\{\begin{array}{l} Y[ES] \\ N[O] \end{array}\right\}]$

EDIT defines the type of editing.

SAM The input file is to be print-edited and converted into a SAM file. The format of the output is described on [page 245](#).

ISAM The input file is to be print-edited and converted into an ISAM file. The format of the output is described on [page 246](#).

NO The input file is to be converted to an ISAM file and unblocked in the process. The format of the output is described on [page 244](#).

LCASE defines conversion from lowercase to uppercase letters. *LCASE* is interpreted only with *EDIT=SAM* or *EDIT=ISAM*.

YES Lowercase letters are not to be converted to upper case.

NO Lowercase letters are to be converted to upper case.

The input file, output file and record selection must be defined using the commands *INPUT*, *OUTPUT* and *SELECT*, respectively, before the *RUN* statement is entered. If the definitions are omitted, the following defaults apply:

```
INPUT=  OMNIS.LOGGING.001
OUTPUT= OMNIS.LOGGING.001.edit
SELECT= INPUT-LOGGING=STD
SELECT=  OUTPUT-LOGGING=STD
```

All the records of the logging file involved are output.

Examples

```
RUN
RUN EDIT=ISAM,LCASE=NO
RUN EDIT=NO
```


SELECT - Select partners/terminals

The *SELECT* statement defines which parts of the input file are to be transferred to the output file.

All outputs from the *SELECT* statement are carried out as OR-connection.

If the entire log is to be output, no *SELECT* statement is required (default).

Operation	Operands
S[ELECT]	$[I[INPUT]-L[OGGING]=\left\{\begin{array}{l} SE[ND] \\ B[OTH] \\ R[EC] \\ S[TD] \end{array}\right\}]$ $[,O[UTPUT]-L[OGGING]=\left\{\begin{array}{l} SE[ND] \\ B[OTH] \\ R[EC] \\ S[TD] \end{array}\right\}]$ [,MSG= (number1[,number2]...[,number10]) [,PAC= pac] $[,P[ID]=\left\{\begin{array}{l} num \\ (num,...) \end{array}\right\}]$ [,TE[RMINAL]= ([ptnname][,praname]) $[,T[ID]=\left\{\begin{array}{l} num \\ (num,...) \end{array}\right\}]$ [,U[SER]= user]

INPUT-LOGGING	specifies which logging file messages that were input at the terminal are to be analyzed.
SEND	The messages that were written to the logging file before being sent to the partner are edited.
BOTH	The messages that were written to the logging file both when they were entered at the terminal and when they were sent to the partner are edited.
REC	The messages that were written to the logging file after being entered at the terminal are edited.
STD	The messages which were written to the logging file before being sent to the partner are edited.
OUTPUT-LOGGING	specifies which of the logging file messages sent by the partner are to be analyzed.
SEND	The messages that were written to the logging file before being sent to the terminal are edited.
BOTH	The messages that were written to the logging file both when they were sent to the terminal and when they were received from the partner are edited.
REC	The messages that were written to the logging file after being received from the partner are edited.
STD	The messages which were written to the logging file before being sent to the terminal are edited.
MSG	specifies the OMNIS message number of messages which are to be prepared by OMNISLC and written to the output file. All other operands are correspondingly restricted by an AND operation with this operand.
number	Up to ten 7-character OMNIS message numbers can be specified.
PAC	specifies the PAC of the partner whose records are to be selected.
PID	specifies a list of PIDs. All records with one of the specified PIDs are to be transferred to the output file. The partner identifier PID is a sequential number which is assigned to the partner after a successful connection setup.
TERMINAL	is the network address of the terminal whose records are to be selected.
ptnname	is the partner name of the terminal.
proname	is the processor name of the terminal.

TID	specifies a list of TIDs. All records with one of the specified TIDs are to be transferred to the output file. The terminal identifier TID is a sequential number which is assigned to the terminal when it signs on to OMNIS.
USER	specifies a user. All records of the specified user are to be included in the output file.
user	specifies the user name.

If only the OMNIS commands or OMNIS messages, or both, are to be selected, then one of the following *SELECT* statements must be entered:

```
SELECT PID=1           "Messages only"
SELECT PID=7           "Commands only"
SELECT PID=(1,7)       "Commands and messages"
SELECT PID=(1,2,3,4,5,6) "All OMNIS outputs with @:"
```

Examples

```
SELECT PID=8
SELECT TID=(7,8)
SELECT PID=(8,13,57,67),TID=(17,33)
```

12.3 Format of the output

Depending on the type of editing specified in the *RUN* statement, the output may be in one of three different formats:

12.3.1 Output when EDIT=NO

When *EDIT=NO*, the logging file is converted one-to-one into an ISAM file; i.e. each logical logging record (not each PAM block) becomes an ISAM record.

The ISAM file is a standard file (*RECFORM=V*, *KEYPOS=5*, *KEYLEN=8*).

The ISAM key contains the logical record address from the logging file:

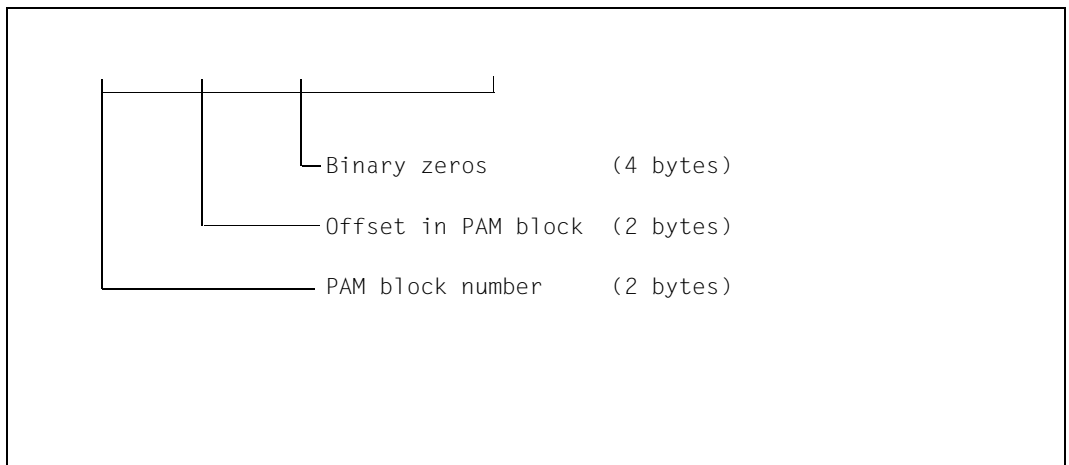


Figure 33: ISAM key structure of a logging file with EDIT=NO

The data field contains the logging record (see [page 30](#)), including the management data. This management data includes chaining addresses for direct positioning within the ISAM file (as the logical PAM record addresses match the ISAM key).

The ISAM file cannot generally be manipulated using an editor program, as it contains non-numeric keys and records longer than 256 bytes. It can, however, be analyzed by private programs (so as to avoid complicated PAM access to the original logging file).

12.3.2 Output when EDIT=SAM

When *EDIT=SAM*, the logging records are written to a SAM file in print-edited form.

Each line contains a prefix and a text portion.

The prefix has the following format:

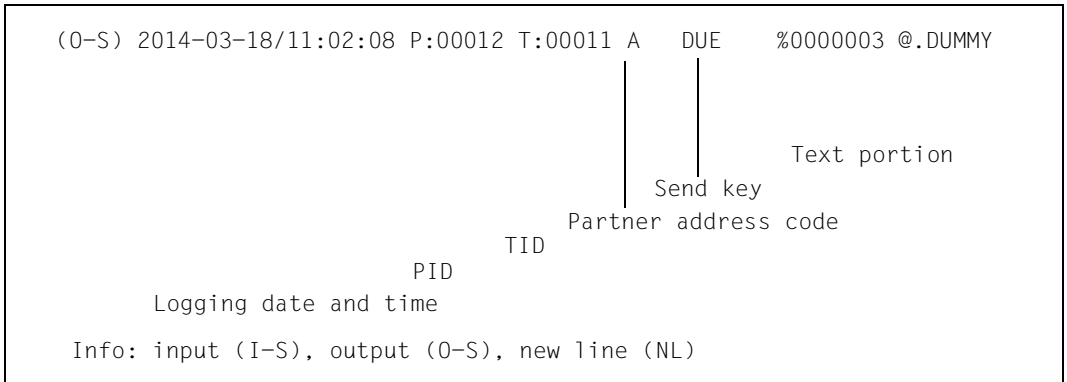


Figure 34: Line prefix in the logging file with EDIT=SAM

The text portion contains the actual data. These are edited just as they would normally be displayed at the terminal.

Line messages are split at the control codes "New Line" and "New Page", or after 80 characters if neither of these control codes is found. All other control codes are ignored.

For **format messages**, the physical terminal control characters are interpreted (positioning sequences, ESC sequences); all non-positioning control characters are ignored. The result is a 24x80 output matrix ("virtual screen surface").

Example of an output

```
(O-S) 2014-03-18/11:09:01 P:00001 T:00011 @ DUE  OMS0001 USER IS CONNECTED TO OMNIS (TERMINAL=MN303028/V214H303, TID=0014)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(NL)
(O-S) 2014-03-18/11:09:01 P:00001 T:00011 @   OMS0002 OMNIS VERSION 8.5A00 READY.
(O-S) 2014-03-18/11:09:01 P:00001 T:00011 @   /
(I-S) 2014-03-18/11:12:05 P:00012 T:00011 A DUE  SHOW-FILE-ATTRIBUTES
(O-S) 2014-03-18/11:12:08 P:00012 T:00011 A   %0000003 @.DUMMY
(NL)
(NL)
(NL)
(NL)
(NL)
```

12.3.3 Output when EDIT=ISAM

When *EDIT=ISAM*, the logging file is converted (in print-edited form) into an ISAM file, i.e. (as with *EDIT=SAM*) the logging data is "wrapped".

This ISAM file can be processed with an editor. The key fields are in standard editor format (*RECFORM=V*, *KEYPOS=5*, *KEYLEN=8*, zoned decimal). ISAM format is an advantage particularly with large logging files, since the user works directly on disk (no overflow of virtual memory).

Structure of the ISAM key:

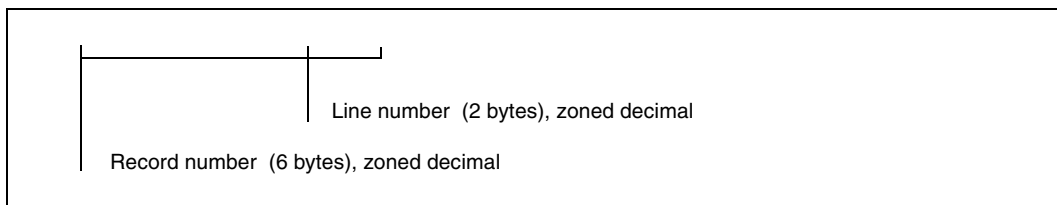


Figure 35: ISAM key structure of a logging file with EDIT=ISAM

The line number contains the logical logging record number. The line number is incremented if a logging record is split up into several ISAM records.

The prefix is written to an ISAM record of its own (line number 00); the data follows in the subsequent lines (line numbers 01 through 99).

Sample output

(For the clarity, the ISAM key has been separated from the data with a space).

```
00000100 (O-S) 2014-03-18/11:09:01 P:00001 T:00011 @ DUE
00000101 OMS0001 USER IS CONNECTED TO (TERMINAL=MN303028/V214H313, TID=0014)
00000200 (O-S) 2014-03-18/11:09:01 P:00001 T:00011 @
00000201 *****
00000202 * *
00000203 * O M N I S V 8.5A Test version (for user ID $OMNIS) *
00000204 * *
00000205 * You are working with the latest version of OMNIS; the normal *
00000206 * OMNIS is loaded under the name NULLIS. *
00000207 * *
00000208 * ... *
00000209 * ... *
00000210 * *
00000211 *****
00000300 (O-S) 2014-03-18/11:09:01 P:00001 T:00011 @
00000301 OMS0002 OMNIS VERSION 8.5A00.00 READY.
00000400 (O-S) 2014-03-18/11:09:01 P:00001 T:00011 @
00000401 /
00005600 (I-S) 2014-03-18/11:12:35 P:00012 T:00011 A DUE
00005601 START-PROGRAM FROM-FILE=$IFG.IFG
...
```

13 OMNISTC: Converting text files

You can specify the number of relevant bytes for the key of the text file (with start parameter *LENGTH*). With the utility OMNISTC text files can be converted to different key lengths.

The utility OMNISTC reads an ISAM file with key length 8 as its input file and generates a converted ISAM file with key length 8.

You can call OMNISTC via the *START-PROGRAM* command or via *START-OMNISTC*, provided that the SDF syntax file *SYSSDF.OMNIS.085* or *SYSSDF.OMNIS.085.USER* has been enabled.

Input file (key length 8)	OMNISTC	Output file (key length 8)
LENGTH=4		LENGTH=6
0001.0001/LOGON.....	—————>	000001.17/LOGON.....
0001.0005/REMARK.....	—————>	000001.50/REMARK.....
0001.0007/LOGOFF.....	—————>	000001.83/LOGOFF.....
0010.0006/LOGON.....	—————>	000010.17/LOGON.....
0010.0008/COPY-FILE...	—————>	000010.50/COPY-FILE...
0010.0999/CREATE-JV...	—————>	000010.83/CREATE-JV...
.		.
.		.
0110.0999/LOGOFF.....	—————>	000110.96/LOGOFF.....

INPUT ..., *LENGTH=* identifies the relevant key of the input file, and *OUTPUT ...*, *LENGTH=* identifies the relevant key of the output file. If the specified relevant length of the key for the input file is greater than the specified relevant length of the key for the output file, OMNISTC will terminate with a corresponding error message.

The OMNISTC program can only process text files with a maximum of $10^{(8 - \text{LENGTH of the output file})}$ statements per sequence.

Using OMNISTC

The program is called with the statement *START-OMNISTC*.

OMNISTC responds with the following prompt:

```
PLEASE ENTER COMMAND OR <HELP>
```

```
*
```

The subsequent statements are read in via SYSDTA. Since no default values are defined, all operands must be specified. If an invalid operand is entered, an error message is issued, and the program asks for the input to be repeated.

OMNISTC commands

I[NPUT] filename, L[ENGTH]= 4..7	Name of the input file and length of the relevant key in this file
O[UTPUT] filename, L[ENGTH]= 4..7	Name of the output file and length of the relevant key in this file
R[UN]	Start of conversion
H[ELP]	Output of the help text
E[ND]	Termination of OMNISTC

14 Programming with OMNIS

The user interface (exit interface) described in this section is used to extend the functional scope of OMNIS.

The exit interface

- decouples exit routines from OMNIS
- supports both Assembler and COBOL
- provides functions for the output/suppression of messages on hardcopy units
- offers extended operating facilities for UTM partners to which multiplex connections exist

14.1 Description

In the basic structure (without exit routines), OMNIS receives all messages centrally and determines sender and addressee of the message on the basis of its table entries.

Messages coming from partners are forwarded to the terminal communicating with the partner and, if desired, are also output to hardcopy units. Terminal inputs are passed on to the partner addressed by the partner address code. It is also possible to log all messages sent via OMNIS/received from a partner/input at a terminal in a logging file. Optionally, the messages can also be logged in a logging file. This logging function can be selected for

- all messages passed via OMNIS
- the messages received from a partner
- the messages entered at a terminal

When an exit routine is loaded and assigned, OMNIS no longer deals with messages, commands and system messages. Instead, they are passed on to the exit routine in special parameter areas together with an event-specific operation code. This also applies to messages and system messages that are to be logged in the logging file or output on a hardcopy unit. The parameter areas contain detailed information on the terminals and partners involved (types, partner names, etc.).

The exit routine can

- ignore the event (message/system message/OMNIS command)
- cause OMNIS to take further action:

For this to be possible, the exit routine must pass an event-specific operation code to OMNIS, and the appropriate parameter areas must be supplied with values.

The following actions are possible:

- execute an OMNIS command,
- output an OMNIS system message on a terminal,
- send a message to a partner or a terminal,
- generate a new message from an existing message,
- modify or block a message (this also applies to messages and system messages which are to be logged in the logging file or output on a hardcopy unit),
- specify a timer interval for cyclic calling of the exit routine,
- set up a connection to a terminal,
- define settings for a terminal/partner.

All exit routines must have XS capability. If an exit routine programmed only for the lower address space is to be used, the non-XS-capable phase of OMNIS (OMNIS.NXS) must be used.

Short form of the starting command for the non-XS-capable OMNIS:

```
/START-PROGRAM *M(SYSLNK.OMNIS.085,OMNIS, RU=A), PROG-MODE=24
```

If the XHCS functionality is to be used in the exit routines, these exit routines must be converted from the edit options to the VTSUCB.

Example

Two terminals have connections with four applications (\$DIALOG, UTMAPP1, UTMAPP2, DCAMAPP).

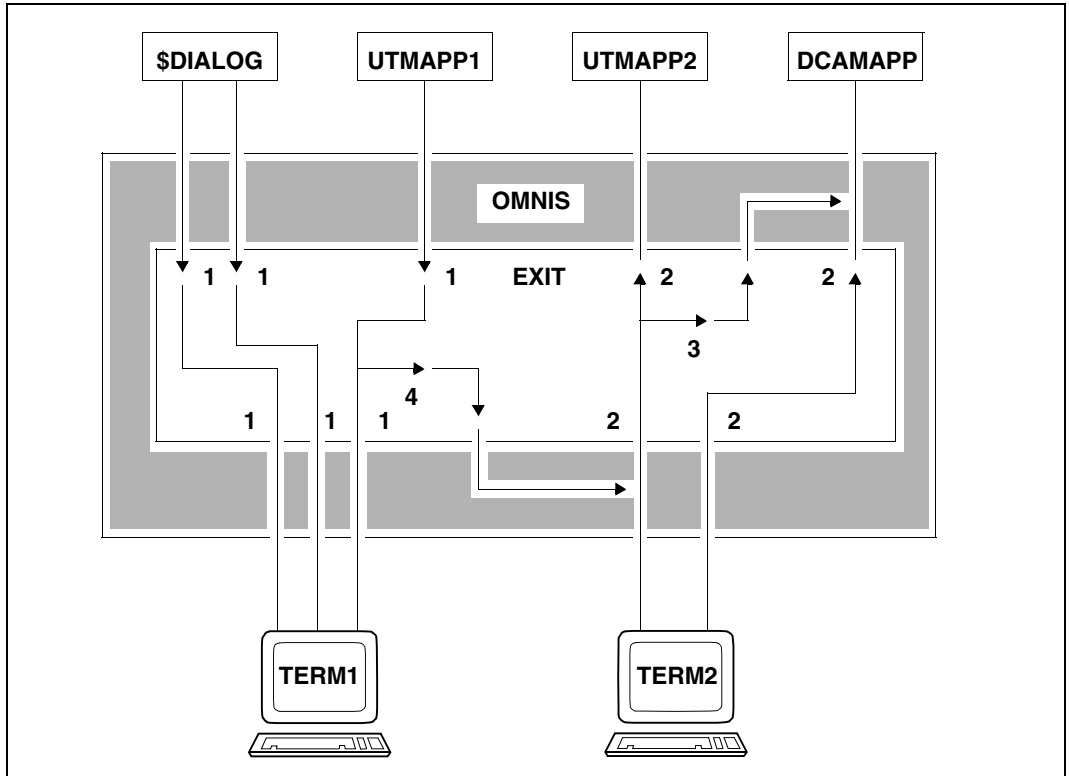


Figure 36: Effects of the operation codes in an exit routine

The operation codes in this example have the following meaning:

- 1 OMEMPT Message from partner to terminal
- 2 OMEMTP Message from terminal to partner
- 3 OMESLPA Select partner
- 4 OMESLTA Select terminal

If the exit routine is called with, for example, the event "message from terminal to partner" and a UTM application is entered as the partner, the exit routine must call OMNIS explicitly with the operation code "message from terminal to partner" in order to be able to pass on the message to the UTM application. If it fails to do so, the message is suppressed.

This allows the exit routine to take full control of message traffic between terminals and partners (or between terminal and OMNIS in the case of commands and system messages).

Once the exit routine has been activated, it can call OMNIS any number of times. It can copy, suppress and divert messages; it can conduct dialogs with partners without these being visible at the terminal; it can communicate with the terminal via messages in line or formatted mode or via physical messages.

In COBOL, an exit routine is terminated with *EXIT*; in Assembler, by a return to the address passed in register 14.

Several exit routines can be active at one time. In this case, the user determines the order in which the messages are sent to the exit routines concerned.

The figure below shows a configuration with three exit routines and the possible output sequence of a partner's message to a terminal.

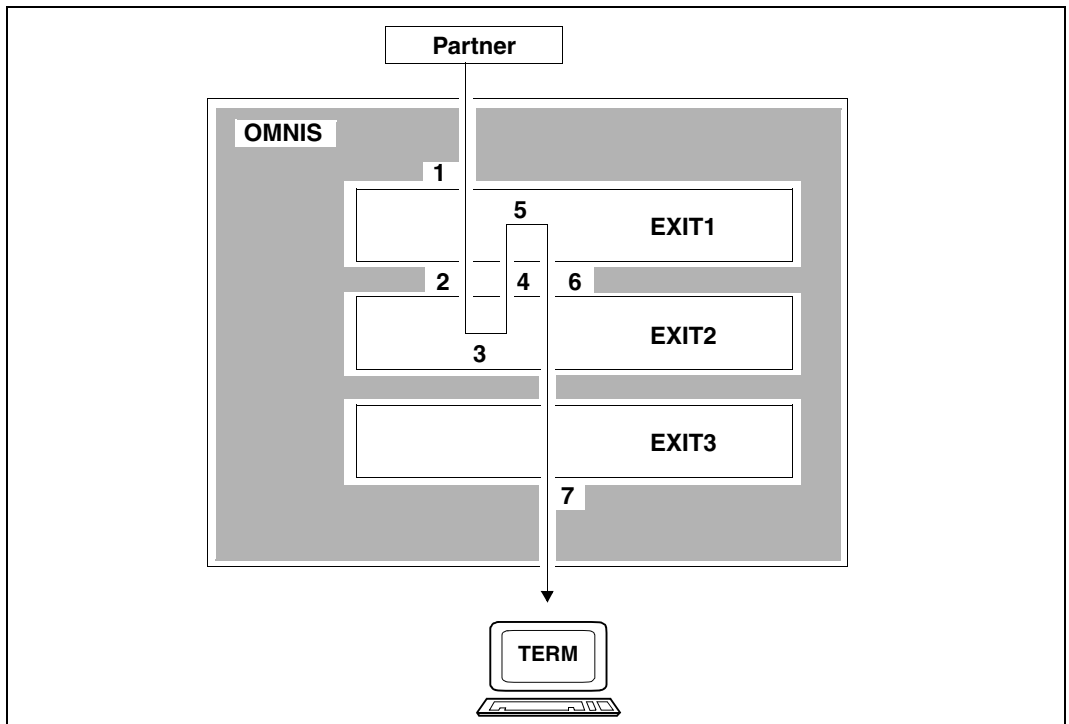


Figure 37: Configuration with three exit routines

Definition of the exit routine:

```
EXIT EX01, ACTION=CREATE, MODULES=(EXIT1, EXIT2, EXIT3)
```

Exit1 is closest to the partner, while Exit3 is closest to the terminal. Messages of the partner are sent first to the exit closest to the partner and last to the exit closest to the terminal. Terminal messages are first sent to the exit closest to the terminal, and sent last to the exit closest to the partner.

Please note that a message is sent to a further exit routine only on the basis of an activity on the part of the previous exit routine.

The following has taken place in the above example:

1. A message was received from a partner and delivered to the exit routine Exit1 closest to the partner.
2. Exit routine Exit1 has passed on the message (either modified or not) to the terminal.
3. The next exit routine on the path to the terminal is Exit2. The message has therefore been passed to it.
4. Exit2 has now returned a message to the partner. The original message has not been passed on to the terminal and is therefore neither delivered to Exit3 nor output at the terminal.
5. Exit1 is the next exit routine on the path from Exit2 to the partner. The reply generated by Exit2 is therefore passed to Exit1.
6. Exit1 does not pass on the message received from Exit2 to the partner but, in turn, outputs a message on the terminal.
7. This message is passed on by both Exit2 and Exit3, and is therefore output to the terminal.

Possible interpretation of the above events:

Exit1 performs an authorization check, Exit2 is a programmed operator and Exit3 is a terminal control facility which administers the screen in a special manner.

Serial exit routines make it possible to distribute the overall exit function amongst a group of up to 14 logically independent routines, known as an exit group.

Imbedding exit routines in OMNIS

The following will provide you with an overview of the way in which OMNIS and exit routines function and the interplay between the two resulting from the way in which the program has been implemented.

In particular, the processing sequences involved in message traffic, logging and the output of messages on hardcopy units using exit routines are described.

The following applies in general to message traffic using exit routines:

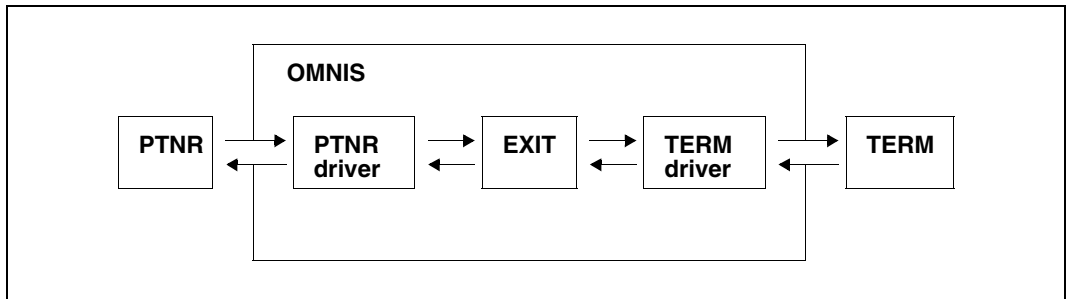


Figure 38: Implementation of message traffic via exit routines

First of all, the message from a partner is analyzed by the partner driver and is then passed on to the exit routine. In the following, the term "partner driver" refers to internal OMNIS routines which take the different communication characteristics of the partners into account and thus pass the messages to the exit routines in such a way that they can be processed regardless of the partner type. If a message is to be sent to a partner, distribution is also accomplished using the partner driver.

Inputs to or from a terminal are implemented in a similar way using "terminal drivers". These are internal OMNIS routines that take the different types of terminals into account.

The following applies when commands and system messages are taken into consideration:

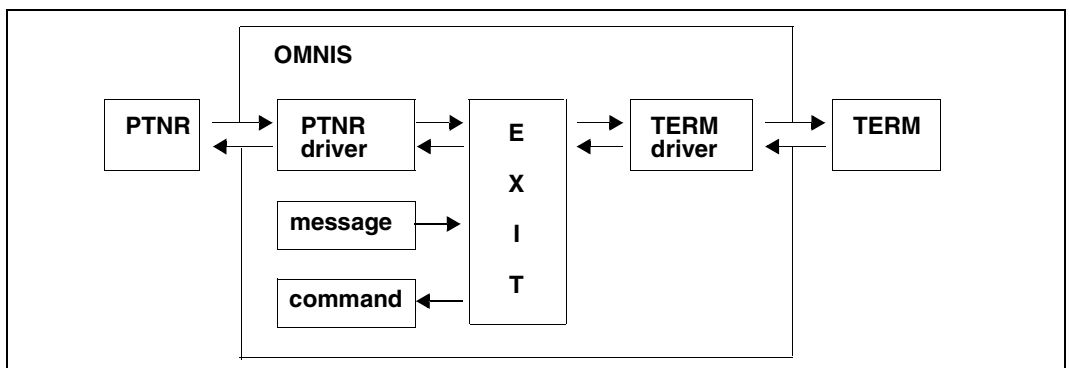


Figure 39: Processing of messages/system messages/commands with exit routines

Messages are sent from a partner to the partner driver (or from the terminal to the terminal driver) for preprocessing and are passed on from there to the exit routine. OMNIS passes on system messages to the exit routine directly. Commands from a terminal are passed to the exit routine via the terminal driver. The exit routine can then initiate the OMNIS command.

Logging using exit routines has been implemented as follows:

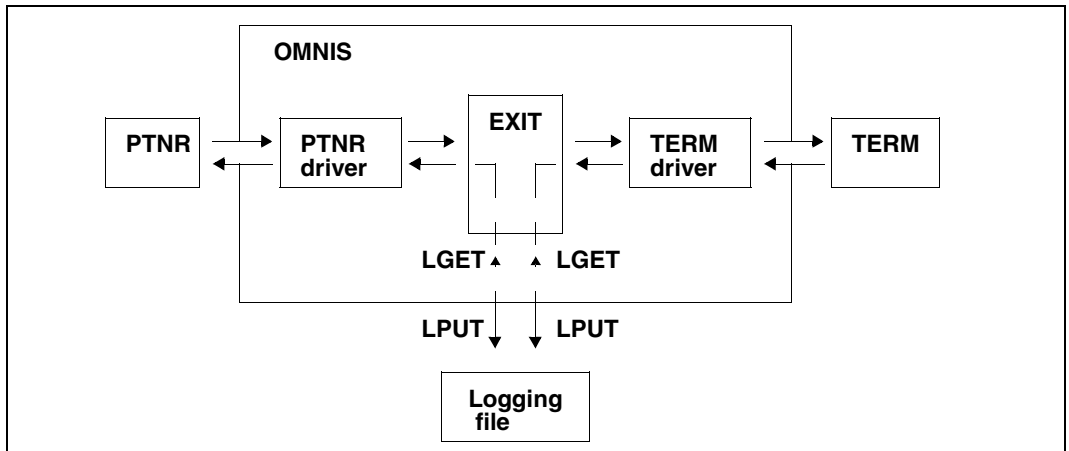


Figure 40: Implementation of logging via exit routines (LOGGING=YES)

If the logging of messages passing via OMNIS in a logging file is desired (*LOGGING=Y*), the messages sent by a partner are passed to the terminal driver via the partner driver and the exit routine. The logging file *OMELPUT* (write) or *OMELGET* (read) is accessed from there (see [page 261](#)). Messages from a terminal are passed on to the partner driver via the terminal driver and the exit routine. The partner driver then initiates access to the logging file *OMELPUT* or *OMELGET*.

The various cases of *INPUT-LOGGING* and *OUTPUT-LOGGING* (*RECEIVE/SEND/BOTH*) can be represented in a similar way.

The output of messages on a hardcopy unit is implemented as follows: :

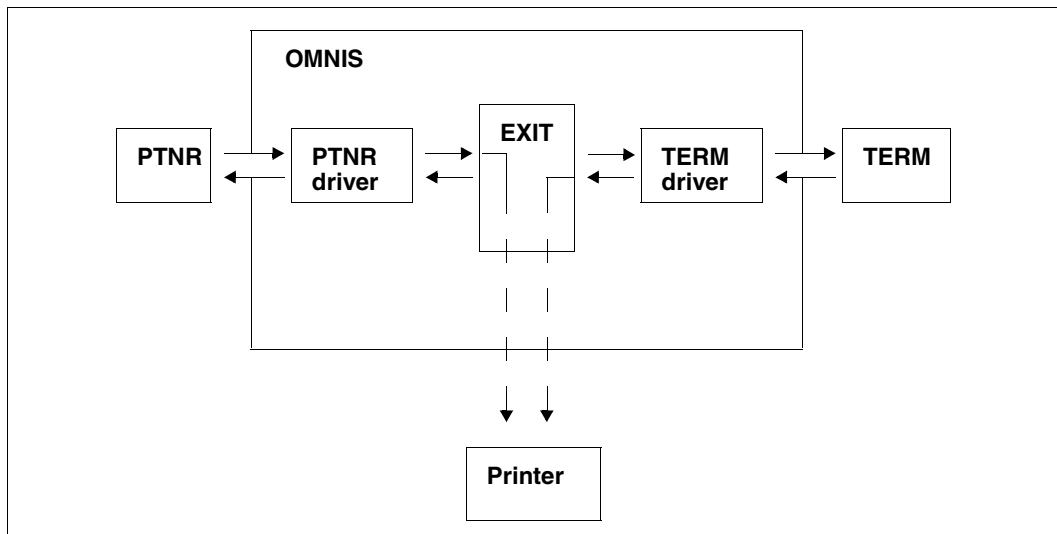


Figure 41: Output of messages on a hardcopy unit with exit routines

The message is passed from the partner (via the partner driver and the exit routine) or from the terminal to the terminal driver, which initiates the appropriate hardcopy output.

14.2 Data areas

OMNIS passes to the exit routine a list of five addresses that are used to address the five-part parameter area *OMEPAR*, the communication area, the terminal work area, the partner work area and the VTSUCB area:

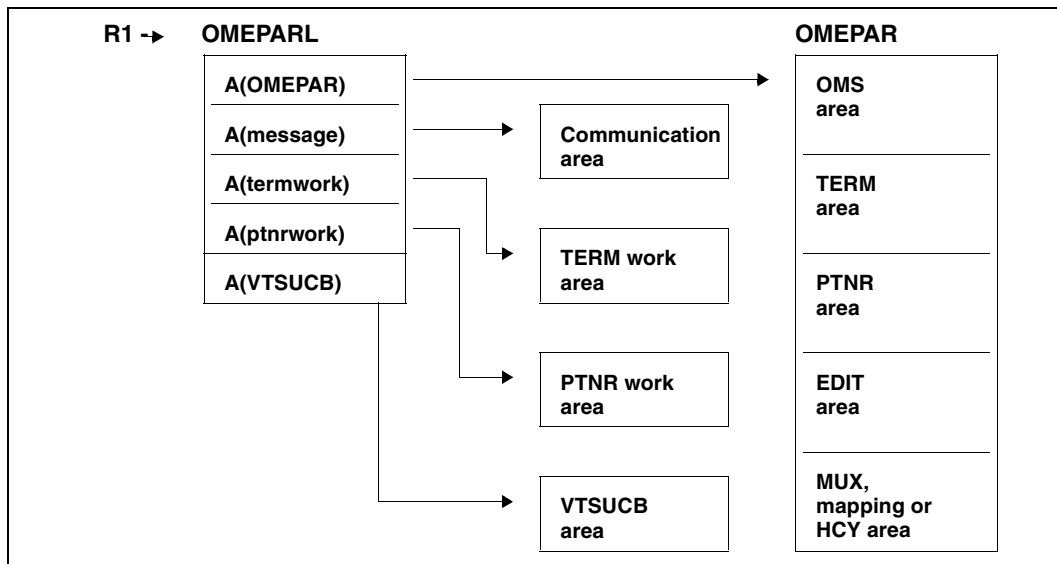


Figure 42: Parameter areas and address pointers for calling exit routines

When OMNIS is called from an exit routine, it must transfer these five parameter areas (modified or not) in accordance with the same conventions.

Parameter areas

The parameter area addressed via *OMEPAR* is divided into five sub-areas: the OMS area, the TERM area, the PTNR area, the EDIT area, and the MUX or mapping or HCY area (alternatives).

OMS area	The OMNIS area contains data required for communication between the exit routine and OMNIS, e.g. operation codes and return codes.
TERM area	The TERM area contains terminal-specific information that is administered by OMNIS, e.g. data from <i>DECLARE-TERMINAL</i> and additional information.
PTNR area	The PTNR area contains partner-specific information that is administered by OMNIS, e.g. data from <i>DECLARE-PARTNER</i> or <i>OPNCON</i> and additional information.
EDIT area	The EDIT area contains the EDIT options, which describe the message format.
MUX area	The MUX area contains information which is exchanged between UTM and OMNIS. This area is only relevant to UTM partners with <i>ROUTE=MUX</i> and the appropriate generation of UTM and OMNIS.

The HCY area can be transferred (under the control of OMNIS) as an alternative to the MUX area:

HCY area	The HCY area contains information concerning the hardcopy printer.
-----------------	--

In the case of an operation code for hardcopy, the HCY area and not the UTM area is valid. Additionally, as an alternative to the MUX area, the mapping area can also be transferred:

Mapping area

The mapping area contains OMNIS messages in mapping format, as supplied by the *MSG7* macro.

In the case of an operation code for OMNIS messages, the mapping area and not the UTM area is valid.

Communication area

The communication area contains the message as received from the partner/ terminal, or as it is to be sent to the partner/terminal. The exit routine may modify this area as required.

The length of the message is noted in the field *OMEOMLEN*. The maximum length of the communication area is defined by means of the operand *IOAREA* in the start parameter *LENGTH*.

TERM work area

The TERM work area is created by OMNIS during connection setup from or to a terminal and is available to the exit routine as a terminal-specific storage area. The TERM work area remains in existence until connection cleardown with *HOLD=NO*. The area is initialized with binary zeros.

OMNIS administers a work area for each terminal. The length of this work area is defined with the operand *TWORK* of the start parameter *LENGTH*. The exit routine can use this area for any desired purpose.



The TERM work area only remains permanently assigned to a terminal as long as a connection to this terminal exists. When a connection is set up (even if this setup has a negative outcome) a new freshly initialized work area is always created.

When OMNIS-MENU is used, the first 250 bytes of the TERM work area are reserved for OMNIS-MENU.

PTNR work area

The PTNR work area is created by OMNIS during connection setup to the partner and is available to the exit routine as a partner-specific storage area. The area remains in existence until connection cleardown and is initialized with binary zeros.

OMNIS administers a work area for each partner. The length of this work area is defined with the operand *PWORK* of the start parameter *LENGTH*. The exit routine can use this area for any desired purpose.



The PTNR work area only remains permanently assigned to a partner as long as a connection to this partner exists. When a connection is set up (even if this setup has a negative outcome) a new freshly initialized work area is always created.

VTSUCB area

The VTSUCB area is created by OMNIS. Its structure is the same as that of the VTSUCB provided by TIAM. The Assembler macro and the COBOL copy element for the VTSUCB are supplied with VTSU-B.

If there is a VTSUCB, it is transferred to the VTSUCB area and is available to the exit routine. At the same time, it is mapped onto the EDIT area in the OMNIS parameter area. Since OMNIS always uses VTSUCB for traffic directed towards the terminal, information from the EDIT options may be lost.

As all new functions will be supported only with VTSUCB, the VTSUCB should always be used for new exit routines.

OMNIS supports VTSUCB-specific functions only if the OMEEVTCB field is set to *Y*.



The communication area, TERM work area and PTNR work area are unstructured and can be used by the user. Their lengths are governed by the start parameter *LENGTH*.

14.3 Operation codes

In the OMS area, the operation codes offering various functions are contained in the field OMEOOPC. Basically, we distinguish between two possible implementations:

- OMNIS calls exit routine
- Exit routine calls OMNIS

Operation codes which are not analyzed in 'OMNIS calls exit routine' must be passed on.

14.3.1 OMNIS calls exit routine

OMNIS calls the exit routine with an event-specific operation code and passes information to it in the various parameter areas.

Data transfer in the parameter areas

When the exit routine is called by OMNIS, different fields in the parameter areas are supplied with values, depending on the operation code.

Opcode	OMS	TERM	PTNR	EDIT	MUX	HCY	Mapping	COMM	TWORK	PWORK	VTSUCB
OMEINIT	X							X			X
OMETIMC	X							X			X
OMECOM	X	X		X				X	X		X
OMEMES	X	X		X			X	X	X		X
OMEMPT	X	X	X	X	○			X	X	X	X
OMEMTP	X	X	X	X	○			X	X	X	X
OMELGET	X	X	X	X	○			X	X	X	X
OMELPUT	X	X	X	X	○			X	X	X	X
OMESHCY	X			X		X		X			X
OMEEND	X										

X . . . OMNIS places data in the area
 ○ . . . Area is used by OMNIS only for UTM partners with a multiplex connection.

Table 1: Parameter areas containing data when OMNIS calls the exit routine

The indicated areas are all filled with data by OMNIS.

Overview of events

OMNIS calls the exit routine when certain events occur. The entry in the *OMEEOPC* field indicates which event led to the current call:

OMEINIT	The exit routine was called (immediately after loading) in order to execute its own initialization. Apart from the OMS area, no other parameter areas are supplied with data. The communication area is present and can be used for further 'exit routine calls OMNIS' calls. If necessary, the user must set terminal and partner him/herself.
OMETIMC	The exit routine was called after timeout of the timer interval <i>OMETIME</i> . Apart from the OMS area, no other parameter areas are supplied with data. The communication area is present and can be used for further 'exit routine calls OMNIS' calls. If necessary, the user must set terminal and partner him/herself.
OMECOM	A command to OMNIS was entered. The communication area contains the OMNIS command and the TERM area contains the terminal on which the command was entered.
OMEMES	An OMNIS message is output on a terminal. The options <i>MODE</i> and <i>REPLY</i> can be selectively specified by means of the EDIT options. For each subsequent call, the most recently specified EDIT options are used unless they are explicitly modified. In the case of <i>OMEMAPPL>0</i> , the OMNIS message exists in mapping format. The communication area contains the OMNIS message and the TERM area contains the terminal to which the message is to be output.
OMEMPT	A message is sent from a partner to a terminal. The data from the MUX is available if the partner is a MUX partner which supports the PUTMUX protocol. The communication area contains the message which is to be sent to the terminal.
OMEMTP	A message is sent from a terminal to a partner. The data from the MUX is available if the partner is a MUX partner which supports the PUTMUX protocol. The communication area contains the message which is to be sent to the partner.
OMELGET	A message was read from the logging file. The message is contained in the communication area.
OMELPUT	A message is written to the logging file. The message is contained in the communication area.
OMESHCY	A message is output to a hardcopy unit. The message is contained in the communication area.

OMEEND OMNIS is being terminated. The exit routine can still perform various tasks (e.g. saving and closing files). OMNIS calls are no longer permitted, however. Apart from the OMS area, no other parameter areas are supplied with data.

OMELGET and OMELPUT are called even if logging is deactivated.

14.3.2 Exit routine calls OMNIS

If an exit routine needs to initiate further actions by OMNIS, it must pass an action-specific operation code to OMNIS. In addition, various parameter areas must contain data. The following sections describe the possible calls, the parameter areas which must contain data and the OMNIS return information.

Data in the parameter areas

If OMNIS is called by the exit routine, certain fields described in the following must contain data. The work areas are not modified by OMNIS.

Opcode	OMS	TERM	PTNR	EDIT/ VTSUCB	MUX	HCY	Mapping	COMM
OMECOM	X	X			O			X
OMEMES	X	X		X			X	X
OMEMTP	X	X	X	X	O			X
OMEMPT	X	X	X	X	O			X
OMELGET	X	X	X		O			X
OMELPUT	X	X	X					X
OMESHCY	X					X		X
OMESLTA	X	X						
OMESLTF	X							
OMESLTN	X	X						
OMESLPA	X	X	X					
OMESLPP	X	X	X					
OMESLPPF	X	X						
OMESLPN	X	X	X					
OMESVPA	X	X	X					X
OMESTXPA	X	X	X					X
OMESHDR	X	X	X					X
OMETIMS	X							
OMEOPTRM	X	X						

X . . . Area must contain data when OMNIS is called.

O . . . Area is evaluated by UTM applications if there is a multiplex connection.

Table 2: Data in the parameter areas when the exit routine calls OMNIS


Overview of actions

When OMNIS is called, the field *OMELOOPC* must contain an operation code indicating which action is to be carried out:

OMECOM	A command is to be issued to OMNIS. The communication area contains the OMNIS command. In the OMS area, the field <i>OMEOMLEN</i> must be supplied with the length of the OMNIS command contained in the communication area.
OMEMES	An OMNIS message is to be output at a terminal in line mode. In the OMS area, the field <i>OMEOMLEN</i> must be supplied with the length of the OMNIS message contained in the communication area. The OMNIS message is sent to the terminal (in accordance with the entries in the TERM area).
OMEMTP	A message is to be sent to the current partner. In the OMS area, the <i>OMEOMLEN</i> field must be supplied with the length of the message contained in the communication area. The message is sent to the partner (in accordance with the entries in the PTNR area).
OMEMPT	A message is to be sent to the current terminal. In the OMS area, the field <i>OMEOMLEN</i> must be supplied with the length of the message contained in the communication area. The message is sent to the terminal (in accordance with the entries in the TERM area).
OMELGET	A message from the logging file is to be sent to a terminal or to a partner.
OMELPUT	A message is to be written to the logging file.
OMESHCY	A message is to be output to a hardcopy unit.
OMESLTA	A certain terminal is to be set. To this end, the fields <i>OMETNAME</i> and <i>OMETPROC</i> in the TERM area must be supplied with the partner name and the processor name.
OMESLTF	The first terminal in the internal OMNIS terminal list is to be set. In the OMS area, only the operation code must contain data.
OMESLTN	The next terminal in the internal OMNIS terminal list is to be set; this is the next of the terminals transferred in the TERM area.



With the OMESLTx calls, only active terminals are used (and this also includes the *OPNCON* and *HOLD* statuses). Declared terminals are not used. The [table 3 on page 266](#) shows which parameter areas contain data after successful OMESLTx calls.

- OMESLPA A certain partner of the current terminal, selected via PAC, is to be set. In the PTNR area, the fields *OMEPPAC* and *OMEPPACL* must contain data.
- OMESLPP A certain partner of the current terminal, selected via partner name and processor name, is to be set. In the PTNR area, the fields *OMEPPNAME* and *OMEPPPROC* must contain data.
- OMESLPPF The first partner in the partner list of the current terminal is to be set. In the OMS area, only the operation code must contain data.
- OMESLPN The next partner of the current terminal is to be set; this is the next of the partners transferred in the PTNR area.
-  The [table 3 on page 266](#) shows which parameter areas contain data after successful OMESLTX calls.
Only partners of the current terminal are sought.
- OMESVPA The screen is to be saved for a partner. In the PTNR area, the fields *OMEPPAC* and *OMEPPACL* must contain data. The field *OMETREST* contains the specification as to whether, when the partner is selected again, the screen contents are sent or whether the message is sent to the partner.
- OMESTXPA A certain partner of the current terminal, selected via PAC, is to be set **exclusively**. To permit this, a Select Partner must previously have been performed. In the PTNR area, the fields *OMEPPAC* and *OMEPPACL* must contain data.
- OMESHDR The first 16 net bytes of a message are to be requested.
- OMETIMS The timer interval for cyclic calling of the exit routine is to be defined. The exit routine is to be called every *OMEOTIME* seconds. The number of seconds between the calls is specified in the field *OMEOTIME*. The precision is 10 seconds. The maximum permissible value is 32767. When *OMEOTIME=0*, cyclic calling is cleared.
- OMEOPTRM A connection to a terminal is to be set up. In the TERM area, the fields *OMETNAME* and *OMETPROC* must contain data.

The following should be noted when setting up the parameter areas before the exit routine calls OMNIS:

The exit routine was previously called by OMNIS, which means that certain parameter areas, depending on the operation code, already contain data as shown in [table 1 on page 261](#). In all cases, the OMS area is already set up.

If the exit routine provides its own OMS area, this area must be initialized with the OMS area provided by OMNIS.

The *OMESLT_x* calls are used for setting up the TERM area. The PTNR area and, if applicable, the MUX area can be set up with the *OMESLP_x* calls (see [table 3 on page 266](#)).

The desired options and messages (commands), respectively, must be placed in the EDIT/VTSUCB area and the communication area. If the exit routine provides the VTSUCB, it must set *OMEVTCB* to 'Y'. The same applies if the exit routine wishes to receive the VTSUCB after a Select call.

The parameter areas in which data must actually be placed depend on the combination of calls used:

A parameter area needs to be set if this is necessary, as indicated by [table 2 on page 263](#), for the desired operation code and the area has not yet been set up (see [table 1 on page 261](#)) or a new (modified) setting is required.

Return information for exit routine calls OMNIS calls

The exit routine is provided with information on the success or otherwise of an OMNIS call in the form of return codes or error messages.

Synchronous return codes

Return codes are transferred to the *OMEORTC* field in the OMS area (see [page 282](#) and [page 294](#)). If this field contains the value *OMERTCOK*, the respective call was executed successfully.

After successful execution of the calls *OMESLT_x* and *OMESLP_x*, the following parameter areas contain data:

OPCODES	OMS	TERM	PTNR	EDIT	MUX	TWORK	PWORK	VTSUCB *
OMESLTA	X	X		X		X		X
OMESLTF	X	X		X		X		X
OMESLTN	X	X		X		X		X
OMESLPA	X		X	X	o	X	X	X
OMESLPP	X		X	X	o	X	X	X
OMESLPF	X		X	X	o	X	X	X
OMESLPN	X		X	X	o	X	X	X

X . . . Area containing data
 O . . . Area containing data for UTM partners with a multiplex connection
 * . . . The VTSUCB area is set up only if *OMEVTCB* was set to 'Y'.

Table 3: Parameter areas containing data after OMESLT_x and OMESLP_x calls

If an invalid operation code was transferred when OMNIS was called, the field *OMEORTC* contains the value *OMERIOPC*.

If an impermissible operation code was transferred when OMNIS was called, the field *OMEORTC* contains the value *OMERNOPC*.

In this case and in the case of other errors, the exit routine is called again without OMNIS taking any further action in the meantime.

Asynchronous return information

is, for the most part, the result of DCAM calls initiated by OMNIS commands. Return information is passed on to the exit routine in the form of OMNIS messages.

Summary of possible return information

Opcode	Synchronous return code	Asynchronous return information
OMECOM	-	OMNIS message
OMEMES	-	OMNIS message
OMEMTP	-	OMNIS message
OMEMPT	-	OMNIS message
OMELGET	-	OMNIS message
OMELPUT	-	OMNIS message
OMESHCY	-	OMNIS message
OMESLTA	OMERSLTA	-
OMESLTF	OMERSLTF	-
OMESLTN	OMERSLTN	-
OMESLPA	OMERSLPA	-
OMESLPP	OMERSLPP	-
OMESVPA	OMERSLPA	-
OMESLPF	OMERSLPF	-
OMESLPN	OMERSLPN	-
OMESTXPA	OMERSLPA	-
OMESHDR	OMERSHDR	* OMNIS message
OMETIMS	-	
OMEOPTRM	OMEROPER/OMEROPAC/ OMEROPOP/OMEROPAP/ OMEROPDC	

Table 4: Return information for exit routine calls OMNIS calls

* If none of the synchronous return codes listed is received, an asynchronous OMNIS message may arrive.

Possible combinations

For each operation code with which the exit routine is called, only certain operation codes are permissible when the exit routine calls OMNIS. These are summarized in the following table, which uses abbreviations for the operation codes (*INIT* for *OMEINIT*, etc.):

E X I T	OMNIS CALLS EXIT ROUTINE										END	
		INIT	COM	MES	MPT	MTP	LGET	LPUT	SHCY	TIMC		
C A L L S	COM	+	+	+	+	+	-	-	-	+	-	
	MES	+	+	+	+	+	-	-	-	+	-	
	MPT	+	+	+	+	+	-	-	-	+	-	
	MTP	+	+	+	+	+	-	-	-	+	-	
	LGET	-	-	-	-	-	+	-	-	-	-	
	LPUT	-	-	-	-	-	-	+	-	-	-	
	SHCY	-	-	-	-	-	-	-	+	-	-	
	SLTx	+	+	+	+	+	-	-	-	+	-	
	SLPx	+	+	+	+	+	-	-	-	+	-	
	SVPA	+	+	+	+	+	-	-	-	+	-	
	STXPA	+	+	+	+	+	-	-	-	+	-	
	SHDR	-	+	+	+	+	-	-	-	+	-	
	TIMS	+	+	+	+	+	-	-	-	+	-	
	OPTRM	+	+	+	+	+	-	-	-	+	-	
	+ . . . permitted - . . . not permitted x . . . stands for A, F, N, P											

Table 5: Permissible calls to OMNIS from exit routines

Example

If an exit routine is called with *OMECOM*, the exit routine can call OMNIS only with *OMECOM*, *OMEMES*, *OMEMPT*, *OMEMTP*, *OMESLTx*, *OMESLPx*, *OMESHDR* and *OMETIMS*, while *OMELGET*, *OMELPUT* and *OMESHCY* may not be used.

14.4 Parameter areas

The individual parameter areas and also their associated fields are described in the following section.

OMS area

The meaning of the individual fields of the OMS area is as follows:

OMEORDER	Version of the exit interface.
OMEORDER	Operation code specifying the event which caused OMNIS to call the exit routine or the action required of OMNIS by the exit routine. The individual operation codes are described on page 261 .
OMEORDER	Return code which OMNIS returns after being called from the exit routine (see page 266).
OMEORDER	Length of the message OMNIS passes on to the exit routine or which the exit routine passes on to OMNIS. If no message exists, this field contains 0.
OMEORDER	This field indicates the intervals at which OMNIS is to call the exit routine with the operation code <i>OMETIMC</i> . The interval is specified in seconds, with a precision of 10 seconds. The field is set up in the <i>OMETIMS</i> call (setting the timer interval). In the case of 0, cyclic calling is cleared.
OMEORDER	Name of the user's own computer, i.e. the computer on which the OMNIS is loaded in which the exit routine is executing.
OMEORDER	Application name under which OMNIS is executing.
OMETWRKL	Length of the TERM work area.
OMEPRWKL	Length of the PTNR work area.
OMEORDER	Length of the key for start sequences.
OMEMAXML	Maximum message length.

TERM area

The individual fields of the TERM area have the following meaning:

- OMETROU indicates whether the terminal is connected with OMNIS directly or indirectly (see [page 282](#) and [page 294](#)).
- OMETTYP indicates the (internal) OMNIS terminal type (see [page 282](#) and [page 294](#)).
- OMETNAME indicates, together with *OMETPROC*, the network address of the terminal (station name). The field is set up in the calls *OMESLTA* (select terminal by Name) and *OMEOPTRM* (connection setup to terminal).
- OMETPROC indicates, together with *OMETNAME*, the network address of the terminal (processor name). The field is set up in the calls *OMESLTA* (select terminal by Name) and *OMEOPTRM* (connection setup to terminal).
- OMETPAC The PAC of the partner to whom the last input at the terminal was directed or the PAC of the partner to whom the next message entered without a PAC is sent. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).
- OMETPACL Length of the current partner's PAC. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).
- OMETOKEY Code of the K key which can be used to switch the terminal to OMNIS command mode. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).
- OMETCKEY Code of the K key which acts as the CALL-KEY at the terminal. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).
- OMETBKEY Code of the K key which acts as the BREAK-KEY at the terminal. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).
- OMETSKEY Code of the send key with which the last message was sent from the terminal. This field can (if required) be set up in the calls *OMECCOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal).

OMETSTYP	Station type of the terminal. The terminal type is mapped in accordance with the DCSTA macro.
OMETCOL	Color in which system messages are output by OMNIS in line mode. This field can (if required) be set up in the calls <i>OMECCOM</i> (command to OMNIS), <i>OMEMES</i> (OMNIS message to terminal), <i>OMEMTP</i> (message to the partner) and <i>OMEMPT</i> (message to terminal).
OMETSTAT	Status of the terminal (see page 282 and page 294).
OMETERR	This error indicator is set if an error occurs on sending a message to the terminal. It cannot be guaranteed that further messages can be output on the terminal (see page 282 and page 294).
OMETADM	specifies whether the terminal has administrator authorization.
OMETIC	specifies whether the terminal has an ID card reader.
OMETVTCB	specifies whether the terminal uses VTSUCB. Since new functions will be supported only with VTSUCB, the VTSUCB should always be used for new exit routines.
OMETTP	specifies the transport protocol (ISO/NEA) of the terminal.
OMETREST	applies only to the operation code <i>OMEOMTP</i> , concerns restart behavior for the previously set partner. In the case of <i>OMETRSTY</i> and if a restart message was entered, provided a saved screen exists, this screen is output and the saved P keys are additionally loaded. Otherwise, the message is sent.
OMETTID	specifies the <i>TID</i> of the terminal.
OMETUSER	specifies the user who has logged in to the terminal through OMNIS-MENU.



The values assigned to the K keys are shown on [page 282](#) (OMEEQU) and [page 294](#) (OMECEQU). 255 for *STD* and 0 for *NO* are special values. These two values are permitted only for codes allocated by OMNIS; they cannot occur as key codes during message reception.

PTNR area

The individual fields of the PTNR area have the following meaning:

OMEPROU	indicates whether the connection is direct or indirect or whether it is a multiplex connection.
OMEPTYP	Type of partner (TIAM, UTM, etc.).
OMEPPNAME	specifies, together with <i>OMEPPROC</i> , the network address of the printer (partner name). The field is set up in the call <i>OMESLPP</i> (select partner by name).
OMEPPROC	specifies, together with <i>OMEPPNAME</i> , the network address of the partner (processor name). The field is set up in the call <i>OMESLPP</i> (select partner by name).
OMEPPAPPL	Name of the substitute application under which the partner knows OMNIS (i.e. it appears to be communicating with a program with the network address <i>OMEPPAPPL/OMEOPROC</i>).
OMEPPAC	The PAC with which the partner is addressed. The field is set up in the call <i>OMESLPA</i> (select partner on basis of PAC).
OMEPPACL	Length of the PAC. The field is set up in the call <i>OMESLPA</i> (select partner on basis of PAC).
OMEPPKEY	Code of the partner key. This field can (if required) be set up in the calls <i>OMECCOM</i> (command to OMNIS), <i>OMEEMES</i> (OMNIS message to terminal), <i>OMEEMTP</i> (message to the partner) and <i>OMEEMPT</i> (message to terminal). <i>255</i> for <i>STD</i> and <i>0</i> for <i>NO</i> are special values. Both these values are only permitted for the codes assigned by OMNIS, they may not be used as key codes for receiving messages.
OMEPPSSH	Message type of the last message received or sent by the partner. This field is relevant only to the partner TIAM.
OMEPPCOL	Color in which messages from the partner are output to the terminal (valid only in line mode). This field can (if required) be set up in the calls <i>OMECCOM</i> (command to OMNIS), <i>OMEEMES</i> (OMNIS message to terminal), <i>OMEEMTP</i> (message to the partner) and <i>OMEEMPT</i> (message to terminal).
OMEPPSTAT	indicates the status of the partner (HOLD, AKTIV, ...).
OMEPPERR	This error flag is set if an error occurred on sending a message to the partner. There is no guarantee that other messages were received by the partner.
OMEPPVTCB	indicates whether the partner uses VTSUCB.

Since, in future, new functions will be supported only with VTSUCB, the VTSUCB should be used for all new exit routines.

OMEPPID specifies the partner's PID.



Possible values for the individual fields can be found on [page 282](#) (OMEEQU) and [page 294](#) (OMECEQU).

EDIT area

The individual fields of the EDIT area have the following meaning:

- OMEEOMO Output mode: indicates whether output is to be in line, format or physical mode.
- OMEEIMO Input mode: indicates whether input is to be in line, format or physical mode.
- OMEEOHC Hardcopy operand: indicates whether output should be logged on a hardcopy unit.
- OMEEOHO If this operand is specified, the entire message is regarded as a single output unit.
- OMEEOEX indicates whether output is to be in extended line mode.
- OMEEOLG If this option is specified, logical control characters (NL, NP) are not interpreted.
- OMEEOBL If this option is specified, output to the terminal is accompanied by an audible alarm.
- OMEEOHD The output message includes a message header.
- OMEEIHD The input message is sent to the partner with a message header.
- OMEEIBS Underline characters are passed on to the user.
- OMEEIFC The code of the send key is passed on to the partner along with the message.
- OMEEICD The input message is requested in blanking mode.
- OMEEILC If this option is specified, lowercase letters also are sent to the partner for the next input.
- OMEEDIA The message is a strict dialog message, i.e. no further outputs are permitted at the terminal. The input is sent to the partner who sent the message to the terminal.
- OMEEVTCB specifies whether or not the exit routine passes the EDIT options in the VTSUCB or whether or not it wants to receive the VTSUCB for a select call. *OMEEVTCB* must be set to *Y* if the exit routine has set the EDIT options in

the VTSUCB for *OMECOM*, *OMEMES*, *OMEMTP* and *OMEMPT*. This also applies in the event that you wish, with the *OMESLxx* calls, to receive the currently valid EDIT options for the selected partner or terminal in the VTSUCB.

Since all new functions will be supported only with the VTSUCB, the VTSUCB should be used in all new exit routines.



The EDIT options are taken over from the parameter area prior to each OMNIS call with the operation code *OMEMES*. However, they only need to be provided with new data if they are intended to be modified. They are not modified by OMNIS messages.

If the exit routine does not describe the EDIT options with VTSUCB, this field can be set up in the calls *OMECOM* (command to OMNIS), *OMEMES* (OMNIS message to terminal), *OMEMTP* (message to the partner) and *OMEMPT* (message to terminal). *OEETCB* is thus set to *N*.

Possible values for the individual fields can be found on [page 282](#) (*OMEEQU*) and [page 294](#) (*OMECEQU*). See the "TIAM" manual [14] for further information on the meaning of the EDIT options.

MUX area

The individual fields of the MUX area have the following meaning:

- OMEUSER** Name of the user; if this field contains only blanks, no user was specified. If the *KDCSIGN* parameters are provided by the exit routine during connection setup to a UTM MUX partner, this field must contain data.
- OMEUPASS** User password; if this field contains only blanks, no password was specified. If the *KDCSIGN* parameters are provided by the exit routine during connection setup to a UTM MUX partner, this field must contain data.
- OMEUNPAS** New password; if this field only contains blanks, no password was specified. If the *KDCSIGN* parameters are provided by the exit routine during connection setup to a UTM MUX partner, this field must contain data.
- OMEUFORM** Name of the format last sent by UTM. If several partial formats are being displayed on the screen, this is the name of the last partial format.
- OMEUTACV** Transaction code of the current conversation. This field is only evaluated by UTM at the start of a conversation.
- OMEUTACT** Transaction code of the current program unit.

- OMEUPROC** Name of the processor under which the UTM session is set up. This field must be supplied with data during connection setup to a UTM MUX partner. You should specify the PTERM name from which the login to UTM is to be performed. If no value is specified for this field (default value: blanks), the processor name of the terminal is used.
- OMEUNAME** Name of the partner under which the UTM session is set up. This field must be supplied with data during connection setup to a UTM MUX partner. You should specify the PTERM name from which the login to UTM is to be performed. If no value is specified for this field (default value: blanks), the *APPNAME* specified in the *OPNCON* command is used. If this is also missing, the station name generated in PDN is used. The combination *OMEUPROC/OMEUPTN* must be generated in UTM.
- OMEUSEQ** Sequence number (value between x'00' and x'FF'). The sequence number is sent to UTM for the purpose of message flow control. When a message is received, it is possible to check whether a message has been lost. OMNIS itself does not execute message flow control using sequence numbers. This field can, if required, be supplied with data in the calls *OMEMTP* and *OMEMPT* for UTM MUX partners.
- OMEUVER** Version number of the UTM MUX protocol.
- OMEUOPC** Operation code for the action to be carried out by UTM. This field can, if required, be supplied with data in the calls *OMEMTP* and *OMEMPT* for UTM MUX partners.
- OMEUSTAT** Status of processing from the point of view of UTM.
- OMEUSEND** indicates whether OMNIS or UTM is authorized to send a message.
- OMEURTC** Return code from UTM.
- OMEUCRDL** Length of the ID card reader information. This field must be supplied with data during connection setup to a UTM MUX partner if the exit routine makes available ID card reader data. If the field contains binary zeros, this means that there is no ID card reader information.
- OMEUCRDD** ID card reader information (with the length *OMEUCRDL*). This field must be supplied with data during connection setup to a UTM MUX partner if the exit routine makes available ID card reader data.
- OMEUSTCH** specifies the station characteristics. The field and its subfields must be supplied with data during connection setup to a UTM MUX partner if terminal attributes other than those possessed by the terminal are to apply in respect of UTM. This also applies if the message traffic is to be handled without a real terminal. The possible settings may be found in the "DCAM Macros" manual [8] (*YINQUIRE* call). Binary zeros mean that the actual *YINQUIRE* data is to be forwarded to UTM. The stations characteristics are

therefore initialized with binary zeros by OMNIS.

- OMEUPTYP specifies the partner type.
- OMEUDVTY specifies the device type.
- OMEUCHR2 specifies the terminal characteristics.
- OMEUCHR3 specifies the terminal characteristics.
- OMEUCHLL specifies the maximum line length.
- OMEUCHLN specifies the maximum number of lines.
- OMEUCHCA contains the hardcopy address.
- OMEUCODE specifies whether the message exists in 7-bit or 8-bit code.
- OMEUCCS is the **c**oded **c**haracter **s**et name of the message. OMNIS is not responsible for the contents of messages which are delivered to an exit routine. However, it passes information about the structure of the message to the exit routine in the fields *OMEUCODE* and *OMEUCCS*.



The fields used to identify the user (*OMEUSER*, *OMEUPASS*, *OMEUNPAS*, *OMEUCRDL*, *OMEUCRDD* and *OMEUSTCH*) are interpreted during execution of the *OPNCON* command and passed on to UTM for *KDCSIGN* handling. Possible values for the individual fields can be found on [page 282](#) (*OME EQU*) and [page 294](#) (*OME EQU*).

Mapping area

The individual fields of the mapping area have the following meaning:

- OMEMAPPL specifies the length of the mapping message (including *OMEMAPPL*). When *OMEMAPPL=0*, no mapping format is involved.
- OMEMAP Mapping portion of the OMNIS message, as supplied by the MSG7 macro. See the "Executive Macros" manual [4].



If an exit routine modifies the message, the mapping portion must similarly be supplied with data (e.g. by resetting *OMEMAPPL*). This ensures that the subsequently called exit routines receive consistent data.

HCY area

The individual fields of the HCY area have the following meaning:

- OMEHNAME specifies, together with *OMEHPROC*, the network address of the printer (partner name).
- OMEHPROC specifies, together with *OMEHNAME*, the network address of the printer (processor name).
- OMEHHAC HAC with which the printer is addressed.
- OMEHHACL Length of the HAC.
- OMEHSTAT specifies the printer status (see [page 282](#) and [page 294](#)).
- OMEHSEQ1, OMEHSEQ2
These fields are taken over during the call *OMESHCY* (output message on a hardcopy unit) and sent to the hardcopy unit. They contain a number between 32 and 127 used for message flow control. This number is included with each output to printer and checked by OMNIS when acknowledgment is received.
- This allows OMNIS to determine (using the sequence numbers) in which message an error occurred.
- OMEHHID specifies the HID of the hardcopy unit.

14.5 Assembler interface

Three macros, which define the interface to OMNIS, are provided for Assembler.

- OMEPARL: List of parameters which are transferred in R1
 OMEPAR: Data structures which are interpreted by OMNIS
 OMEEQU: Equates for supplying the data structures in OMEPAR with values

14.5.1 Calling an exit routine in Assembler (OMEPARL)

Five data field addresses are passed to the exit routine via register R1. The parameter list is described by the macro *OMEPARL*:

```

*          OMEPARL  ASC VER000
          PRINT GEN
DSECT     START 0
*****
* OMEPARL                                           *
*****
          OMEPARL  MF=D
OMEPARL  DSECT  ,
OMEPARO  DS      A          OMNIS-SPECIFIC DATA
OMEPARM  DS      A          MESSAGE-BUFFER
OMEPART  DS      A          TERMINAL-WORK-AREA
OMEPARP  DS      A          PARTNER-WORK-AREA
OMEPARV  DS      A          VTSUCB-AREA

```

The addresses in the list have the following meaning:

- OMEPARO Address of the OMNIS parameter area (*OMEPAR*) which contains the OMS, TERM, PTNR, EDIT, MUX, HCY or mapping area.
- OMEPARM Address of the message which OMNIS received from the terminal or the partner / is sending.
- OMEPART Address of the terminal-specific work area.
- OMEPARP Address of the partner-specific work area.
- OMEPARV Address of the VTSUCB area.

It is possible to leave the exit routine by returning to the address transferred in register 14 if no actions are to be performed.

If the call is to be processed further, then OMNIS must be called.

Exit routine calls OMNIS

- The address of the parameter list (OMEPARL) must be specified in register R1.
- The address of the save area must be specified in register R13. Since the exit routines can be called recursively, only exit routines structured with ASSEMBH-XT should be used.
- Register R15 must be loaded with the address of OMSCALL (L R15,=V(OMSCALL)).
- Control is returned with BASR R14, R15.

14.5.2 Data structures (OMEPAR)

```

*****
*  OMEPAR DEFINES THE PARAMETER LIST FOR THE NEW          *
*  EXIT INTERFACE                                         *
*****
    OMEPAR MF=D
*****
*  OMEPAR DEFINES THE OMNIS DATA FOR EXIT INTERFACE      *
*****
    OMEPAR  DSECT ,
*****
*GENERAL PARAMETERS                                     *
*****
*  OMNIS PARAMETERS                                       *
*****
OMEOVER DC   H'4'  R   VERSION OF EXIT-INTERFACE
OMEOPC  DC   H'0'  R/W CODE OF OPERATION
OMEORTC DC   H'0'  R   RETURNCODE OF OMNIS CALL
OMEOMLEN DC  H'0'  R/W LENGTH OF MESSAGE
OMEQTIME DC  H'0'  R/W INTERRUPT TIME
OMEOPROC DC  CL8'  'R  NAME OF OWN PROCESSOR
OMEONAME DC  CL8'  'R  APPLICATION-NAME OF OMNIS
OMETWRKL DC  H'0'  R   LENGTH OF TERMINAL WORKAREA
OMEPRKLC DC  H'0'  R   LENGTH OF PARTNER WORKAREA
OMEQKEYL DC  H'0'  R   LENGTH START-SEQUENCE KE
OMEMAXML DC  H'0'  R   MAX. MESSAGELENGTH
    SPACE
    DS      4A      RESERVED
*****
*  TERMINAL AREA                                          *
*****
OMETROU DS   CL1  R   TERMINAL IS DIRECT/INDIRECT
OMETTYP DS   CL1  R   OMNIS-TYPE OF TERM-ENTRY

```

```

OMETNAME DS    CL8   R/W  STATION NAME OF TERM.
OMETPROC DS    CL8   R/W  PROCESSOR NAME OF TERM.
OMETPAC  DS    CL4   R/W  PAC OF PARTNER
OMETPACL DS    H     R/W  LENGTH OF PAC
OMETOKEY DS    H     R/W  CODE OF OMNIS-KEY
OMETCKEY DS    H     R/W  CODE OF CALL-KEY
OMETBKEY DS    H     R/W  CODE OF BREAK-KEY
OMETSKEY DS    H     R/W  KEY USED TO SEND MESSAGE
OMETSTYP DS    H     R    STATION-TYPE OF TERMINAL
OMETCOL  DS    CL1   R/W  COLOR OF MESSAGES
OMETSTAT DS    CL1   R    STATUS OF TERMINAL
OMETERR  DS    CL1   R    ERROR-CODE OF TERMINAL
OMETADM  DS    CL1   R    OMNIS ADMINISTRATION (Y/N)
OMETIC   DS    CL1   R    IDENTITY-CARD READER
OMETVTCB DS    CL1   R    TERM USES VTSUCB
OMETTP   DS    CL1   R    TRANSPORT-PROTOCOL OF APPL
OMETREST DS    CL1   R/W  RESTART OF SELECTED PARTNER
OMETRES1 DS    CL2   R    RESERVED
OMETTID  DS    A     R    TID OF TERMINAL
OMETUSER DC    CL8   R    USERNAME OR BLANK

```

DS 4ARESERVED

* PARTNER AREA *

```

OMEPROU DS    CL1   R    ROUTE=DIRECT/INDIRECT/MUX
OMEPTYP DS    CL3   R    TYP OF PTN-ENTRY
OMEpname DS    CL8   R/W  PARTNER NAME
OMEPPROC DS    CL8   R/W  PARTNER PROCESSOR
OMEPPAPL DS    CL8   R    APPLICATION-NAME
OMEPPAC  DS    CL4   R/W  PAC OF PARTNER
OMEPPACL DS    H     R/W  LENGTH OF PAC
OMEPPKEY DS    H     R/W  CODE OF KPAC
OMEPPSSH DS    H     R    TYPE OF MESSAGE
OMEPCOL  DS    CL1   R/W  COLOR OF PARTNER-MESSAGE
OMEPPSTAT DS    CL1   R    STATUS OF PARTNER
OMEPPERR DS    CL1   R    ERROR-CODE OF PARTNER
OMEPPVTCB DS    CL1   R    PARTNER USES VTSUCB
OMEPPRES1 DS    CL2   R    RESERVED
OMEPPID  DS    A     R    PID OF PARTNER
OMEPPRES2 DS    7A    R    RESERVED

```

* EDIT OPTIONS *

```

OMEEOMO DS    CL1   R/W  MODE OF OUTPUT
OMEEOIMO DS    CL1   R/W  MODE OF INPUT
OMEEOHC  DS    CL1   R/W  HARDCOPY
OMEEOHO  DS    CL1   R/W  HOMOGENEOUS OUTPUT
OMEEOEX  DS    CL1   R/W  EXTENDED-LINE OUTPUT

```



```

OMEEOLG DS CL1 R/W LOG.CHARS INTERPRET.
OMEEOBL DS CL1 R/W BELL
OMEEOHD DS CL1 R/W HEADER PRESENT
OMEEIHD DS CL1 R/W HEADER REQUIRED
OMEEIBS DS CL1 R/W BACKSPACE
OMEEIFC DS CL1 R/W FUNCTION-CODE
OMEEICD DS CL1 R/W CONFIDENTIAL INPUT
OMEEILC DS CL1 R/W LOWER CASE TRANSLAT.
*****
* OMNIS CONTROL FIELDS *
*****
OMEEDIA DS CL1 R/W STRICT DIALOG
OMEEVTCB DS CL1 R/W VTSUCB PROVIDED BY EXIT
          DS CL1 RESERVED
OMEERES1 DS 4A RESERVED
OMEHEADL EQU *
*****
* MAPPING FORMAT FOR OMNIS MESSAGES CORRESPONDING TO MSG7 MACRO *
*****
MEMAPPL DS H R/W LENGTH OF MAPPING FORMAT
MEMAPP DS CL254 R/W MAPPING FORMAT
          ORG OMEHEADL
*****
* MUX AREA *
*****
          ORG OMEHEADL
OMEUSER DS CL8 R/W KDCSIGN NAME
OMEUPASS DS CL8 R/W KDCSIGN PASSWORD
OMEUNPAS DS CL8 R/W NEW PASSWORD TO UTM
OMEUFORM DS CL8 R FORMAT NAME
OMEUTACV DS CL8 R TAC, STARTING CONVERSATION
OMEUTACT DS CL8 R TAC OF ACTIVE TRANSACTION
OMEUPROC DS CL8 R/W PROCESSOR NAME
OMEUNAME DS CL8 R/W PARTNER NAME
OMEUSEQ DS H R/W SEQUENCE NUMBER
OMEUVER DS CL4 R VERSION OF PROTOCOL
OMEUOPC DS CL1 R/W OPERATION CODE OF UTM
OMEUSTAT DS CL1 R PEND-CODE OF UTM
OMEUSEND DS CL1 R PERMISSION TO SEND
          DS CL1 RESERVED
OMEURTC DS H R RETURN CODE OF UTM
OMEUCRDL DS H R/W ID-CARD LENGTH
OMEUCRDD DS CL256 R/W ID-CARD DATA
OMEUSTCH DS OCL8 STACHARS FOR BIND-REQUEST
OMEUPTY DS CL1 R/W PARTNER TYPE
OMEUDVTY DS CL1 R/W DEVICE-TYPE
OMEUCHR2 DS CL1 R/W TERMINAL CHARACTERISTIC
OMEUCHR3 DS CL1 R/W TERMINAL CHARACTERISTIC

```

```

OMEUCHLL DS    CL2    R/W  MAX. LINE-LENGTH
OMEUHLN  DS    CL1    R/W  MAX. NUMBER OF LINES
OMEUCHCA DS    CL1    R/W  CENTRAL HARDCOPY ADDRES
OMEUCODE DS    CL1    R    7/8-BIT MESSAGE
OMEURES1 DS    CL1    R    RESERVED
OMEUCCS  DS    CL8    R    CCSNAME OF MESSAGE
OMEURES2 DS    2A    R    RESERVED
          ORG OMEHEADL
*****
*   HARDCOPY DATA; DEFINED ONLY IF OME00PC = OMESSCY   *
*****
OMEHNAME DS    CL8    R    STATION NAME OF HARDCOPY
OMEHPROC DS    CL8    R    PROCESSOR NAME OF HARDCOPY
OMEHHAC  DS    CL4    R    HAC OF HARDCOPY
OMEHHACL DS    H      R    LENGTH OF HAC
OMEHSTAT DS    CL1    R    STATUS OF HARDCOPY
          DS    CL1    R    RESERVED
OMEHSEQ1 DS    H      R/W  SEQUENCE NUMBER PART 1
OMEHSEQ2 DS    H      R/W  SEQUENCE NUMBER PART 2
OMEHHID  DS    A      R    HID OF HARDCOPY
OMEHRES  DS    7A    R    RESERVED
          ORG
OMELEN   EQU    *-OMEPAR  LENGTH OF PARAMETER AREA
          OMEEQU MF=D

```

14.5.3 Definitions for supplying the data structures with values (OMEEQU)

```

*           OMEEQU    ASC VER000
          PRINT GEN
DSECT     START 0
          OMEEQU MF=D
OMEEQU    DSECT ,
*****
* EQUATES FOR EXIT INTERFACE VERSION          FIELD: OMEOVER *
*****
OMEAVERS EQU  X'0004'          ACTUAL EXIT INTERFACE VERSION
OMEVERS1 EQU  X'0001'
OMEVERS2 EQU  X'0002'
OMEVERS3 EQU  X'0003'
OMEVERS4 EQU  X'0004'
*****
* EQUATES FOR OPERATION CODE                  FIELD: OME00PC *
*****
OMEINIT  EQU  X'0004'  INITIALIZATION
OMECOM   EQU  X'0008'  OMNIS-COMMAND
OMEMES   EQU  X'0010'  OMNIS-MESSAGE

```

```

OMEMPT EQU X'0030' MESSAGE FROM PTN TO TRM
OMEMTP EQU X'0040' MESSAGE FROM TRM TO PTN
OMELGET EQU X'0090' READ FROM LOGGING FILE
OMELPUT EQU X'0094' WRITE INTO LOGGING FILE
OMESHCY EQU X'0100' SEND MESSAGE TO HARDCOPY
OMESLTA EQU X'1010' SELECT SPECIFIC TERMINAL
OMESLTF EQU X'1018' SELECT FIRST TERMINAL
OMESLTN EQU X'101C' SELECT NEXT TERMINAL
OMESLPA EQU X'1020' SELECT PARTNER (PAC)
OMESLPP EQU X'1024' SELECT PARTNER (PTN/PRO)
OMESLPF EQU X'1028' SELECT FIRST PARTNER
OMESLPN EQU X'102C' SELECT NEXT PARTNER
OMESVPA EQU X'1030' SAVE TERMINAL MEMORY AND/OR P-KEYS
OMESTXPA EQU X'1034' SET EXCLUSIVE PARTNER (PAC)
OMESHDR EQU X'1040' SELECT FIRST 16 BYTES
OMETIMS EQU X'1050' SET TIMER INTERRUPT
OMETIMC EQU X'1054' TIMER CALL
OMEEND EQU X'1056' UNLOAD OF EXIT-ROUTINE
OMEOPTRM EQU X'1400' OPEN CONNECTION TO TERMINAL
*****
* EQUATES FOR RETURN CODE FIELD: OMEORTC *
*****
OMERTCOK EQU X'0000' SUCCESSFUL COMPLETION
OMERIOPC EQU X'0004' ILLEGAL OPERATION CODE
OMERNOPC EQU X'0008' OPERATION CODE NOT ALLOWED
OMERSLTA EQU X'0010' TERMINAL NOT FOUND
OMERSLTF EQU X'0018' FIRST TERMINAL NOT FOUND
OMERSLTN EQU X'001C' NEXT TERMINAL NOT FOUND
OMERSLPA EQU X'0020' PARTNER NOT FOUND (PAC)
OMERSLPP EQU X'0024' PARTNER NOT FOUND (PTN/PRO)
OMERSLPF EQU X'0028' FIRST PARTNER NOT FOUND
OMERSLPN EQU X'002C' NEXT PARTNER NOT FOUND
OMERSHDR EQU X'0030' NO SELECTION OF HEADER
OMEROPER EQU X'0034' OMETNAME OR OMETPROC MISSING
OMEROPAC EQU X'0038' TERMINAL ALREADY ACTIVE
OMEROPOP EQU X'003C' TERMINAL ALREADY IN OPN STAT
OMEROPAP EQU X'0040' TERM IS A PROGRAM AND NOT DE
OMEROPDC EQU X'0044' DCM-ERROR WHEN REQUESTING OP
*****
* EQUATES FOR TERM ROUTING CODE FIELD: OMETROU *
*****
OMETIND EQU C'I' TERM IS INDIRECTLY CONNECTED
OMETDIR EQU C'D' TERM IS DIRECTLY CONNECTED
*****
* EQUATES FOR TERM TYPE FIELD: OMETTYP *
*****
OMETTERM EQU C'T' TERM IS A TERMINAL
OMETCONS EQU C'C' TERM IS A CONSOLE

```

```

OMETAPPL EQU  C'A'      TERM IS A PROGRAM
OMETSKP EQU  C'S'      TERM IS AN SKP
*****
*   EQUATES FOR TERM STATUS                               FIELD: OMETSTAT  *
*****
OMETACT EQU  C'A'      TERM IS ACTIVE
OMETHOLD EQU C'H'      TERM IS IN HOLD
OMETINA EQU  C'I'      TERM IS INACTIVE
OMETOPN EQU  C'O'      OPNCON PROCESSING
*****
*   EQUATES FOR TERM ERROR CODE                         FIELD: OMETERR   *
*****
OMETERRY EQU  C'Y'      ERROR HAS OCCURED
OMETERRN EQU  C'N'      NO ERROR OCCURED
*****
*   EQUATES FOR TERM ADM CODE                           FIELD: OMETADM   *
*****
OMETADMY EQU  C'Y'      OMNIS-ADMINISTRATOR=Y
OMETADMN EQU  C'N'      OMNIS-ADMINISTRATOR=N
*****
*   EQUATES FOR TERM IDENTITY CARD READER              FIELD: OMETIC   *
*****
OMETICY EQU  C'Y'      CARD READER AT TERMINAL
OMETICN EQU  C'N'      NO CARD READER
*****
*   EQUATES FOR TERM VTSUCB                             FIELD: OMETVTCB *
*****
OMETVTCTY EQU C'Y'      TERM USES VTSUCB
OMETVTCTN EQU C'N'      NO VTSUCB
*****
*   EQUATES FOR APPL TRANSPORT PROTOCOL                FIELD: OMETTP   *
*****
OMETTPN EQU  C'N'      APPL USES NEA-TRANSPORT-PROTOCOL
OMETTPI EQU  C'I'      APPL USES ISO-TRANSPORT-PROTOCOL
*****
*   EQUATES FOR PARTNER RESTART                        FIELD: OMETREST *
*****
OMETRSTY EQU  C'Y'      PARTNER RESTART FROM TERMINAL
OMETRSTN EQU  C'N'      NO PARTNER RESTART FROM TERMINAL
*****
*   EQUATES FOR PTN ROUTING CODE                       FIELD: OMEPROU  *
*****
OMEPIIND EQU  C'I'      PTN IS INDIRECT
OMEPIDIR EQU  C'D'      PTN IS DIRECT
OMEPIMUX EQU  C'M'      PTN IS MUX-PARTNER
*****
*   EQUATES FOR PARTNER TYPE                           FIELD: OMEPTY   *
*****

```

```

OMEPTIAM EQU C'TIA' TIAM-PARTNER
OMEPCAM EQU C'DCA' DCAM-PARTNER
OMEPUTM EQU C'UTM' UTM-PARTNER
OMEPUCON EQU C'UCN' UCON-PARTNER
OMEPSVP EQU C'SVP' SVP-PARTNER
OMEPSKP EQU C'SKP' SKP-PARTNER
*****
* EQUATES FOR MESSAGE TYPE FIELD: OME.SSH *
*****
OMESUW EQU 54 USER WRITE
OMESUR EQU 53 USER READ
OMESURA EQU 21 USER READ ANSWER
OMESUWR EQU 55 USER WRITE-READ
OMESUWRA EQU 23 USER WRITE-READ ANSWER
OMESSW EQU 50 SYSTEM WRITE
OMESSR EQU 49 SYSTEM READ
OMESSRA EQU 17 SYSTEM READ ANSWER
OMESSWR EQU 51 SYSTEM WRITE-READ
OMESSWRA EQU 19 SYSTEM WRITE-READ ANSWER
OMESESC EQU 1 ESCAPE-INTERRUPT
OMESBRK EQU 2 BREAK-INTERRUPT
OMESILL EQU 4 ILLEGAL CHARACTER
OMESASY EQU 65 ASYNCHRONOUS MESSAGE
*****
* EQUATES FOR COLORS OF TERMTYP 9752 FIELD: OME.COL *
*****
OMECSTD EQU C'S' STANDARD COLOR IS YELLOW
OMECBLU EQU C'B' BLAU
OMECCYA EQU C'C' CYAN
OMECPRE EQU C'G' GREEN
OMECYEL EQU C'Y' YELLOW
OMECMAG EQU C'M' MAGNETA
OMECRED EQU C'R' RED
OMECWHI EQU C'W' WHITE
*****
* EQUATES FOR PTN STATUS FIELD: OMEPSTAT *
*****
OMEPACT EQU C'A' PTN IS ACTIVE
OMEPHOLD EQU C'H' PTN IS IN HOLD
OMEPIA EQU C'I' PTN IS INACTIVE
OMEPOPN EQU C'O' OPNCON-PROCESSING
*****
* EQUATES FOR PTN ERROR CODE FIELD: OMEPERR *
*****
OMEPEPERR EQU C'Y' NO ERROR
OMEPEPERRN EQU C'N' ERROR HAS OCCURED
*****
* EQUATES FOR PTN VTSUCB FIELD: OMEPVTCB *

```

```

*****
OMEPVTCY EQU  C'Y'    PTN USES VTSUCB
OMEPVTCN EQU  C'N'    NO VTSUCB
*****
*   EQUATES FOR EDIT-OPTIONS                                FIELD: OMEE.. *
*****
*   MODE=
OMEELIN EQU  C'L'    LINE-MODE
OMEEPHY EQU  C'P'    PHYS-MODE
OMEEFOR EQU  C'F'    FORM-MODE
*   HARDCOPY=
OMEEHCCY EQU  C'Y'    HCCOPY=YES
OMEEHCCN EQU  C'N'    HCCOPY=NO
*   HOMOG.=
OMEEHOMY EQU  C'Y'    HOM=YES
OMEEHOMN EQU  C'N'    HOM=NO
*   EXTENDED-LINE=
OMEEXTY EQU  C'Y'    EXT=YES
OMEEXTN EQU  C'N'    EXT=NO
*   LOG. CHARS=
OMEELGY EQU  C'Y'    LOG=YES
OMEELGN EQU  C'N'    LOG=NO
*   BELL=
OMEEBELY EQU  C'Y'    BELL=YES
OMEEBELN EQU  C'N'    BELL=NO
*   HEADER=
OMEHHDY EQU  C'Y'    HDR=YES
OMEHHDN EQU  C'N'    HDR=NO
*   BACKSPACE=
OMEBSY EQU  C'Y'    BACKSPACE=YES
OMEBSN EQU  C'N'    BACKSPACE=NO
*   FUNCTION-CODE=
OMEFCDY EQU  C'Y'    FUNCT=YES
OMEFCDN EQU  C'N'    FUNCT=NO
*   CONFIDENTIAL DATA INPUT =
OMEICDY EQU  C'Y'    CONFIDENTIAL DATA=YES
OMEICDN EQU  C'N'    CONFIDENTIAL DATA=NO
*   LCASE=
OMEELCAY EQU  C'Y'    LCASE=YES
OMEELCAN EQU  C'N'    LCASE=NO
*   DIALOG=
OMEEDIAY EQU  C'Y'    STRICT DIALOG
OMEEDIAN EQU  C'N'    NO STRICT DIALOG
*   VTSUCB=
OMEVTCY EQU  C'Y'    OMNIS HAS PROVIDED VTSUCB
OMEVTCN EQU  C'N'    NO VTSUCB
*****
*   EQUATES FOR TERMINAL KEYS                                FIELD: OME..KEY *

```

```

OMEDUE1 EQU 00      DUE1-KEY
OMEDUE2 EQU 00      DUE2-KEY
OMEK1   EQU 1       K1-KEY
OMEK2   EQU 2       K2-KEY
OMEK3   EQU 3       K3-KEY
OMEK4   EQU 4       K4-KEY
OMEK5   EQU 5       K5-KEY
OMEK6   EQU 6       K6-KEY
OMEK7   EQU 7       K7-KEY
OMEK8   EQU 8       K8-KEY
OMEK9   EQU 9       K9-KEY
OMEK10  EQU 10      K10-KEY
OMEK11  EQU 11      K11-KEY
OMEK12  EQU 12      K12-KEY
OMEK13  EQU 13      K13-KEY
OMEK14  EQU 14      K14-KEY
OMEF1   EQU 21      F1-KEY
OMEF2   EQU 22      F2-KEY
OMEF3   EQU 23      F3-KEY
OMEF4   EQU 24      F4-KEY
OMEF5   EQU 25      F5-KEY
OMEF6   EQU 26      F6-KEY
OMEF7   EQU 27      F7-KEY
OMEF8   EQU 28      F8-KEY
OMEF9   EQU 29      F9-KEY
OMEF10  EQU 30      F10-KEY
OMEF11  EQU 31      F11-KEY
OMEF12  EQU 32      F12-KEY
OMEF13  EQU 33      F13-KEY
OMEF14  EQU 34      F14-KEY
OMEF15  EQU 35      F15-KEY
OMEF16  EQU 36      F16-KEY
OMEF17  EQU 37      F17-KEY
OMEF18  EQU 38      F18-KEY
OMEF19  EQU 39      F19-KEY
OMEF20  EQU 40      F20-KEY
OMEF21  EQU 41      F21-KEY
OMEF22  EQU 42      F22-KEY
OMEF23  EQU 43      F23-KEY
OMEF24  EQU 44      F24-KEY

```

* EQUATES FOR UTM DIALOG STATUS FIELD: OMEUSTAT *

```

OMEUDSIG EQU C'S'   SIGNON PROCESSING
OMEUDOV  EQU C'O'   OUT-OF-CONVERSATION (PEND FI,ER)
OMEUDIV  EQU C'V'   WAIT-IN-CONVERSATION (PEND RE)
OMEUDWT  EQU C'T'   WAIT-IN-TRANSACTION (PEND KP)

```

```

OMEUDINV EQU  C'X'      INVALID STATUS
*****
*  EQUATES FOR UTM SEND PERMISSION          FIELD: OMEUSEND *
*****
OMEUSUTM EQU  C'U'      AT UTM-APPLICATION
OMEUSTRM EQU  C'D'      AT TERMINAL
*****
*  EQUATES FOR UTM MESSAGE TYPE           FIELD: OMEUOPC  *
*****
OMEUMUSR EQU  C'M'      USER MESSAGE
OMEUMBCT EQU  C'B'      BROADCAST
OMEUMRES EQU  C'R'      RESTART (KDCDISP)
OMEUMUBD EQU  C'U'      UNBIND
OMEUMINV EQU  C'X'      INVALID OPCODE
*****
*  EQUATES FOR EXTENDED CODE OF MESSAGE   FIELD: OMEUCODE *
*****
OMEUCOD7 EQU  C'7'      7-BIT ENCODED MESSAGE
OMEUCOD8 EQU  C'8'      8-BIT ENCODED MESSAGE
*****
*  EQUATES FOR HCY STATUS (OF OMNIS)      FIELD: OMEHSTAT *
*****
OMEHACT EQU   C'A'      HCY IS ACTIVE
OMEHHOLD EQU  C'H'      HCY IS IN HOLD
OMEHINA EQU   C'I'      HCY IS INACTIVE
OMEHOPN EQU   C'O'      OPNCON-PROCESSING
      END      ,

```


14.5.4 H-Assembler

Since OMNIS is programmed in H-Assembler, exit routines programmed in H-Assembler can also be used. The H-Assembler runtime system is permanently linked to OMNIS. It can be called directly from an exit routine and does not need to be defined explicitly as EXTERNAL in the procedure header.

14.5.5 Columbus-Assembler

If Columbus-Assembler is used, there is no need for the user to administer the register save operation, as this is done by the runtime system. Since the Columbus-Assembler entries are also defined in the H-Assembler runtime system, Columbus-Assembler does not need to be defined explicitly as EXTERNAL in the procedure header.

14.6 COBOL interface

Two COBOL calls are offered as interface support:

OMECPAR Parameter list for the interface (data structures)

OMECEQU Predefined constants for supplying the data structures with values

14.6.1 Structure of an exit routine in COBOL

An exit routine in COBOL has the following structure:

```

IDENTIFICATION DIVISION.

PROGRAM-ID.  OMSEXIT.
AUTHOR.     OMNIS-DEVELOPER.
ENVIRONMENT DIVISION.
DATA DIVISION
WORKING-STORAGE SECTION.
* Various fields with constants which correspond to the Assembler
* equates:
COPY OMECEQU.
LINKAGE SECTION.
* Definition of the parameter area:
COPY OMECPAR.
* MESSAGE, DSTWORK and PTNWORK may be any length and may have any
* structure:
01 MESSAGE          PIC X(4096).
01 DSTWORK          PIC X(256).
01 PTNWORK          PIC X(256).
* Definition of the VTSUCB
COPY VTSUCBC.
PROCEDURE DIVISION.
* Entry point at which OMNIS calls the exit routine:
ENTRY USING OMEPAR MESSAGE DSTWORK PTNWORK VTSUCB
* OMNIS call:
CALL OMSCALL USING OMEPAR MESSAGE DSTWORK PTNWORK VTSUCB

```

14.6.2 Data structures (OMECPAR)

The meaning of the various fields is described on [page 263](#). The Assembler names and the COBOL names are the same.

```

*****
*   Copy Element                               OMECPAR   *
*****
*   OMECPAR INTERFACE TO OMNIS EXIT-ROUTINES.
    01  OMEPAR.
*   OMNIS OVERALL PARAMETERS.
    05  OMEOMS.
*   OMNIS SPECIFIC PARAMETERS.
        10  OMEOVER      PIC S9999 USAGE IS COMP SYNC.
        10  OMEOOPC     PIC S9999 USAGE IS COMP SYNC.
        10  OMEORTC     PIC S9999 USAGE IS COMP SYNC.
        10  OMEOMLEN    PIC S9999 USAGE IS COMP SYNC.
        10  OMEOTIME    PIC S9999 USAGE IS COMP SYNC.
        10  OMEOPROC    PIC X(8).
        10  OMEONAME    PIC X(8).
        10  OMETWRKL    PIC S9999 USAGE IS COMP SYNC.
        10  OMEPWRKL    PIC S9999 USAGE IS COMP SYNC.
        10  OMEOKEYL    PIC S9999 USAGE IS COMP SYNC.
        10  OMEMAXML    PIC S9999 USAGE IS COMP SYNC.
        10  FILLER      PIC S999999 USAGE IS COMP SYNC
                          OCCURS 4 TIMES.

    05  OMETERM.
*   TERMINAL SPECIFIC PARAMETERS.
        10  OMETROU     PIC X.
        10  OMETTYP     PIC X.
        10  OMETNAME    PIC X(8).
        10  OMETPROC    PIC X(8).
        10  OMETPAC     PIC X(4).
        10  OMETPACL    PIC S9999 USAGE IS COMP SYNC.
        10  OMETOKEY    PIC S9999 USAGE IS COMP SYNC.
        10  OMETCKEY    PIC S9999 USAGE IS COMP SYNC.
        10  OMETBKEY    PIC S9999 USAGE IS COMP SYNC.
        10  OMETSKY     PIC S9999 USAGE IS COMP SYNC.
        10  OMETSTYP    PIC S9999 USAGE IS COMP SYNC.
        10  OMETCOL     PIC X.
        10  OMETSTAT    PIC X.
        10  OMETERR     PIC X.
        10  OMETADM     PIC X.
        10  OMETIC      PIC X.
        10  OMETVTCB    PIC X.
        10  OMETTP      PIC X(1).
        10  OMETREST    PIC X(1).
        10  FILLER      PIC X(2).

```

```

        10 OMETTID      PIC S999999 USAGE IS COMP SYNC
        10 OMETUSER    PIC X(8).
        10 FILLER      PIC S999999 USAGE IS COMP SYNC
                           OCCURS 4 TIMES.

05 OMEPTN.
* PARTNER SPECIFIC PARAMETERS.
    10 OMEPROU        PIC X.
    10 OMEPTYP        PIC X(3).
    10 OMEPNAME       PIC X(8).
    10 OMEPPROC       PIC X(8).
    10 OMEPAPPL       PIC X(8).
    10 OMEPPAC        PIC X(4).
    10 OMEPPACL       PIC S9999  USAGE IS COMP SYNC.
    10 OMEPKEY        PIC S9999  USAGE IS COMP SYNC.
    10 OMEPSSH        PIC S9999  USAGE IS COMP SYNC.
    10 OMEPCOL        PIC X.
    10 OMEPSTAT       PIC X.
    10 OMEPERR        PIC X.
    10 OMEPVTCB       PIC X.
    10 FILLER         PIC X(2)
    10 OMEPPID        PIC S999999 USAGE IS COMP SYNC
    10 FILLER         PIC S999999 USAGE IS COMP SYNC
                           OCCURS 7 TIMES.

05 OMEEDIT.
* EDIT OPTIONS PARAMETERS.
    10 OMEEOMO        PIC X.
    10 OMEEIMO        PIC X.
    10 OMEEOHC        PIC X.
    10 OMEEOHO        PIC X.
    10 OMEEOEX        PIC X.
    10 OMEEOLG        PIC X.
    10 OMEEOBL        PIC X.
    10 OMEEOHD        PIC X.
    10 OMEEIHD        PIC X.
    10 OMEEIBS        PIC X.
    10 OMEEIFC        PIC X.
    10 OMEEICD        PIC X.
    10 OMEEILC        PIC X.
    10 OMEEEDIA       PIC X.
    10 OMEEVTCB       PIC X.
    10 FILLER         PIC X.
    10 FILLER         PIC S999999 USAGE IS COMP SYNC
                           OCCURS 4 TIMES.

05 OMEMAP.
* MAPPING FORMAT FOR OMNIS MESSAGES CORRESP. TO MSG7 MACRO
    10 OMEMAPPL       PIC S9999  USAGE IS COMP SYNC.
    10 OMEMAPP        PIC X(358).

```

05 OMEUTM.

* UTM-SPECIFIC PARAMETERS.

```

10 OMEUSER      PIC X(8).
10 OMEUPASS     PIC X(8).
10 OMEUNPAS     PIC X(8).
10 OMEUFORM     PIC X(8).
10 OMEUTACV     PIC X(8).
10 OMEUTACT     PIC X(8).
10 OMEUPROC     PIC X(8).
10 OMEUNAME     PIC X(8).
10 OMEUSEQ      PIC S9999 USAGE IS COMP SYNC.
10 OMEUVER      PIC X(4).
10 OMEUOPC      PIC X(1).
10 OMEUSTAT     PIC X(1).
10 OMEUEND      PIC X(1).
10 FILLER       PIC X(1).
10 OMEURTC      PIC S9999 USAGE IS COMP SYNC.
10 OMEUCRDL     PIC S9999 USAGE IS COMP SYNC.
10 OMEUCRDD     PIC X(256).
10 OMEUPTYP     PIC X(1).
10 OMEUDVTY     PIC X(1).
10 OMEUCHR2     PIC X(1).
10 OMEUCHR3     PIC X(1).
10 OMEUCHLL     PIC X(2).
10 OMEUHLN     PIC X(1).
10 OMEUCHCA     PIC X(1).
10 OMEUCODE     PIC X(1).
10 FILLER       PIC X(1).
10 OMEUCCS      PIC X(8).
10 FILLER       PIC S999999 USAGE IS COMP SYNC
                    OCCURS 2 TIMES.

```

05 OMEHCY REDEFINES OMEUTM.

* HARDCOPY SPECIFIC PARAMETERS.

```

10 OMEHNAME     PIC X(8).
10 OMEHPROC     PIC X(8).
10 OMEHHAC      PIC X(4).
10 OMEHHACL     PIC S9999 USAGE IS COMP SYNC.
10 OMEHSTAT     PIC X.
10 FILLER       PIC X.
10 OMEHSEQ1     PIC S9999 USAGE IS COMP SYNC.
10 OMEHSEQ2     PIC S9999 USAGE IS COMP SYNC.
10 OMEHHID      PIC S999999 USAGE IS COMP SYNC
10 FILLER       PIC S999999 USAGE IS COMP SYNC
                    OCCURS 82 TIMES.

```

14.6.3 Definitions for supplying the data structures with values (OMECEQU)

The meaning of the various fields is described on [page 263](#). The Assembler names and the COBOL names are the same.

```

*   OMECEQU  CONSTANTS FOR EXIT INTERFACE.
*   EXIT INTERFACE VERSION FIELD: OMEOVER
01  OMEVERS1          PIC S9999 VALUE IS 1 USAGE IS COMP.
01  OMEVERS2          PIC S9999 VALUE IS 2 USAGE IS COMP.
01  OMEVERS3          PIC S9999 VALUE IS 3 USAGE IS COMP.
01  OMEVERS4          PIC S9999 VALUE IS 4 USAGE IS COMP.

*   OPERATION CODES      FIELD: OMEOOPC
01  OMEINIT           PIC S9999 VALUE IS 4 USAGE IS COMP.
01  OMECOM            PIC S9999 VALUE IS 8 USAGE IS COMP.
01  OMEMES           PIC S9999 VALUE IS 16 USAGE IS COMP.
01  OMEMPT           PIC S9999 VALUE IS 48 USAGE IS COMP.
01  OMEMTP           PIC S9999 VALUE IS 64 USAGE IS COMP.
01  OMELGGET         PIC S9999 VALUE IS 144 USAGE IS COMP.
01  OMELGPUT         PIC S9999 VALUE IS 148 USAGE IS COMP.
01  OMEESHY          PIC S9999 VALUE IS 256 USAGE IS COMP.
01  OMESLTA          PIC S9999 VALUE IS 4112 USAGE IS COMP.
01  OMESLTF          PIC S9999 VALUE IS 4120 USAGE IS COMP.
01  OMESLTN          PIC S9999 VALUE IS 4124 USAGE IS COMP.
01  OMESLPA          PIC S9999 VALUE IS 4128 USAGE IS COMP.
01  OMESLPP          PIC S9999 VALUE IS 4132 USAGE IS COMP.
01  OMESLPPF         PIC S9999 VALUE IS 4136 USAGE IS COMP.
01  OMESLPN          PIC S9999 VALUE IS 4140 USAGE IS COMP.
01  OMESVPA          PIC S9999 VALUE IS 4144 USAGE IS COMP.
01  OMESTXPA         PIC S9999 VALUE IS 4148 USAGE IS COMP.
01  OMESHDR          PIC S9999 VALUE IS 4160 USAGE IS COMP.
01  OMETIMS          PIC S9999 VALUE IS 4176 USAGE IS COMP.
01  OMETIMC          PIC S9999 VALUE IS 4180 USAGE IS COMP.
01  OMEEND           PIC S9999 VALUE IS 4184 USAGE IS COMP.
01  OMEOPTRM         PIC S9999 VALUE IS 5120 USAGE IS COMP.
*   ERROR CODES        FIELD: OMEORTC
01  OMERTCOK         PIC S9999 VALUE IS 0 USAGE IS COMP.
01  OMERIOPC         PIC S9999 VALUE IS 4 USAGE IS COMP.
01  OMERNOPC         PIC S9999 VALUE IS 8 USAGE IS COMP.
01  OMERSLTA         PIC S9999 VALUE IS 16 USAGE IS COMP.
01  OMERSLTF         PIC S9999 VALUE IS 24 USAGE IS COMP.
01  OMERSLTN         PIC S9999 VALUE IS 28 USAGE IS COMP.
01  OMERSLPA         PIC S9999 VALUE IS 32 USAGE IS COMP.
01  OMERSLPP         PIC S9999 VALUE IS 36 USAGE IS COMP.
01  OMERSLPPF        PIC S9999 VALUE IS 40 USAGE IS COMP.
01  OMERSLPN         PIC S9999 VALUE IS 44 USAGE IS COMP.
01  OMERSHDR         PIC S9999 VALUE IS 48 USAGE IS COMP.
01  OMEROPER         PIC S9999 VALUE IS 52 USAGE IS COMP.

```

```

01  OMEROPAC          PIC S9999 VALUE IS 56 USAGE IS COMP.
01  OMEROPOP          PIC S9999 VALUE IS 60 USAGE IS COMP.
01  OMEROPAP          PIC S9999 VALUE IS 64 USAGE IS COMP.
01  OMEROPDC          PIC S9999 VALUE IS 68 USAGE IS COMP.
*   TERMINAL ROUTING CODE  FIELD: OMETROU
01  OMETIND           PIC X(1) VALUE IS "I".
01  OMETDIR           PIC X(1) VALUE IS "D".
*   TERMINAL TYPE          FIELD: OMETTYP
01  OMETTERM          PIC X(1) VALUE IS "T".
01  OMETCONS          PIC X(1) VALUE IS "C".
01  OMETAPPL          PIC X(1) VALUE IS "A".
01  OMETSKP           PIC X(1) VALUE IS "S".
*   TERMINAL STATUS        FIELD: OMETSTAT
01  OMETACT           PIC X(1) VALUE IS "A".
01  OMETHOLD          PIC X(1) VALUE IS "H".
01  OMETINA           PIC X(1) VALUE IS "I".
01  OMETOPN           PIC X(1) VALUE IS "O".
*   TERMINAL ERROR CODE    FIELD: OMETERR
01  OMETERRY          PIC X(1) VALUE IS "Y".
01  OMETERRN          PIC X(1) VALUE IS "N".
*   TERMINAL ADM-CODE      FIELD: OMETADM
01  OMETADMY          PIC X(1) VALUE IS "Y".
01  OMETADMN          PIC X(1) VALUE IS "N".
*   TERMINAL CARD-READ     FIELD: OMETIC
01  OMETICY           PIC X(1) VALUE IS "Y".
01  OMETICN           PIC X(1) VALUE IS "N".
*   TERMINAL VTSUCB        FIELD: OMETVTCB
01  OMETVTCY          PIC X(1) VALUE IS "Y".
01  OMETVTCN          PIC X(1) VALUE IS "N".
*   APPL TRANSPORT-PROF    FIELD: OMETTP
01  OMETTPN           PIC X(1) VALUE IS "N".
01  OMETTPI           PIC X(1) VALUE IS "I".
*   PARTNER RESTART        FIELD: OMETREST
01  OMETRSTY          PIC X(1) VALUE IS "Y".
01  OMETRSTN          PIC X(1) VALUE IS "N".
*   PTN ROUTING CODE       FIELD: OMEPROU
01  OMEPIND           PIC X(1) VALUE IS "I".
01  OMEPDIR           PIC X(1) VALUE IS "D".
01  OMEPMUX           PIC X(1) VALUE IS "M".
*   PTN TYPE                FIELD: OMEPTYP
01  OMEPTIAM          PIC X(3) VALUE IS "TIA".
01  OMEPDCAM          PIC X(3) VALUE IS "DCA".
01  OMEPUTM           PIC X(3) VALUE IS "UTM".
01  OMEPPDN           PIC X(3) VALUE IS "PDN".
01  OMEPUCON          PIC X(3) VALUE IS "UCN".
01  OMEPSVP           PIC X(3) VALUE IS "SVP".
01  OMEPSKP           PIC X(3) VALUE IS "SKP".
*   MESSAGE TYPE           FIELD: OME.SSH

```

```

01 OMESUW          PIC S9999 VALUE IS 54 USAGE IS COMP.
01 OMESUR          PIC S9999 VALUE IS 53 USAGE IS COMP.
01 OMESURA        PIC S9999 VALUE IS 21 USAGE IS COMP.
01 OMESUWR        PIC S9999 VALUE IS 55 USAGE IS COMP.
01 OMESUWRA       PIC S9999 VALUE IS 23 USAGE IS COMP.
01 OMESSW         PIC S9999 VALUE IS 50 USAGE IS COMP.
01 OMESSR         PIC S9999 VALUE IS 49 USAGE IS COMP.
01 OMESSRA        PIC S9999 VALUE IS 17 USAGE IS COMP.
01 OMESSWR        PIC S9999 VALUE IS 51 USAGE IS COMP.
01 OMESSWRA       PIC S9999 VALUE IS 19 USAGE IS COMP.
01 OMESESC        PIC S9999 VALUE IS 1  USAGE IS COMP.
01 OMESBRK        PIC S9999 VALUE IS 2  USAGE IS COMP.
01 OMESILL        PIC S9999 VALUE IS 4  USAGE IS COMP.
01 OMESASY        PIC S9999 VALUE IS 65 USAGE IS COMP.
*  COLORS 9752    FIELD: OME.COL
01 OMECSTD        PIC X(1) VALUE IS "S".
01 OMECBLU        PIC X(1) VALUE IS "B".
01 OMECCYA        PIC X(1) VALUE IS "C".
01 OMECGRE        PIC X(1) VALUE IS "G".
01 OMECYEL        PIC X(1) VALUE IS "Y".
01 OMECMAG        PIC X(1) VALUE IS "M".
01 OMECRED        PIC X(1) VALUE IS "R".
01 OMECWHI        PIC X(1) VALUE IS "W".
*  PARTNER STATUS FIELD: OME.PSTAT
01 OMEPACT        PIC X(1) VALUE IS "A".
01 OMEPHOLD       PIC X(1) VALUE IS "H".
01 OMEPINA        PIC X(1) VALUE IS "I".
01 OMEPOP        PIC X(1) VALUE IS "O".
*  PARTNER ERROR FLAG FIELD: OME.PERR
01 OMEPERRY       PIC X(1) VALUE IS "Y".
01 OMEPERRN       PIC X(1) VALUE IS "N".
*  PARTNER VTSUCB FIELD: OME.PVTCB
01 OMEPVTCY       PIC X(1) VALUE IS "Y".
01 OMEPVTCN       PIC X(1) VALUE IS "N".
*  EDIT OPTIONS  FIELD: OME.E..
MODE=
01 OMEELIN        PIC X(1) VALUE IS "L".
01 OMEEPHY        PIC X(1) VALUE IS "P".
01 OMEEFOR        PIC X(1) VALUE IS "F".
*  HARDCOPY=
01 OMEEHCYY       PIC X(1) VALUE IS "Y".
01 OMEEHCYN       PIC X(1) VALUE IS "N".
*  HOMOG.=
01 OMEEHOMY       PIC X(1) VALUE IS "Y".
01 OMEEHOMN       PIC X(1) VALUE IS "N".
*  EXTENDED-LINE=
01 OMEEEXTY       PIC X(1) VALUE IS "Y".
01 OMEEEXTN       PIC X(1) VALUE IS "N".

```



```

*   LOG. CHARS=
01  OMEELOGY          PIC X(1) VALUE IS "Y".
01  OMEELOGN          PIC X(1) VALUE IS "N".
*   BELL=
01  OMEEBELY          PIC X(1) VALUE IS "Y".
01  OMEEBELN          PIC X(1) VALUE IS "N".
*   HEADER=
01  OMEEHDRY          PIC X(1) VALUE IS "Y".
01  OMEEHDRN          PIC X(1) VALUE IS "N".
*   BACKSPACE=
01  OMEEBSY           PIC X(1) VALUE IS "Y".
01  OMEEBSN           PIC X(1) VALUE IS "N".
*   FUNCTION KEY CODE=
01  OMEEFCDY          PIC X(1) VALUE IS "Y".
01  OMEEFCDN          PIC X(1) VALUE IS "N".
*   CONFIDENTIAL DATA=
01  OMEEICDY          PIC X(1) VALUE IS "Y".
01  OMEEICDN          PIC X(1) VALUE IS "N".
*   LCASE=
01  OMEELCAY          PIC X(1) VALUE IS "Y".
01  OMEELCAN          PIC X(1) VALUE IS "N".
*   STRICT DIALOG=
01  OMEEDIAY          PIC X(1) VALUE IS "Y".
01  OMEEDIAN          PIC X(1) VALUE IS "N".
*   VTSUCB=
01  OMEEVTCY          PIC X(1) VALUE IS "Y".
01  OMEEVTCN          PIC X(1) VALUE IS "N".
*   CODE FOR KEYS      FIELD: OME..KEY
01  OMEDUE1           PIC S9999 VALUE IS 0 USAGE IS COMP.
01  OMEDUE2           PIC S9999 VALUE IS 0 USAGE IS COMP.
01  OMEK1              PIC S9999 VALUE IS 1 USAGE IS COMP.
01  OMEK2              PIC S9999 VALUE IS 2 USAGE IS COMP.
01  OMEK3              PIC S9999 VALUE IS 3 USAGE IS COMP.
01  OMEK4              PIC S9999 VALUE IS 4 USAGE IS COMP.
01  OMEK5              PIC S9999 VALUE IS 5 USAGE IS COMP.
01  OMEK6              PIC S9999 VALUE IS 6 USAGE IS COMP.
01  OMEK7              PIC S9999 VALUE IS 7 USAGE IS COMP.
01  OMEK8              PIC S9999 VALUE IS 8 USAGE IS COMP.
01  OMEK9              PIC S9999 VALUE IS 9 USAGE IS COMP.
01  OMEK10             PIC S9999 VALUE IS 10 USAGE IS COMP.
01  OMEK11             PIC S9999 VALUE IS 11 USAGE IS COMP.
01  OMEK12             PIC S9999 VALUE IS 12 USAGE IS COMP.
01  OMEK13             PIC S9999 VALUE IS 13 USAGE IS COMP.
01  OMEK14             PIC S9999 VALUE IS 14 USAGE IS COMP.
01  OMEF1              PIC S9999 VALUE IS 21 USAGE IS COMP.
01  OMEF2              PIC S9999 VALUE IS 22 USAGE IS COMP.
01  OMEF3              PIC S9999 VALUE IS 23 USAGE IS COMP.
01  OMEF4              PIC S9999 VALUE IS 24 USAGE IS COMP.

```

```

01 OMEF5          PIC S9999 VALUE IS 25 USAGE IS COMP.
01 OMEF6          PIC S9999 VALUE IS 26 USAGE IS COMP.
01 OMEF7          PIC S9999 VALUE IS 27 USAGE IS COMP.
01 OMEF8          PIC S9999 VALUE IS 28 USAGE IS COMP.
01 OMEF9          PIC S9999 VALUE IS 29 USAGE IS COMP.
01 OMEF10         PIC S9999 VALUE IS 30 USAGE IS COMP.
01 OMEF11         PIC S9999 VALUE IS 31 USAGE IS COMP.
01 OMEF12         PIC S9999 VALUE IS 32 USAGE IS COMP.
01 OMEF13         PIC S9999 VALUE IS 33 USAGE IS COMP.
01 OMEF14         PIC S9999 VALUE IS 34 USAGE IS COMP.
01 OMEF15         PIC S9999 VALUE IS 35 USAGE IS COMP.
01 OMEF16         PIC S9999 VALUE IS 36 USAGE IS COMP.
01 OMEF17         PIC S9999 VALUE IS 37 USAGE IS COMP.
01 OMEF18         PIC S9999 VALUE IS 38 USAGE IS COMP.
01 OMEF19         PIC S9999 VALUE IS 39 USAGE IS COMP.
01 OMEF20         PIC S9999 VALUE IS 40 USAGE IS COMP.
01 OMEF21         PIC S9999 VALUE IS 41 USAGE IS COMP.
01 OMEF22         PIC S9999 VALUE IS 42 USAGE IS COMP.
01 OMEF23         PIC S9999 VALUE IS 43 USAGE IS COMP.
01 OMEF24         PIC S9999 VALUE IS 44 USAGE IS COMP.
*   UTM DIALOG STATUS FIELD: OMEUSTAT
01 OMEUDSIG      PIC X(1) VALUE IS "S".
01 OMEUDOV      PIC X(1) VALUE IS "O".
01 OMEUDIV      PIC X(1) VALUE IS "V".
01 OMEUDWT      PIC X(1) VALUE IS "T".
01 OMEUDINV     PIC X(1) VALUE IS "X".
*   SEND PRIORITY FIELD: OMEUSEND
01 OMEUSUTM     PIC X(1) VALUE IS "U".
01 OMEUSTRM     PIC X(1) VALUE IS "D".
*   UTM MESSAGE TYPE FIELD: OMEUOPC
01 OMEUMUSR     PIC X(1) VALUE IS "M".
01 OMEUMBCT     PIC X(1) VALUE IS "B".
01 OMEUMRES     PIC X(1) VALUE IS "R".
01 OMEUMUBD     PIC X(1) VALUE IS "U".
01 OMEUMINV     PIC X(1) VALUE IS "X".
*   EXTENDED CODE OF MSG FIELD: OMECODE
01 OMEUCOD7     PIC X(1) VALUE IS "7".
01 OMEUCOD8     PIC X(1) VALUE IS "8".
*   HCY STATUS FIELD: OMEHSTAT
01 OMEHACT      PIC X(1) VALUE IS "A".
01 OMEHHOLD     PIC X(1) VALUE IS "H".
01 OMEHINA      PIC X(1) VALUE IS "I".
01 OMEHOPN      PIC X(1) VALUE IS "O".

```

14.6.4 Resolution of recursive calls

Whereas Assembler permits recursive calls, COBOL destroys the register contents if recursive calls are used. In particular, the recursions which occur in the case of OMNIS return information for partner calls (operation codes *OMEMTP*, *OMEMPT* and *OMECOM*) must be avoided.

The following figure shows, with the aid of an *OMECOM* call (*OPNCON* command) as an example, how such a recursion is generated:

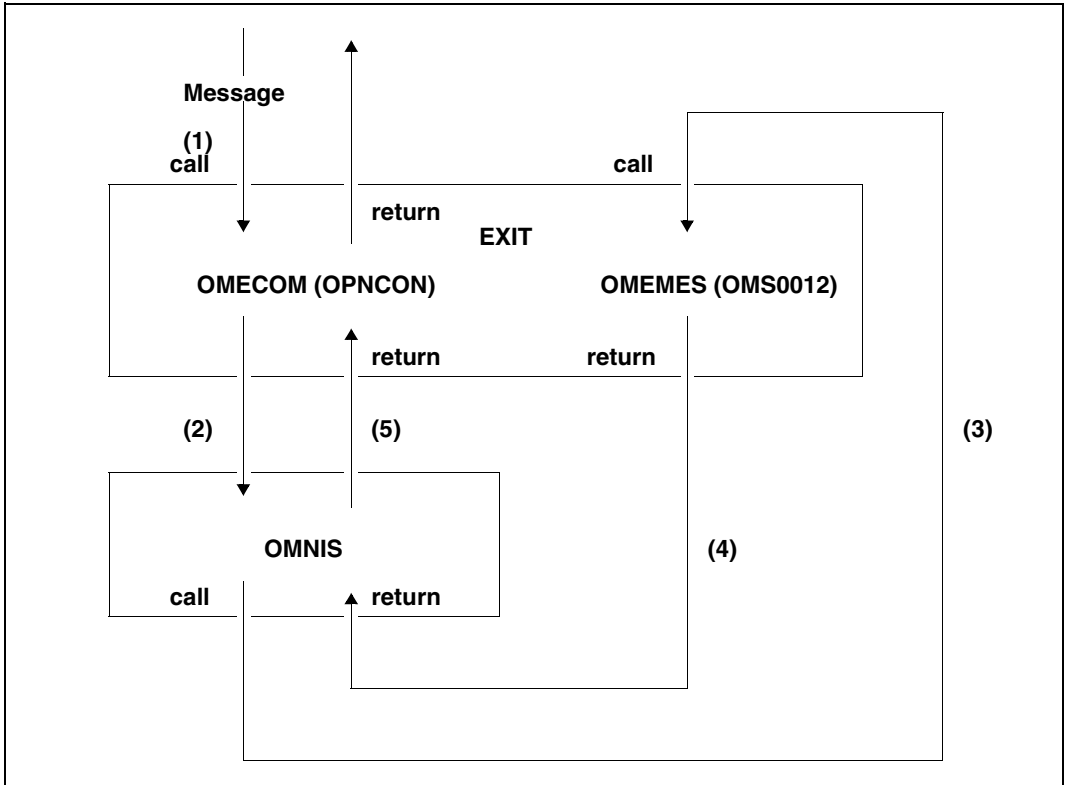


Figure 43: Recursive calls in exit routines

The numbers represent the various processing steps. The exit routine is called with a message (1) and, in turn, calls OMNIS with *OMECOM* (2). The exit routine is then called again with *OMEMES* for the OMNIS system message OMS0012 (3). After the return, the program branches back via OMNIS to the exit routine (5). Due to the second call of the exit routine (call (3)), the exit routine may overwrite data which is accessed later, after return (5).

With the aid of an Assembler routine, it is possible to buffer the return information of the critical calls and to output this information only when the related calls have been processed. This can be done as shown below:

Load OMNIS parameter list			
OMNIS message		OMNIS command or message TERM → PTNR or message PTNR → TERM	Operation code otherwise
VAREXIT = AKTIV ?		VAREXIT = AKTIV	not interpreted
Yes	No	Exit calls OMNIS (OMSCALL)	
Write OMNIS message to buffer	Process OMNIS message	OMNIS message in buffer ?	
		Yes No	
		Process OMNIS message	VAREXIT=INAKTIV

14.7 Messages from exit routines

It is recommended that user-generated messages be output in the same format as standard messages (i.e. OMSnnnn text). Message numbers OMS7000 through OMS7999 are reserved for this purpose (i.e. users may select numbers from this range for their messages).

14.8 Preparations for use

This section provides information on the steps required to prepare an exit routine for use (assembling/compiling, linking, loading, activating and testing).

Due to the conversion from OMNIS to H-Assembler, the CRTE modules (*IT0...*) are permanently linked and must be excluded from the linking operation for exit routines that are also based on CRTE.

Assembling and linking (Assembler)

The *SYSLIB.OMNIS.085* and *STSOS.MACROLIB* macro libraries are required for the assembly of exit routines in Assembler. Once a routine has been assembled without error, the generated module must be placed in the module library defined in the start parameters.

If the exit routine is written in Columbus-Assembler notation, the linker statements *EXCLUDE \$NUCENTR* and *EXCLUDE \$NUCEXIT* must be included.

Compiling and linking (COBOL)

If the exit routine is written in COBOL, it must be linked with the COBOL runtime system to create a prelinked module. The COBOL copy elements *OMECPAR* and *OMECEQU* are located in the *SYSLIB.OMNIS.085* library. Once a routine has been compiled without error, the generated module must be placed in the module library defined in the start parameters.

Linking several modules of an exit routine to create a prelinked module

The following example shows, with the aid of a procedure, how several modules of an exit routine are linked to create a prelinked module.

If an exit routine has been written in Columbus-Assembler, the following BINDER statement must be entered:

```
//RESOLVE-BY-AUTOLINK LIB=lib,SYMBOL-NAME=(-OMSCALL,-$NUCENTR,-$NUCEXIT)

/BEGIN-PROCEDURE LOGGING=A
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-PROGRAM FROM-FILE=$BINDER
//START-LLM-CREATION INTERNAL-NAME=moduleName
//INCLUDE-MODULES LIB=lib,ELEM=moduleName1
//INCLUDE-MODULES LIB=lib,ELEM=moduleName2
[ ... ]
//MODIFY-SYMBOL-ATTRIBUTES SYMBOL-NAME=moduleName1,ALIGNMENT=*PAGE
//MODIFY-SYMBOL-ATTRIBUTES SYMBOL-NAME=moduleName2,ALIGNMENT=*PAGE
[ ... ]
//RESOLVE-BY-AUTOLINK LIB=(lib2,lib1)
```

```
//MODIFY-MAP-DEFAULTS LOG-STRUC=*NO,PHYS-STRUC=*NO,
//
//          PROG-MAP=*PAR(INV-XREF-LIST=*ALL),UNRESOLVED=*NO
//SAVE-LLM LIB=lib,elem=modulname
//END
//ASSIGN-SYSDTA TO-FILE=*PRIMARY
//END-PROCEDURE
```

where:

modulename	Name of the module to which linking is to take place
modulename1, modulename2 ..	Names of the individual exit routine modules
lib1, lib2 ..	Libraries from which elements are needed for linking, e.g. runtime systems

Loading and activating

Exit routine modules are loaded when an exit group is defined:

An exit group defined with *DECLARE-EXIT* is loaded the next time OMNIS is started. If the definition is made with the *EXIT* command, the operand *LOAD* can be used to specify whether, when the command is executed, all modules or only those modules which are not yet loaded are to be loaded.

The modules of an exit group are activated by assigning the exit group globally, terminal-specifically or partner-specifically with the command *SET*, *OPTION* or *PARAM* (operand *EXIT=eac*), respectively.

An exit routine remains loaded even if it is deactivated again (with *SET*, *OPTION* or *PARAM* with *EXIT=NO*). Renewed activation simply causes the exit group to be assigned again.

14.8.1 Testing

For testing exit routines with the aid of AID, OMNIS can be loaded in the dialog with *SYMTEST=ALL*:

```
/START=PROGRAM FROM=FILE=*M(SYSLNK.OMNIS.085,OMNIS,RUN=MODE=ADVANCED),
TEST=OPTIONS=AID
% BLS0523 ELEMENT 'OMNIS', VERSION '085' FROM LIBRARY ':Z:$OMSE.
SYSLNK.OMNIS.083' IN PROCESSING
% BLS0524 LLM 'OMNIS', VERSION '085' OF '2002-03-18' LOADED
*
start parameters
*
END
```

OMNIS itself is linked without AID (message BLS0600); the operand *TEST=OPTIONS=AID* causes the ESD records of the exit routine to be included later when dynamic loading is performed with *LINK*.

The exit routine can be loaded from a transaction-mode terminal or from the startup file. After loading, all addresses in the exit routine can be accessed symbolically.

For the purpose of setting checkpoints, OMNIS can be interrupted at any time with [K2](#).

For testing purposes, the Real Timer can be deactivated using the OMNIS command *SET RTIMER=OFF*.

Assembler

The ASSEMBH-RTS uses the standard stack administration where R13 points to the first stack entry. Therefore, R13 must not be destroyed. R14 must also be saved for returning to OMNIS. R1 must point to the parameter list *OMEPARL* (see [page 257](#)). All the other registers may be used freely in exit routines.

COBOL

Testing exit routines written in COBOL differs from the testing of exit routines written in Assembler only to the extent that there is no stack administration, and all the data is located in the COBOL program itself.

The problem of recursive calls has already been described on [page 299](#), which also indicates a solution for dealing with recursions.

Error analysis

If a serious error occurs in an exit routine, OMNIS will usually abort. It is impossible to tell whether the cause of the error is to be found in OMNIS or in the exit routine. With the help of the OMNIS routine ERROR-STXIT, a dump can be generated and the cause of the error and the address at which the error occurred output to SYSLST.

The user must now determine whether the error occurred in OMNIS or in the exit routine. OMNIS is loaded at the beginning of the user address space.

As additional information for a dump, an insert in message IDA0N51 shows the OMNIS message number and the address at which the dump was requested.

14.9 Examples

In this section, an example is used to describe the planning ([page 305](#)) and the implementation of exit routines in detail. A suggested solution for the task is shown in COBOL notation “[COBOL examples](#)” on [page 314](#). A "blank example" is also provided in both notations as an aid to implementation.

14.9.1 Designing an exit routine

First, the requirements need to be determined.

The following functions must be provided by the exit routine in question (*LOGONEX*):

1. After a connection to a terminal has been established, a connection to the application \$DIALOG with the partner address code *PTNI* is to be set up automatically.
2. The required password is to be requested explicitly by the exit routine in blanking mode.
3. Once the connection has been successfully established, *LOGON* under the ID OMNIS is to be effected.
4. EDT is to be called after successful *LOGON*.
5. A dialog is to be conducted with EDT.

Message traffic between \$DIALOG, OMNIS and the terminal could take place as follows:

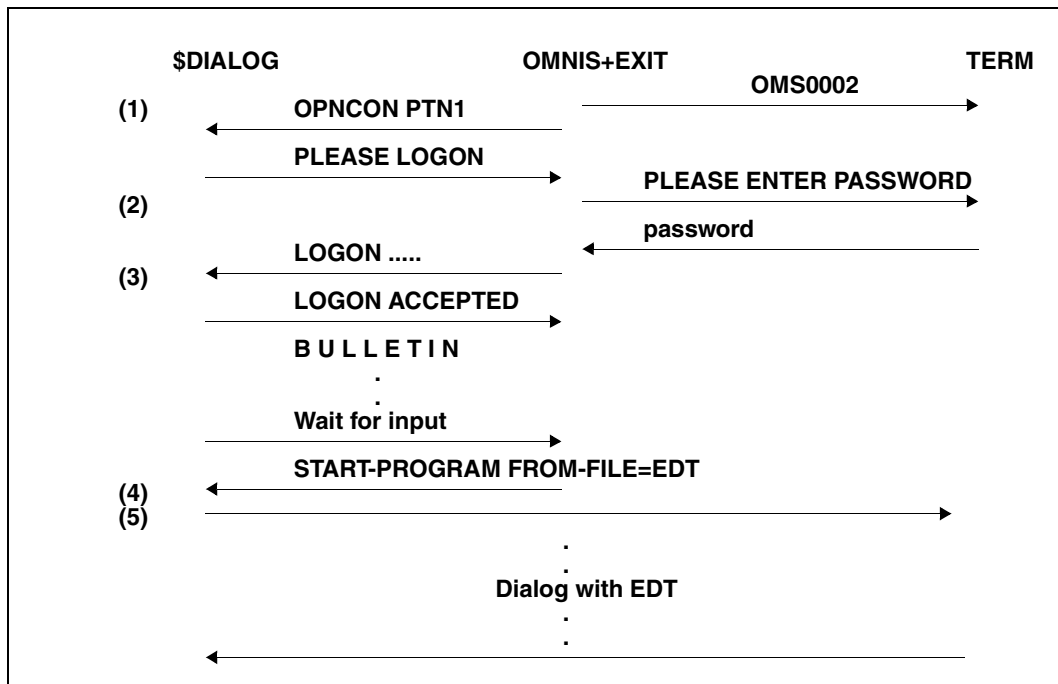


Figure 44: Message traffic between \$DIALOG, OMNIS and a terminal with the aid of an exit routine

The following structure chart shows a possible solution:

Load parameter area and TERM work area									
Which event is calling exit routine?									
OMNIS system message is to be output on the terminal		Message from a partner is to be output on the terminal				Enter command to OMNIS		Initializa-tion	other-wise
Terminal signed on to OMNIS		TIAM waiting for input ? (system read)				Next step password interrogation		No Initializa-tion required	Send "no further actions" to OMNIS
Yes ? No	Yes	?		No	Yes	? No			
Set up a TIAM connection	Issue OMNIS message without checking	Next step LOGON		STATUS of exit routine is "LOGON" or "EXEC"		Accept the password Set partner "PTN1" Send LOGON command to partner		Issue OMNIS command without checking	
		Yes	?		No	Yes	? No		
Send OPNCON command to OMNIS	LOGON command requires a password	Next step EXEC call		EXIT still active, messages are suppressed		Pass on message to terminal		Password longer than 8 charact. ?	
		Yes	?		No			Yes	No
Place OPCODE, command and length in transfer area	Send message "PLEASE ENTER PASSWORD" to terminal	Call the EDT		Send input prompt to terminal				Pass-word length = 8 chars	Pass-word length OK
Note next step "LOGON" in TERM work area	Note next step "QUERY PASSWORD"	Send EXEC command to partner		Note next step "EXIT EXAMPLE FINISHED"					

Blank example in Columbus-Assembler:

```

*****
*   BLANK EXAMPLE OF COLUMBUS ASSEMBLER   *
*****
      PRINT NOGEN,NOREF
LEERCASS START 0
      TITLE '* TEST USER ROUTINE LEERCASS *'
      SPACE ,
+LEERCASS+
| @ENTR
| TYP=E
| SPACE ,
| @DATA CLASS=B,BASE=R1,DSECT=OMEPARL
| @DATA CLASS=B,BASE=R2,DSECT=OMEPAR
| SPACE ,
| L      R2,OMEPARO
|   +-@IF
|   |   NE
|   |   * NO INITIALIZATION REQUIRED ?
|   |   CLC  OMEOOPC,=AL2(OMEINIT)
|   |   -@THEN
|   |   * PASS ON ALL MESSAGES UNCHANGED
|   |   @PASS EXTNAME=OMSCALL
|   +-@BEND
|   @EXIT
+--@END
      SPACE ,
*****
*   DSECTS AND EQUATES   *
*****
      SPACE ,
      SPACE ,
      OMEPARL MF=D
      OMEPAR MF=D
      OMEEQU MF=D
      END ,

```

The **exit routine LOGONEX** could, for example, be implemented in **Columbus-Assembler** as follows:

```

LOGONEX  START 0
          PRINT NOGEN,NOREF
          TITLE '*      LOGONEX      *'
          SPACE ,
*****
*
* EXIT-ROUTINE LOGONEX
*
*****
          SPACE ,
+LOGONEX+
|  -@ENTR|
|      TYP=E
|      SPACE ,
|      @DATA CLASS=B,BASE=R1,DSECT=OMEPARL
|      @DATA CLASS=B,BASE=R2,DSECT=OMEPAR
|      @DATA CLASS=B,BASE=R8,DSECT=DSTWORK
|      SPACE ,
|      +-@BEGIN
|      |      * LOAD PARAMETER AREA AND TERM WORK AREA
|      |      L      R2,OMEPARO
|      |      L      R8,OMEPART
|      +-@BEND
|
|      +-@CAS2
|      |      OMEOOPC
|      |      * WHAT EVENT CALLED THE EXIT ROUTINE?
|      -@OF-=AL2(OMEMES)
|      |      * OMNIS MESSAGE IS ISSUED TO TERMINAL
|      +-@IF
|      |      |      EQ
|      |      |      * TERMINAL HAS SIGNED ON TO OMNIS (OMS0002)
|      |      |      L      R7,OMEPARM
|      |      |      CLC  1(7,R7),=C'OMS0002'
|      |      -@THEN
|      |      |      * A TIAM CONNECTION IS SET UP
|      |      |      * OPNCON COMMAND TO OMNIS IS SENT
|      |      |      * PLACE OPCODE, COMMAND AND LENGTH IN TRANSFER
|      |      |      * AREA
|      |      |      * NOTE NEXT STEP "LOGON" IN TERM WORK AREA
|      |      MVC  OMEOOPC,=AL2(OMECOM)
|      |      L      R7,OMEPARM
|      |      MVC  0(OPNCONL,R7),OPNCON
|      |      LA   R3,OPNCONL
|      |      STH  R3,OMEOMLEN

```

```

      @PASS EXTNAME=OMSCALL
      MVI  DSTFLAG,DSTFLAGL
-@ELSE
      * ISSUE OMNIS MESSAGE WITHOUT INSPECTION
      @PASS EXTNAME=OMSCALL
+-@BEND
-@OF==AL2(OMEMPT)
* MESSAGE FROM PARTNER IS ISSUED TO TERMINAL
-@IF
      EQ
      * TIAM IS WAITING FOR INPUT (SYSTEM READ)
      CLI  OMEPSSH+1,OMESSR
-@THEN
      +-@IF
          EQ
          * IS NEXT STEP LOGON?
          CLI  DSTFLAG,DSTFLAGL
      -@THEN
          * LOGON COMMAND REQUIRES A PASSWORD
          * SEND MESSAGE "PLEASE ENTER PASSWORD" TO TERMINAL
          * NOTE NEXT STEP "QUERY PASSWORD"
          MVC  OMEOOPC,=AL2(OMEMES)
          L    R7,OME Parm
          MVC  0(MESSAGEL,R7),MESSAGE
          LA   R3,MESSAGEL
          STH  R3,OMEOMLEN
          MVI  OMEEICD,OME EICDY
          MVI  OMEEDIA,OME EDIAY
          @PASS EXTNAME=OMSCALL
          MVI  DSTFLAG,DSTFLAGP
      -@ELSE
          +-@IF
              EQ
              * IS NEXT STEP EXEC CALL?
              CLI  DSTFLAG,DSTFLAGE
          -@THEN
              * EDT IS CALLED
              * EXEC COMMAND TO PARTNER IS SENT
              * NOTE NEXT STEP "EXIT EXAMPLE"
              * FINISHED"
              MVC  OMEOOPC,=AL2(OMEMTP)
              L    R7,OME Parm
              MVC  0(EXECL,R7),EXEC
              LA   R3,EXECL
              STH  R3,OMEOMLEN
              @PASS EXTNAME=OMSCALL
              MVI  DSTFLAG,DSTFLAGP
          -@ELSE

```

```

|         |         |         |         |         * PASS ON INPUT PROMPT TO TERMINAL
|         |         |         |         |         @PASS EXTNAME=OMSCALL
|         |         |         |         |         +--@BEND
|         |         |         |         |         +--@BEND
|         |         |         |         |         -@ELSE
|         |         |         |         |         * PARTNER OUTPUT MESSAGE
|         |         |         |         |         +--@IF
|         |         |         |         |         |         EQ
|         |         |         |         |         |         * IS STATUS OF EXIT ROUTINE "LOGON" OR
|         |         |         |         |         |         * "EXEC"?
|         |         |         |         |         |         CLI  DSTFLAG,DSTFLAGL
|         |         |         |         |         |         @OR EQ
|         |         |         |         |         |         CLI  DSTFLAG,DSTFLAGE
|         |         |         |         |         |         -@THEN
|         |         |         |         |         |         * EXIT STILL ACTIVE: MESSAGES ARE
|         |         |         |         |         |         * SUPPRESSED
|         |         |         |         |         |         -@ELSE
|         |         |         |         |         |         * PASS ON MESSAGE TO TERMINAL
|         |         |         |         |         |         @PASS EXTNAME=OMSCALL
|         |         |         |         |         |         +--@BEND
|         |         |         |         |         +--@BEND
|         |         |         |         |         -@OF--=AL2(OMEOMCOM)
|         |         |         |         |         * COMMAND FOR OMNIS HAS BEEN ENTERED
|         |         |         |         |         +--@IF
|         |         |         |         |         |         EQ
|         |         |         |         |         |         * IS NEXT STEP PASSWORD QUERY?
|         |         |         |         |         |         CLI  DSTFLAG,DSTFLAGP
|         |         |         |         |         |         -@THEN
|         |         |         |         |         |         * PASSWORD IS ACCEPTED
|         |         |         |         |         |         * SET PARTNER "PTN1"
|         |         |         |         |         |         * SEND LOGON COMMAND TO PARTNER
|         |         |         |         |         |         * NOTE NEXT STEP "EXEC"
|         |         |         |         |         |         +--@IF
|         |         |         |         |         |         |         GT
|         |         |         |         |         |         |         * IS PASSWORD LONGER THAN 8 CHARACTERS ?
|         |         |         |         |         |         |         LH   R3,OMEOMLEN
|         |         |         |         |         |         |         CH   R3,=H'8'
|         |         |         |         |         |         |         -@THEN
|         |         |         |         |         |         |         * PASSWORD LENGTH = 8 CHARACTERS
|         |         |         |         |         |         |         LA   R3,8
|         |         |         |         |         |         +--@BEND
|         |         |         |         |         |         BCTR  R3,R0
|         |         |         |         |         |         EX   R3,MVCEX
|         |         |         |         |         |         MVC   OMEOOPC,=AL2(OMESLPA)
|         |         |         |         |         |         MVC   OMEPPAC,=C'PTN1'
|         |         |         |         |         |         MVC   OMEPPACL,=H'4'
|         |         |         |         |         |         @PASS EXTNAME=OMSCALL
|         |         |         |         |         |         MVC   OMEOOPC,=AL2(OMEMTP)

```

```

|      |      |      LA   R3,LOGON
|      |      |      ST   R3,OME Parm
|      |      |      LA   R3,LOGONL
|      |      |      STH  R3,OMEOMLEN
|      |      |      @PASS EXTNAME=OMSCALL
|      |      |      MVC  LOGONP,=CL8' '
|      |      |      MVI  DSTFLAG,DSTFLAGE
|      |      |      -@ELSE
|      |      |      *   ISSUE OMNIS COMMAND WITHOUT INSPECTION
|      |      |      @PASS EXTNAME=OMSCALL
|      |      |      +-@BEND
|      |      |      -@OF--=AL2(OMEINIT)
|      |      |      *   INITIALIZATION
|      |      |      *   NO INITIALIZATION REQUIRED
|      |      |      -@OFRE
|      |      |      *   DO NOTHING - MOVE ON TO OMNIS
|      |      |      @PASS EXTNAME=OMSCALL
|      |      |      +-@BEND
|      |      |      @EXIT
|      |      |      MVCEX   MVC   LOGONP(0),0(R7)
+-@END

      SPACE ,
*****
*
*   DECLARATIONS, DSECTS AND EQUATES
*
*****

      SPACE ,
      PRINT NOGEN
OPNCON   DC   C'OPNCON PTN1'
OPNCONL  EQU   *-OPNCON
MESSAGE  DC   C'PLEASE ENTER LOGON PASSWORD: '
MESSAGEL EQU   *-MESSAGE
LOGON    DC   C'LOGON OMNIS,1,C''
LOGONP   DC   CL8' '
          DC   C''''
LOGONL   EQU   *-LOGON
EXEC     DC   C'EXEC $EDT'
EXECL    EQU   *-EXEC

      SPACE
      SPACE
DSTWORK  DSECT
DSTFLAG  DS   CL1
DSTFLAGL EQU  C'L'
DSTFLAGP EQU  C'P'
DSTFLAGE EQU  C'E'
DSTFLAGF EQU  C'F'

      SPACE

```



```
SPACE  
OMEPARL MF=D  
OMEPAR MF=D  
OMEEQU MF=D  
END ,
```

14.9.2 COBOL examples

The following "blank example" can be used for embedding exit routines in OMNIS.

First, the **blank example in COBOL notation**:

```

*****
*      BLANK EXAMPLE OF COBOL EXIT      *
*****
IDENTIFICATION DIVISION.
*****
PROGRAM-ID.  LEERCOB.
AUTHOR.     OMNISQS.

ENVIRONMENT DIVISION.
*****

DATA DIVISION.
*****

WORKING-STORAGE SECTION.

COPY OMECEQU.

LINKAGE SECTION.

COPY OMECPAR.

01  OMEMESSAGE          PIC X(4096).

01  OMEDSTWORK          PIC X(256).

01  OMEPTNWORK          PIC X(256).

COPY VTSUCB

PROCEDURE DIVISION USING OMEPAR, OMEMESSAGE,
                        OMEDSTWORK, OMEPTNWORK, VTSUCB.

MAIN SECTION.

MAIN.

      IF OMEOOPC NOT EQUAL OMEINIT
*      NO INITIALIZATION ?
      THEN

```

```
*      PASS ON ALL MESSAGES UNCHANGED
      CALL "OMSCALL" USING OMEPAR, OMEMESSAGE,
          OMEDSTWORK, OMEPTNWORK, VTSUCB.

      EXIT PROGRAM.
```

The blank example in Columbus-Cobol:

```

*****
*   BLANK EXAMPLE OF COLUMBUS COBOL   *
*****
      IDENTIFICATION DIVISION.
*****
      PROGRAM-ID.   LEERCOC.
      AUTHOR.      OMNISQS.
*
      ENVIRONMENT DIVISION.
*****
*
      DATA DIVISION.
*****
*
      WORKING-STORAGE SECTION.
*
      COPY OMECEQU.
*
      LINKAGE SECTION.
*
      COPY OMECPAR.
      01 OMEMESSAGE          PIC X(4096).
      01 OMEDSTWORK          PIC X(256).
      01 OMEPTNWORK          PIC X(256).
      COPY VTSUCB
*
*
*
+LEERCOC+
  -@ENTRY :TYP=E:
    USING OMEPAR OMEMESSAGE OMEDSTWORK OMEPTNWORK VTSUCB.
    *
    +-@IF
      OMEOOPC NOT EQUAL OMEINIT
      *          NO INITIALIZATION ?
    -@THEN
      *          FORWARD ALL MESSAGES UNCHANGED
      @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                                OMEDSTWORK OMEPTNWORK VTSUCB.
    +-@BEND

    @EXIT
  +-@END

```

The **exit routine LOGONEX** presented on [page 305](#) could, for example, be implemented as follows in **COBOL**:

```

*****
*  EXIT ROUTINE LOGONEX IN COBOL                                     *
*****
  IDENTIFICATION DIVISION.
*****
  PROGRAM-ID.  LOGONEX.
  AUTHOR.     OMNISQS.

  ENVIRONMENT DIVISION.
*****

  DATA DIVISION.
*****

  WORKING-STORAGE SECTION.

  COPY OMECEQU.

  01  PASSWORD.
      05  PASSWORD-E OCCURS 8 TIMES INDEXED BY INDEX-P PIC X.

  LINKAGE SECTION.

  COPY OMECPAR.

  01  OMEMESSAGE.
      05  MESSAGE1          PIC X(4096).
      05  MSG-OMS2-RED REDEFINES MESSAGE1.
          10  FILLER          PIC X(1).
          10  MSG-OMS2        PIC X(7).
          10  FILLER          PIC X(4088).
      05  OPNCON-RED REDEFINES MESSAGE1.
          10  OPNCON          PIC X(11).
          10  FILLER          PIC X(4085).
      05  LOG-PAS-RED REDEFINES MESSAGE1.
          10  LOG-PAS         PIC X(30).
          10  FILLER          PIC X(4066).
      05  LOGON-RED REDEFINES MESSAGE1.
          10  LOGON           PIC X(13).
          10  LOGONP          PIC X(8).
          10  LOGONE          PIC X(1).
          10  FILLER          PIC X(4074).
      05  MSG-PASS REDEFINES MESSAGE1.
          10  MSG-PASS-E      OCCURS 8 TIMES INDEXED BY INDEX-M PIC X.
          10  FILLER          PIC X(4088).

```

```

05 EXEC-RED REDEFINES MESSAGE1.
    10 EXEC          PIC X(9).
    10 FILLER        PIC X(4087).

01 OMEDSTWORK.
    05 DSTWORK          PIC X(256).
    05 DSTWORK-RED REDEFINES DSTWORK.
        10 DSTFLAG      PIC X.
            88 DSTFLAGL VALUE "L".
            88 DSTFLAGP VALUE "P".
            88 DSTFLAGE VALUE "E".
            88 DSTFLAGF VALUE "F".
        10 FILLER        PIC X(255).

01 OMEPTNWORK          PIC X(256).

PROCEDURE DIVISION USING OMEPAR, OMEMESSAGE
                        OMEDSTWORK, OMEPTNWORK, VTSUCB.

MAIN SECTION.

DETERMINE-PROCESSING.
* WHAT EVENT CALLED THE EXIT ROUTINE ?

    IF OMEOOPC = OMEMES
*     OMNIS MESSAGE IS ISSUED TO TERMINAL
    THEN
        IF MSG-OMS2 = "OMS0002"
*     THE TERMINAL HAS SIGNED ON TO OMNIS (OMS0002)
        THEN
*     A TIAM CONNECTION IS SET UP
*     OPNCON COMMAND TO OMNIS IS SENT
*     PLACE OPCODE, COMMAND AND LENGTH IN TRANSFER AREA
*     NOTE NEXT STEP "LOGON" IN TERM WORK AREA
        MOVE OMECOM TO OMEOOPC,
        MOVE "OPNCON PTN1" TO OPNCON,
        MOVE 11 TO OMEOMLEN,
        CALL "OMSCALL" USING OMEPAR, OMEMESSAGE,
                        OMEDSTWORK, OMEPTNWORK, VTSUCB.

        SET DSTFLAGL TO TRUE,
    ELSE
*     ISSUE OMNIS MESSAGE WITHOUT INSPECTION
        CALL "OMSCALL" USING OMEPAR, OMEMESSAGE
                        OMEDSTWORK, OMEPTNWORK, VTSUCB.

    ELSE
        IF OMEOOPC = OMEMPT
*     MESSAGE FROM PARTNER IS ISSUED TO TERMINAL

```

```

THEN
  IF OMEPSSH = OMESSR
*   TIAM IS WAITING FOR INPUT (SYSTEM READ)
  THEN
    IF DSTFLAGL
*     IS NEXT STEP LOGON?
    THEN
*     LOGON COMMAND REQUIRES A PASSWORD
*     SEND MESSAGE "PLEASE ENTER PASSWORD" TO TERMINAL
*     NOTE NEXT STEP "QUERY PASSWORD"
      MOVE OMEMES TO OMEOOPC,
      MOVE "PLEASE ENTER LOGON PASSWORD:" TO LOG-PAS,
      MOVE 30 TO OMEOMLEN,
      MOVE OMEEICDY TO OMEEICD,
      MOVE OMEEDIAY TO OMEEDIA,
      CALL "OMSCALL" USING OMEPAR, OMEMESSAGE,
        OMEDSTWORK, OMEPTNWORK, VTSUCB.
      SET DSTFLAGP TO TRUE,
    ELSE
      IF DSTFLAGE
*     IS NEXT STEP AN EXEC CALL?
      THEN
*     EDT IS CALLED
*     EXEC COMMAND TO PARTNER IS SENT
*     NOTE NEXT STEP "EXIT EXAMPLE FINISHED"
        MOVE OMEMTP TO OMEOOPC,
        MOVE "EXEC $EDT" TO EXEC,
        MOVE 9 TO OMEOMLEN,
        CALL "OMSCALL" USING OMEPAR, OMEMESSAGE,
          OMEDSTWORK, OMEPTNWORK, VTSUCB.
        SET DSTFLAGF TO TRUE,
      ELSE
*     PASS ON INPUT PROMPT TO TERMINAL
        CALL "OMSCALL" USING OMEPAR, OMEMESSAGE,
          OMEDSTWORK, OMEPTNWORK, VTSUCB.
    ELSE
*     PARTNER OUTPUT MESSAGE
      IF DSTFLAGL OR DSTFLAGE
*     IS NEXT STEP LOGON OR EXEC ?
      THEN
*     EXIT ROUTINE STILL ACTIVE
*     SUPPRESS MESSAGES
        NEXT SENTENCE,
      ELSE
*     PASS ON MESSAGE TO TERMINAL
        CALL "OMSCALL" USING OMEPAR, OMEMESSAGE
          OMEDSTWORK, OMEPTNWORK, VTSUCB.

```

```

ELSE
  IF OMEOOPC = OMECOM
  *   COMMAND FOR OMNIS HAS BEEN ENTERED
  THEN
    IF DSTFLAGP
  *   IS NEXT STEP PASSWORD QUERY?
    THEN
  *   PASSWORD IS ACCEPTED
  *   SET PARTNER "PTN1"
  *   SEND LOGON COMMAND TO PARTNER
  *   NOTE NEXT STEP "EXEC"
    MOVE OMESLPA TO OMEOOPC
    MOVE "PTN1" TO OMEPPAC
    MOVE 4 TO OMEPPACL
    CALL "OMSCALL" USING OMEPAR, OMEMESSAGE
                                OMEDSTWORK, OMEPTNWORK, VTSUCB.
    MOVE OMEMTP TO OMEOOPC
    MOVE SPACES TO PASSWORD
    SET INDEX-M TO 1
    SET INDEX-P TO 1
    PERFORM WORD-SETTING UNTIL INDEX-M > OMEOMLEN
    MOVE "LOGON OMNIS,1,C" TO LOGON
    MOVE PASSWORD TO LOGONP
    MOVE "" TO LOGONE
    MOVE 22 TO OMEOMLEN
    CALL "OMSCALL" USING OMEPAR, OMEMESSAGE
                                OMEDSTWORK, OMEPTNWORK, VTSUCB.
    MOVE SPACES TO PASSWORD
    SET DSTFLAGE TO TRUE
  ELSE
  *   ISSUE OMNIS COMMAND WITHOUT INSPECTION
    CALL "OMSCALL" USING OMEPAR, OMEMESSAGE
                                OMEDSTWORK, OMEPTNWORK, VTSUCB.

  ELSE
  *   NOTHING TO DO - MOVE ON TO OMNIS
    CALL "OMSCALL" USING OMEPAR, OMEMESSAGE.
                                OMEDSTWORK, OMEPTNWORK, VTSUCB.

EXIT PROGRAM.

WORD-SETTING.
MOVE MSG-PASS-E(INDEX-M) TO PASSWORD-E(INDEX-P)
SET INDEX-M UP BY 1
SET INDEX-P UP BY 1.

```


The **exit routine LOGONEX** could, for example, be implemented as follows in **Columbus-Cobol**:

```

*****
*   EXIT ROUTINE LOGONEX                               *
*****
      IDENTIFICATION DIVISION.
*****
      PROGRAM-ID.    LOGONEX.
      AUTHOR.       OMNIS-DEVELOPER.
*
      ENVIRONMENT DIVISION.
*****
*
      DATA DIVISION.
*****
*
      WORKING-STORAGE SECTION.
*
      COPY OMECEQU.
*
      01  PASSWORD.
          05  PASSWORD-E OCCURS 8 TIMES INDEXED BY INDEX-P PIC X.
*
      LINKAGE SECTION.
*
      COPY OMECPAR.
*
*
*
      01  OMEMESSAGE.
          05  MESSAGE1          PIC X(4096).
          05  MSG-OMS2-RED REDEFINES MESSAGE1.
              10  FILLER          PIC X(1).
              10  MSG-OMS2        PIC X(7).
              10  FILLER          PIC X(4088).
          05  OPNCON-RED REDEFINES MESSAGE1.
              10  OPNCON          PIC X(11).
              10  FILLER          PIC X(4085).
          05  LOG-PAS-RED REDEFINES MESSAGE1.
              10  LOG-PAS         PIC X(30).
              10  FILLER          PIC X(4066).
          05  LOGON-RED REDEFINES MESSAGE1.
              10  LOGON           PIC X(13).
              10  LOGONP          PIC X(8).
              10  LOGONE          PIC X(1).
              10  FILLER          PIC X(4074).
          05  MSG-PASS REDEFINES MESSAGE1.

```

```

        10 MSG-PASS-E OCCURS 8 TIMES INDEXED BY INDEX-M PIC X.
        10 FILLER PIC X(4088).
05 EXEC-RED REDEFINES MESSAGE1.
        10 EXEC PIC X(9).
        10 FILLER PIC X(4087).
*
01 OMEDSTWORK.
05 DSTWORK PIC X(256).
05 DSTWORK-RED REDEFINES DSTWORK.
    10 DSTFLAG PIC X.
        88 DSTFLAGL VALUE "L".
        88 DSTFLAGP VALUE "P".
        88 DSTFLAGE VALUE "E".
        88 DSTFLAGF VALUE "F".
    10 FILLER PIC X(255).
*
01 OMEPTNWORK PIC X(256).
*
+LOGONEX+
-@ENTRY :TYP=E:
    USING OMEPAR OMEMESSAGE OMEDSTWORK OMEPTNWORK VTSUCB.
*
+CASE+
-@CASE
    * WHAT EVENT CALLED THE EXIT ROUTINE?
    OFMES
-@OF-OME00PC = OMEMES
    * OMNIS MESSAGE IS ISSUED TO THE TERMINAL
+-@IF
    MSG-OMS2 = "OMS0002"
    * THE TERMINAL HAS SIGNED ON TO OMNIS (OMS0002)
-@THEN
    * A TIAM CONNECTION IS SET UP
    * OPNCON COMMAND TO OMNIS IS SENT
    * PLACE OPCODE, COMMAND AND LENGTH IN TRANSFER AREA
    * NOTE NEXT STEP "LOGON" IN TERM WORK AREA
    MOVE OMECOM TO OME00PC.
    MOVE "OPNCON PTN1" TO OPNCON.
    MOVE 11 TO OMEOMLEN.
    @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                                OMEDSTWORK OMEPTNWORK VTSUCB.
    SET DSTFLAGL TO TRUE.
-@ELSE
    * ISSUE OMNIS MESSAGE WITHOUT INSPECTION
    @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE.
                                OMEDSTWORK OMEPTNWORK VTSUCB.
+-@BEND
OFMPT

```

```

-@OF-OME00PC = OMEMPT
  * MESSAGE FROM PARTNER IS TO BE ISSUED TO THE TERMINAL
+-@IF
  OMEPSSH = OMESSR
  * TIAM IS WAITNG FOR INPUT (SYSTEM READ)
-@THEN
  +-@IF
    DSTFLAGL IN DSTFLAG
    * IS NEXT STEP LOGON?
  -@THEN
    * LOGON COMMAND REQUIRES A PASSWORD
    * SEND MESSAGE "PLEASE ENTER PASSWORD" TO TERMINAL
    * NOTE NEXT STEP "QUERY PASSWORD"
    MOVE OMEMES TO OME00PC.
    MOVE "PLEASE ENTER LOGON PASSWORD:" TO LOG-PAS.
    MOVE 30 TO OMEOMLEN.
    MOVE OMEEICDY TO OMEEICD.
    MOVE OMEEDIAY TO OMEEDIA.
    @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                          OMEPTNWORK VTSUCB.
    SET DSTFLAGP TO TRUE.
  -@ELSE
    +-@IF
      DSTFLAGE IN DSTFLAG
      * IS NEXT STEP AN EXEC CALL?
    -@THEN
      * EDT IS CALLED
      * EXEC COMMAND TO PARTNER IS SENT
      * NOTE NEXT STEP "EXIT EXAMPLE FINISHED"
      MOVE OMEMTP TO OME00PC,
      MOVE "EXEC $EDT" TO EXEC.
      MOVE 9 TO OMEOMLEN.
      @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                          OMEPTNWORK VTSUCB.
      SET DSTFLAGF TO TRUE.
    -@ELSE
      * PASS ON INPUT PROMPT TO TERMINAL
      @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                          OMEPTNWORK VTSUCB.
    +-@BEND
  +-@BEND
-@ELSE
  * PARTNER OUTPUT MESSAGE
+-@IF
  (DSTFLAGL IN DSTFLAG) OR (DSTFLAGE IN DSTFLAG)
  * IS NEXT STEP LOGON OR EXEC ?
-@THEN
  * EXIT ROUTINE IS STILL ACTIVE

```

```

        * SUPPRESS THE MESSAGE
    -@ELSE
        * PASS ON MESSAGE TO TERMINAL
        @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
            OMEDSTWORK OMEPTNWORK VTSUCB.
    +-@BEND
+-@BEND
*
OFCOM
-@OF-OME00PC = OMECOM
    * COMMAND TO OMNIS HAS BEEN ENTERED
    +-@IF
        DSTFLAGP IN DSTFLAG
        * IS NEXT STEP PASSWORD QUERY?
    -@THEN
        * PASSWORD IS ACCEPTED
        * SET PARTNER "PTN1"
        * SEND LOGON COMMAND TO PARTNER
        * NOTE NEXT STEP "EXEC"
        MOVE OMESLPA TO OME00PC.
        MOVE "PTN1" TO OMEPPAC.
        MOVE 4 TO OMEPPACL.
        @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
            OMEDSTWORK OMEPTNWORK VTSUCB.

        MOVE OMEMTP TO OME00PC.
        MOVE SPACES TO PASSWORD.
        SET INDEX-M TO 1.
        SET INDEX-P TO 1.
        +WHILE+
            -@WHILE
                INDEX-M NOT > OMEOMLEN
            -@DO
                MOVE MSG-PASS-E(INDEX-M) TO PASSWORD-E(INDEX-P).
                SET INDEX-M UP BY 1.
                SET INDEX-P UP BY 1.
        +-@BEND
        MOVE "LOGON OMNIS,1,C'" TO LOGON.
        MOVE PASSWORD TO LOGONP .
        MOVE "" TO LOGONE .
        MOVE 22 TO OMEOMLEN.
        @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
            OMEDSTWORK OMEPTNWORK VTSUCB.

        MOVE SPACES TO PASSWORD.
        SET DSTFLAGE TO TRUE.
    -@ELSE
        * ISSUE OMNIS COMMAND WITHOUT INSPECTION
        @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
            OMEDSTWORK OMEPTNWORK VTSUCB.

```

```
      +--@BEND
      *
      OFREST
-@OFREST
      * NOTHING TO DO - MOVE ON TO OMNIS
      @PASS OMSCALL :TYP=E: USING OMEPAR OMEMESSAGE
                                   OMEDSTWORK OMEPTNWORK VTSUCB.
      +--@BEND
      @EXIT
+--@END
```

Related publications

You will find the manuals on the internet at <http://manuals.ts.fujitsu.com>. You can order printed copies of those manuals which are displayed with an order number.

- [1] **OMNIS/OMNIS-MENU (BS2000)**
Functions and Commands
User Guide
- [2] **OMNIS-PROP (BS2000)**
User Guide
- [3] **BS2000 OSD/BC**
Commands
User Guide
- [4] **BS2000 OSD/BC**
Executive Macros
User Guide
- [5] **ASSEMBH (BS2000)**
User Guide
- [6] **Assembler Instructions (BS2000)**
Reference Manual
- [7] **BCAM (BS2000)**
User Guide
- [8] **DCAM (BS2000)**
Macros
User Guide
- [9] **HIPLEX AF (BS2000)**
High Availability of Applications in BS2000
User Guide

Related publications

- [10] **JV (BS2000)**
Job Variables
User Guide

- [11] **openUTM (BS2000)**
Generating Applications
User Guide

- [12] **openUTM**
Concepts and Functions
User Guide

- [13] **SDF-P (BS2000)**
Programming in the Command Language
User Guide

- [14] **TIAM (BS2000)**
User Guide

Index

*UNCHANGED
 in DECLARE-USER 169
*UNLOCK 69, 70

A

access
 to OMNIS 79
 to the partners 80
ACK
 in DECLARE-TERMINAL statement 157
ADD
 in DECLARE-GROUP statement 122
ADM
 in DECLARE-TERMINAL statement 158
ADMI
 in DECLARE-MENU 127
administration commands 66
administration concept 63
administration of OMNIS 61
administration password 63
administrator function 69, 70
administrator terminal 63
APPL
 in DECLARE-MENU 127
APPLICATION
 start parameter 51
application
 global 44, 194
 local 44, 194
application declaration
 OMNISKD 112
application definition
 OMNISMD 192
application list 194, 196, 198

application number 41
application switch
 behavior during 196
application table 44
APPLICATION-NUMBER
 in DECLARE-APPLICATION 115
apptab 44
APPNAME
 in DECLARE-APPLICATION 115
 in DECLARE-PARTNER statement 141
authtab 45
AUTO-LOGOFF
 in DECLARE-PARTNER statement 141
 in DECLARE-TERMINAL statement 158
AUTOLOGOFF-BUT
 in DECLARE-APPLICATION 115
AUTOMATIC-CLSCON
 in DECLARE-APPLICATION 115
AVAILABILITY
 in DECLARE-MUX 129

B

BERID
 in DECLARE-PARTNER 116
 in DECLARE-PARTNER statement 141
BREAK-KEY
 in DECLARE-TERMINAL statement 158
BS2000 console
 declaring 120
BULLETIN
 operand in FILE statement 53
bulletin file 26

C

- CALL-HOLD
 - in DECLARE-PARAMETER 134
- CALL-INF
 - in DECLARE-TERMINAL statement 158
- CALL-KEY
 - in DECLARE-TERMINAL statement 159
- CANCEL 66
- CHANGE
 - in DECLARE-PARTNER statement 142
 - in DECLARE-TERMINAL statement 159
- CHANGELOG 66
- CLASS
 - in DECLARE-PARTNER statement 142
- CMD 66
- CMSG
 - in DECLARE-APPLICATION 119
 - in DECLARE-CONSOLE statement 120
 - in DECLARE-PARTNER statement 147
- CODE
 - in DECLARE-APPLICATION 116
 - in DECLARE-PARTNER statement 142
- COLOR
 - in DECLARE-PARTNER statement 142
 - in DECLARE-TERMINAL statement 159
- Columbus Assembler
 - exit routines 289
- command files 28
- command scopes 64
- COMMENT
 - in DECLARE-APPLICATION 116
 - in DECLARE-EXIT command 121
 - in DECLARE-GROUP command 122
 - in DECLARE-HARDCOPY command 123
 - in DECLARE-MENU 126
 - in DECLARE-MUX 129
 - in DECLARE-OMNIS 131
 - in DECLARE-PARAMETER 134
 - in DECLARE-PARTNER 142
 - in DECLARE-PKEYSET 149
 - in DECLARE-START-SEQUENCE 150
 - in DECLARE-TAC 151
 - in DECLARE-TERMINAL 159
 - in DECLARE-USER 168
- communication area
 - exit routines 259
- CONFIG
 - operand in FILE statement 53
- configuration declaration 101
- configuration example 207, 210
- configuration file
 - generation, example of 209
- configuration input file
 - example 209
- CONFUPDATE 66
- CONNECT
 - in DECLARE-HARDCOPY command 124
 - in DECLARE-MUX statement 130
 - in DECLARE-OMNIS statement 132
 - in DECLARE-PARTNER statement 142
 - in DECLARE-TERMINAL statement 159
- connection clear-down
 - on inactivity 127, 137, 171, 183
- CONNECTION-ATTEMPTS
 - in DECLARE-PARAMETER 134
- console
 - in DECLARE-CONSOLE statement 120
- console as terminal
 - BS2000 generation 87
- console commands 89
- CPASS
 - in DECLARE-APPLICATION 116
 - in DECLARE-MUX statement 130
 - in DECLARE-OMNIS statement 132
 - in DECLARE-PARTNER statement 143
- creating an exit routine 305

D

data in parameter areas
 exit routines 263

data input
 OMNISMD 179

data protection 79

DCAM programming
 prerequisites for OMNIS 99

DECLARE-APPLICATION 112

DECLARE-CONSOLE 120

DECLARE-EXIT 121

DECLARE-GROUP 122

DECLARE-HARDCOPY 123

DECLARE-MENU 125

DECLARE-MUX 129

DECLARE-OMNIS 131

DECLARE-PARAMETER 133

DECLARE-PARTNER 138

DECLARE-PKEYSET 149

DECLARE-START-SEQUENCE 150

DECLARE-TAC 151

DECLARE-TERMINAL 153

DECLARE-USER 167

declaring
 a configuration 101
 a configuration (order of declarations) 106

Define Application 192

Define Menu Entry 190

Define User 181

DELETE-PKEYS
 in DECLARE-PARAMETER 134

DEVICE
 in DECLARE-MENU 127

disconnection, automatic 197

DISCONNECTION-MASK
 in DECLARE-PARAMETER 135

DISPLAY-MODE
 in DECLARE-TERMINAL statement 159

DUMP 66

E

eac
 in DECLARE-EXIT command 121

EDIT
 operand in RUN statement 240

EDIT area
 exit routines 273

END 66
 OMNISLC 238
 start parameter 52

ENTER file 27

entries in startup file
 example 211

ENTRY-LIST
 in DECLARE-MENU 126

error
 behavior in the event of 85

examples
 configuration 207
 exit routines (COBOL) 314

exclusive partner 183

EXCLUSIVE-PARTNER
 in DECLARE-TERMINAL statement 160
 in DECLARE-USER 168

EXIT
 in DECLARE-PARTNER statement 143
 in DECLARE-TERMINAL statement 160

exit group declaring 121

exit routine calls OMNIS 264

exit routine creation 305

exit routines
 Assembler interface 278
 assembling/compiling 301
 COBOL interface 290
 Columbus Assembler 289
 data in parameter areas 263
 data transfer 261
 H-Assembler 289
 interface to OMNIS 257
 introduction 249
 loading and activating 302
 testing 303

exit routines (embedding in OMNIS) 254

exit routines (parameter areas) 258

F

- FAILURE
 - in DECLARE-HARDCOPY command 124
- FILE 66
 - start parameter 53
- file
 - conversion from earlier versions 225
 - processing/presenting 225
- FKEY
 - in DECLARE-MENU 126
- Function Menu
 - OMNISMD 180
- function overview: OMNISMD 175

G

- gac
 - in DECLARE-GROUP statement 122
- generation data 200
- generation file 44
- generation in UTM
 - example 220
- gentab 44
- global applications 44, 194
- go-back key
 - for OMNISMD 202

H

- H-Assembler
 - exit routines 289
- hac
 - in DECLARE-HARDCOPY command 123
- hardcopy key 187
- HARDCOPY-DEVICE
 - in DECLARE-USER 168
- HCOPY
 - in DECLARE-MENU 127
 - in DECLARE-PARTNER statement 143
 - in DECLARE-TERMINAL statement 160
- HCY area in exit routines 277
- HELP
 - OMNISLC 238
- help function
 - OMNISMD 179

- HIPLEX AF support
 - in DECLARE-MUX 129
- HOLD
 - in DECLARE-PARTNER statement 143
 - in DECLARE-TERMINAL statement 161
- HSMS 25

I

- ID card information 183
- IMPLICIT-APPLICATION-SWITCH-DIALOG
 - in DECLARE-PARAMETER 135
- IMPLICIT-APPLICATION-SWITCH-KKEY
 - in DECLARE-PARAMETER 135
- IMPLICIT-APPLICATION-SWITCH-PKEY
 - in DECLARE-PARAMETER 135
- index key of start sequence 41
- INF 66
- INFO
 - in DECLARE-USER 168
- INPUT
 - OMNISLC 239
- INPUT-LOGGING
 - in DECLARE-TERMINAL statement 162
 - operand in SELECT statement 242
- INPUT-SAVE
 - in DECLARE-TERMINAL 162
- installation
 - of OMNIS-MENU 59
- interface library 42
- IOAREA
 - operand in LENGTH statement 54
- ISO in APPLICATION statement 51

J

- job variables 72

K

- key length OMNISFC 227
- KKEY
 - in DECLARE-MENU 126
- KPAC
 - in DECLARE-PARTNER statement 144
 - in DECLARE-TERMINAL statement 163

L

LCASE

- in DECLARE-APPLICATION 116
- in DECLARE-PARTNER statement 144
- operand in RUN statement 240

LENGTH

- in DECLARE-USER 169
- start parameter 54

LENGTH IOAREA

- minimum length 212

LENGTH TWORK

- minimum length 212

LEVEL

- operand in SECURITY statement 55

libraries 42

LINE

- in DECLARE-APPLICATION 116
- message suppression 195

LINE25

- in DECLARE-PARTNER statement 144
- in DECLARE-TERMINAL statement 163

LISTENING

- in DECLARE-TERMINAL 163

LMSG

- in DECLARE-APPLICATION 119
- in DECLARE-CONSOLE statement 120
- in DECLARE-PARTNER statement 144, 147

load distribution

- through parallel multiplex connections 97

local applications 44, 194

LOGGING

- in DECLARE-PARTNER statement 145
- in DECLARE-TERMINAL statement 163
- operand in FILE statement 53

logging file 30

- editing 235
- printing 34
- processing 34
- structure 31

login mask file 45

login mask, OMNIS-MENU

- changing the layout 60

logout key 202

logout mask

- output 200

LOGOUT-HOLD-AVAILABILITY

- in DECLARE-USER 169

LPASS

- in DECLARE-MUX statement 130
- in DECLARE-OMNIS statement 132
- in DECLARE-PARTNER statement 145

M

macro library 42

mapping area

- exit routines 276

master agent (SNMP) 15

MENU

- in DECLARE-USER 169

menu declaration

- OMNISKD 125

menu entry

- defining 190

menu entry type 191

menu name 187

menu table 44

MENUNAME

- in DECLARE-TAC 152

menutab 44

MESSAGE 67

message file 38

message flow control 183

message receive buffer

- minimum length 212

messages

- exit routines 300

metasyntax

- OMNISKD 109

MIGRATE procedure 234

MODIFY-AUTHORIZATION-MASK

- in DECLARE-PARAMETER 135

MODLIB

- operand in FILE statement 53

module library 42

MODULES

- in DECLARE-EXIT command 121

- MSG
 - in DECLARE-PKEYSET 149
- MTAB
 - in DECLARE-PARTNER statement 145
 - in DECLARE-TERMINAL statement 164
- multiple start
 - of OMNIS 58
- multiplex connection
 - declaring 129
- multiplex connections 94
 - generation in OMNIS 94
 - generation in UTM 94
 - OMNIS-UTM, example 224
- MUX area
 - exit routines 274
- N**
- NAME
 - in DECLARE-MENU 126
 - in DECLARE-USER 168
- notational conventions OMNISKD 109
- O**
- oac in DECLARE-OMNIS statement 131
- OCCUPY 67
- off-line diagnostics 65
- OMECEQU
 - exit copy element/COBOL 294
- OMECPAR
 - exit copy element/COBOL 291
- OMEEQU
 - exit EQUATES/Assembler 282
- OMEPAR
 - exit data structures/Assembler 279
- OMEPARL
 - exit parameter list/Assembler 278
- omm.appltab 44
- omm.authtab 45
- omm.gentab 44, 200
- omm.menutab 44, 187
- omm.pkeylib 45
- omm.plus.in 149
- omm.tactab 45
- omm.usertab 44, 182
- OMMFADM 69, 70
- OMMGAP1 192
- OMMGAP2 193
- OMMGGEN 200
- OMMGMAI 180
- OMMGMEE 190
- OMMGMEN 186
- OMMGPRT 203
- OMMGTXT 205
- OMMGUSR 181
- OMNIS
 - declaring 131
 - ENTER file, example 207
 - multiple start 58
 - starting 48
 - OMNIS administrator tasks 61
 - OMNIS calls exit routine 261
 - OMNIS configuration example 207
 - OMNIS copy elements 42
 - OMNIS files 23
 - creating 47
 - OMNIS files and libraries
 - overview 23
 - OMNIS macros 42
 - OMNIS message file 59
 - OMNIS-COMMAND-MODE
 - in DECLARE-USER 169
 - OMNIS-DST work area
 - minimum length 212
 - OMNIS-KKEY
 - in DECLARE-PARAMETER 135
 - OMNIS-MENU administration 68
 - OMNIS-MENU definitions 173
 - OMNIS-MENU File Converter 225
 - OMNIS-MENU files 43
 - OMNIS-PKEY
 - in DECLARE-PARAMETER 135
 - OMNISFC 225
 - startup procedure 226
 - OMNISKD 101
 - configuration example 208
 - OMNISKD input file 104
 - example 213

- OMNISKD input files
 - creating OMNISFC 225
- OMNISLC 34, 235
- OMNISMD 173
 - Define Application 192
 - Define Menu Entry 190
 - Define Parameters 200
 - Define User Menu 186
 - edit start sequence 205
 - entering users 181
 - Function Menu 180
 - function overview 175
 - help function 179
 - printing tables 203
 - screen layout 179
 - starting 173
 - terminating 206
- OMS area
 - exit routines 269
- on-line diagnostics 64
- ONE-USER-ONE-CONNECTION
 - in DECLARE-PARAMETER 136
- OPASS
 - in DECLARE-PARTNER statement 145
 - in DECLARE-TERMINAL statement 164
- operation codes 262
- OPNCON
 - in DECLARE-OMNIS statement 132
 - in DECLARE-TERMINAL statement 164
- OPTION 67
- OUTPUT 239
- output
 - when EDIT=ISAM 246
 - when EDIT=NO 244
 - when EDIT=SAM 245
- OUTPUT-LOGGING
 - in DECLARE-TERMINAL statement 164
 - operand in SELECT statement 242
- P**
- P key library 45
- P key set
 - declaring 149
 - generation, example 220
- P key sets
 - converting 228
- P keys
 - saving 196
- PAC
 - in DECLARE-APPLICATION 114, 126
 - in DECLARE-PARTNER statement 145
 - in DECLARE-TERMINAL statement 165
 - in SELECT command 242
- pac
 - in DECLARE-PARTNER statement 141
- PAGING 53
- paging file 24
- parameter
 - declaring, OMNISKD 133
 - defining for OMNISMD 200
- partner
 - declaring 138
- partner group
 - declaring 122
- PASSWORD
 - in DECLARE-USER 169
- password
 - complexity 136
 - transfer 105, 227
- password complexity 201
- PATH
 - in DECLARE-APPLICATION 114
 - in DECLARE-MENU 126
- PID
 - operand in SELECT statement 242
- PKEY
 - in DECLARE-APPLICATION 117
 - in DECLARE-MENU 126
 - in DECLARE-PARTNER statement 146
 - in DECLARE-TERMINAL statement 165
- PKEYLIB
 - in DECLARE-PKEYSET 149
- pkeylib 45

PLUS input file 149
 converting 228
POSITION
 in DECLARE-USER 170
predefined variables 71
PREFIX
 operand in APPLICATION statement 51
PRINCIPAL
 in DECLARE-USER 170
printer declaring 123
printer name for hardcopy 182, 187
PRIORITY 67
privileged functions 63
procedure
 MIGRATE 234
programming with OMNIS 249
PRONAME
 in DECLARE-APPLICATION 114
 in DECLARE-HARDCOPY command 123
 in DECLARE-MUX statement 129
 in DECLARE-OMNIS statement 131
 in DECLARE-PARTNER statement 145
 in DECLARE-TAC 151
 in DECLARE-TERMINAL statement 157
PROTECT-PW
 in DECLARE-PARAMETER 136
PROTOCOL
 in DECLARE-APPLICATION 117
 in DECLARE-PARTNER statement 146
PTAST
 in DECLARE-APPLICATION 117
 in DECLARE-PKEYSET 149
PTNNNAME
 in DECLARE-APPLICATION 117
 in DECLARE-HARDCOPY command 123
 in DECLARE-MUX statement 129
 in DECLARE-OMNIS statement 132
 in DECLARE-PARTNER statement 146
 in DECLARE-TAC 151
 in DECLARE-TERMINAL statement 157
PTNR area
 exit routines 272
PTNR work area
 exit routines 259

R
Readme file 10
RECORDS
 in DECLARE-START-SEQUENCE 150
recursive calls in exit routines
 COBOL interface 299
REPLY
 in DECLARE-TERMINAL 165
reply suppression 165
reset key
 in DECLARE-PARAMETER 136
RESET-KKEY
 in DECLARE-PARAMETER 136
RESET-PKEY
 in DECLARE-PARAMETER 136
restart message 197
RESTART-MESSAGE
 in DECLARE-APPLICATION 117
return codes (exit routines) 266
ROUTE
 in DECLARE-APPLICATION 117
 in DECLARE-PARTNER statement 146
RUN 240
S
S variables 71
SAVE
 in DECLARE-APPLICATION 117
 in DECLARE-PARTNER statement 146
 in DECLARE-TERMINAL statement 165
SCREEN
 in DECLARE-APPLICATION 118
 in DECLARE-PARTNER statement 146
 in DECLARE-TERMINAL statement 165
screen contents
 saving 196
screen layout
 OMNISMD 179
SECURITY 84
 start parameter 55
SELECT 241
SET 67
SETSW
 start parameter 57

- SIGNON
 - in DECLARE-APPLICATION 118
- SNMP 13
- SNMP management 13
 - architecture 13
- special character for application switching
 - in DECLARE-PARAMETER 136
- start sequence 41
 - declaring 150
- START-OMNISFC 226
- START-SEQUENCE
 - in DECLARE-PARTNER statement 147
- START-SEQUENCE-NUMBER
 - in DECLARE-START-SEQUENCE 150
- start-up file 39
- starting OMNIS-MENU
 - example 212
- starting OMNISKD
 - example 219
- starting OMNISMD 173
- Startsequenz bearbeiten, OMNISMD 205
- STARTUP
 - operand in FILE statement 53
- statements
 - OMNISKD 109
 - OMNISLC 238
- subagent (SNMP) 15, 16
- suppression of LINE message 195
- SWITCH
 - in DECLARE-TAC 152
- switch character
 - OMNISMD 201
- SWITCH-CHARACTER
 - in DECLARE-PARAMETER 136
- syntax
 - of OMNISLC statements 238
 - OMNISKD 109
- SYSLIB 42
- T**
- table printouts 203
- TAC allocation table 45, 151
- TAC table
 - declaring entry 151
- TACNAME
 - in DECLARE-TAC 151
- tactab 45
- TERM area
 - exit routines 270
- TERM work area
 - exit routines 259
- TERMINAL
 - in SELECT statement 242
- terminal
 - declaring 153
- terminal as console
 - BS2000 generation 89
- TERMTYP
 - in DECLARE-TERMINAL statement 166
- test mode 64
- TEXT
 - in DECLARE-MENU 126
 - in FILE statement 53
- text file 40
 - conversion 227
- TEXTKEY
 - operand in LENGTH statement 54
- TEXTKEY-LENGTH
 - in DECLARE-PARAMETER 137
- TID
 - operand in SELECT statement 243
- TIME
 - in DECLARE-OMNIS statement 132
- time stamp for password 202
- TIMEOUT
 - in DECLARE-PARAMETER 127, 137
 - in DECLARE-USER 171
- TRACE 68
- trace file 24
- TWORK
 - operand in LENGTH statement 54
- TYP
 - in DECLARE-PARTNER statement 147

TYPE

- in DECLARE-APPLICATION 118
- in DECLARE-MENU 127

U

Unlock terminal 69

Update Configuration 70

USE

- in DECLARE-HARDCOPY command 124

USER

- in SELECT statement 243

user

entering 181

unique 136, 200

user list 182

user menu

defining 186

user menu list 187

user menu name 182

user name 182

user number 41

user table 44

USER-ID

creating 47

USER-NUMBER

in DECLARE-APPLICATION 118, 119

in DECLARE-MENU 127

in DECLARE-USER 171

USER-PROTOCOL

in DECLARE-TERMINAL statement 166

usertab 44

UTM administration 93

UTM generation 91

without a multiplex connection 91

V

variable concept 71

version change 229

figure for OMNIS 230

figure for OMNIS-MENU 231

MIGRATE procedure 234

OMNIS 229

OMNIS-MENU 231

VTSUCB area 260