

English



FUJITSU Software BS2000

# OMNIS V8.5 / OMNIS-MENU V3.5

Functions and Commands

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](mailto: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](http://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

<b>1</b>	<b>Preface . . . . .</b>	<b>9</b>
<b>1.1</b>	<b>Objectives and target groups of this manual . . . . .</b>	<b>9</b>
<b>1.2</b>	<b>Summary of contents . . . . .</b>	<b>10</b>
<b>1.3</b>	<b>Changes since the last edition of the manual . . . . .</b>	<b>11</b>
<b>1.4</b>	<b>Notational conventions . . . . .</b>	<b>11</b>
<b>2</b>	<b>Introduction to OMNIS . . . . .</b>	<b>13</b>
<b>2.1</b>	<b>The functional principle of OMNIS . . . . .</b>	<b>13</b>
<b>2.2</b>	<b>Possible implementations . . . . .</b>	<b>25</b>
2.2.1	Dialog control . . . . .	25
2.2.2	Administration of applications . . . . .	26
2.2.3	Implementation in product development . . . . .	34
<b>2.3</b>	<b>Fundamentals of operation with OMNIS . . . . .</b>	<b>35</b>
2.3.1	Configurations and connection options . . . . .	35
2.3.2	Files for OMNIS . . . . .	42
2.3.3	OMNIS diagnostics . . . . .	44

<b>3</b>	<b>Connection setup and cleardown</b>	<b>47</b>
<b>3.1</b>	<b>Connections to terminals</b>	<b>47</b>
3.1.1	BS2000 consoles used as terminals	50
3.1.2	DCAM/CMX applications used as terminals	51
3.1.3	ISO applications used as terminals	52
<b>3.2</b>	<b>Connections to partners</b>	<b>53</b>
3.2.1	TIAM partners	59
3.2.2	UCON partners	61
3.2.3	DCAM partners	66
3.2.4	UTM partners	68
<b>3.3</b>	<b>Start sequences</b>	<b>73</b>
<b>3.4</b>	<b>Loss of connection to terminals and partners</b>	<b>74</b>
<b>3.5</b>	<b>Taking over connections from terminals and partners</b>	<b>77</b>
<b>3.6</b>	<b>OMNIS-OMNIS connections</b>	<b>80</b>
<b>3.7</b>	<b>Connections to printers</b>	<b>82</b>
<b>3.8</b>	<b>OMNIS and HIPLEX AF</b>	<b>84</b>
<b>3.9</b>	<b>Load distribution</b>	<b>86</b>
3.9.1	Load distribution with HIPLEX MSCF	86
3.9.2	Load distribution on the basis of a variable concept	87

---

<b>4</b>	<b>Dialog with OMNIS</b>	<b>89</b>
<b>4.1</b>	<b>Address codes and identifiers</b>	<b>89</b>
<b>4.2</b>	<b>Connection of terminals to OMNIS</b>	<b>93</b>
<b>4.3</b>	<b>Operation in line mode</b>	<b>94</b>
<b>4.4</b>	<b>Operation in formatted mode</b>	<b>96</b>
<b>4.5</b>	<b>Switching between multiple partners in formatted mode</b>	<b>98</b>
<b>4.6</b>	<b>Switching from formats to OMNIS command mode</b>	<b>101</b>
<b>4.7</b>	<b>Special points relating to the use of K keys in OMNIS</b>	<b>104</b>
<b>4.8</b>	<b>Output control</b>	<b>105</b>
4.8.1	Message flow control	106
4.8.2	Screen overflow control	108
4.8.3	Halting the output	109
4.8.4	Interrupting the dialog	110
<b>4.9</b>	<b>Automatic system operating through message tables</b>	<b>111</b>
<b>4.10</b>	<b>Screen and P key saving</b>	<b>117</b>
<b>4.11</b>	<b>Color control</b>	<b>118</b>
<b>4.12</b>	<b>Displaying a dialog on another terminal</b>	<b>120</b>
<b>4.13</b>	<b>Logging</b>	<b>122</b>
<b>5</b>	<b>Functionality of OMNIS-MENU</b>	<b>125</b>
<b>6</b>	<b>The OMNIS-MENU user interface</b>	<b>131</b>
<b>6.1</b>	<b>Logging in to OMNIS-MENU</b>	<b>132</b>
<b>6.2</b>	<b>OMNIS-MENU Application Menu</b>	<b>135</b>
<b>6.3</b>	<b>Application switching</b>	<b>139</b>
6.3.1	Explicit application switching	139
6.3.2	Implicit application switching	141
<b>6.4</b>	<b>Logging out of OMNIS-MENU</b>	<b>145</b>
<b>6.5</b>	<b>Viewing error information</b>	<b>147</b>
<b>6.6</b>	<b>Diagnostic information in the event of an error</b>	<b>148</b>

---

<b>7</b>	<b>Special considerations of OMNIS-MENU . . . . .</b>	<b>149</b>
<b>7.1</b>	<b>User names and passwords in OMNIS-MENU . . . . .</b>	<b>149</b>
<b>7.2</b>	<b>Management of K, F and P keys . . . . .</b>	<b>154</b>
<b>7.3</b>	<b>Application switching behavior . . . . .</b>	<b>156</b>
<b>7.4</b>	<b>Handling ID card information . . . . .</b>	<b>158</b>
<b>7.5</b>	<b>Administrator messages . . . . .</b>	<b>158</b>
<b>7.6</b>	<b>Use of XHCS . . . . .</b>	<b>158</b>
<b>8</b>	<b>Kerberos support via OMNIS-MENU . . . . .</b>	<b>159</b>
<b>8.1</b>	<b>Precondition . . . . .</b>	<b>161</b>
<b>8.2</b>	<b>Access verification done by OMNIS-MENU . . . . .</b>	<b>161</b>
<b>8.3</b>	<b>Checking of the Kerberos principal . . . . .</b>	<b>162</b>
<b>8.4</b>	<b>Process of an authentication . . . . .</b>	<b>162</b>
<b>8.5</b>	<b>Declaration of the principal in OMNIS-MENU . . . . .</b>	<b>163</b>
<b>8.6</b>	<b>Connection clear down, connection loss . . . . .</b>	<b>164</b>
<b>8.7</b>	<b>Connection take-over . . . . .</b>	<b>165</b>

---

<b>9</b>	<b>OMNIS commands</b>	<b>167</b>
<b>9.1</b>	<b>Overview of the OMNIS commands</b>	<b>167</b>
<b>9.2</b>	<b>Command scope</b>	<b>168</b>
<b>9.3</b>	<b>Syntax description</b>	<b>171</b>
<b>9.4</b>	<b>Descriptions of the individual commands</b>	<b>175</b>
	CANCEL - Terminate connection	175
	CHANGELOG - Manage logging file	180
	CLSCON - Close connection to partner	189
	CMD - Execute BS2000 command	191
	CONFUPDATE - Update the configuration	192
	DUMP - Output dump	195
	END - Terminate OMNIS	197
	EXIT - Administer exit routines	198
	FILE - Process command file	201
	GROUP - Define partner group	204
	HALT - Sign off terminal from OMNIS	206
	HCOPY - Enter hardcopy unit in printer table	208
	HELP - Assistance with commands and messages	210
	INF - Output information	214
	MDEF - Define message table	229
	MESSAGE - Send messages	234
	OCCUPY - Take over existing connection	237
	OPNCON - Open connection to a partner	240
	OPTION - Define terminal-specific attributes	252
	PARAM - Define partner-specific attributes	266
	PRIORITY - Define scope of commands	274
	RESTART - Restart OMNIS-OMNIS connection/printer connection	277
	SET - Set operating parameters	278
	TRACE - Switch trace on/off	293
	TURN - Page in the dialog logging file	295

# Contents

---

<b>10</b>	<b>OMNIS Messages</b> . . . . .	<b>297</b>
<b>10.1</b>	<b>Message layout</b> . . . . .	<b>297</b>
<b>10.2</b>	<b>Short messages</b> . . . . .	<b>298</b>
	<b>Glossary</b> . . . . .	<b>299</b>
	<b>Abbreviations</b> . . . . .	<b>303</b>
	<b>Related publications</b> . . . . .	<b>305</b>
	<b>Index</b> . . . . .	<b>307</b>



---

# 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 OMNIS users and the OMNIS administrator.

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 Introduction to OMNIS

This chapter provides information about the functional principle, possible implementations and fundamentals of operation with OMNIS.

### 2.1 The functional principle of OMNIS

OMNIS is primarily a “session manager” which lets you set up connections from one terminal to a number of partners in a network concurrently.

In a network, a terminal can maintain only one connection to one communication partner in a host computer at any one time.

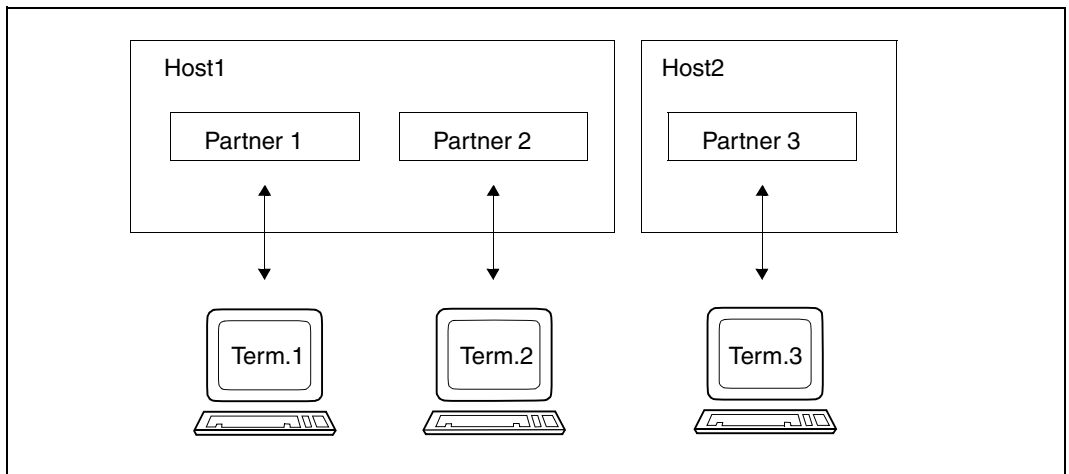


Figure 1: Connections between terminals and partners without OMNIS

OMNIS is a program that allows this restriction to be overcome. As far as the terminal user is concerned, he or she is connected to a number of partners at the same time.

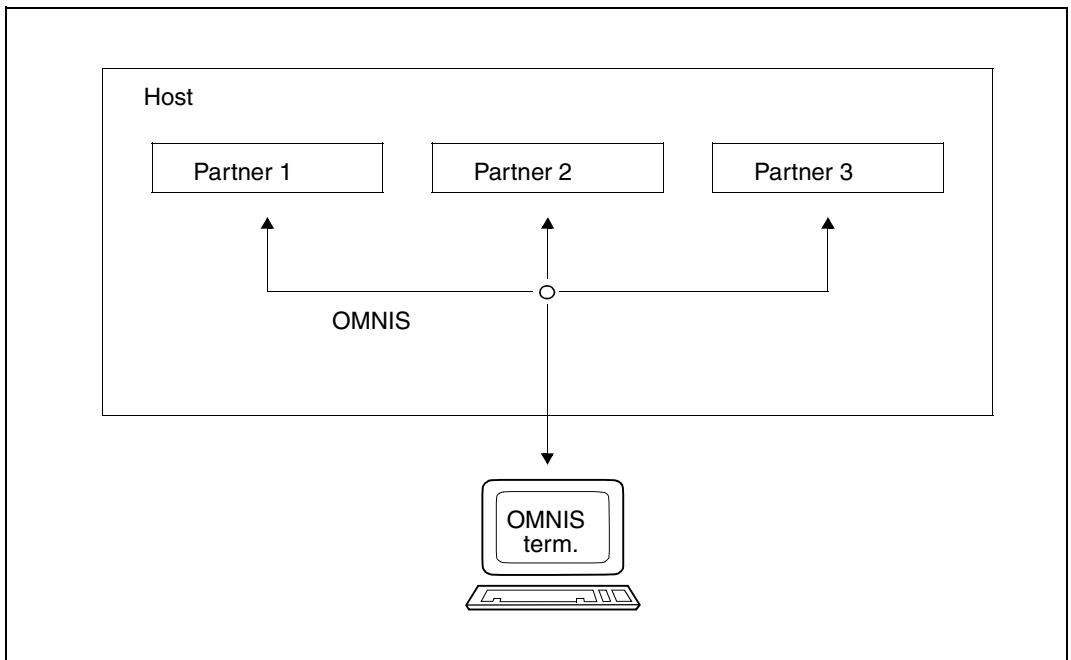


Figure 2: Connections between one terminal and three partners via OMNIS

In fact, the terminal is really connected only to one partner (OMNIS). OMNIS passes messages between the terminal and the communication partners to which the terminal has an apparent connection.

OMNIS also makes it possible to have more than one connection to a partner from the same terminal (in timesharing mode, the terminal then has a number of interactive processes at the same time).

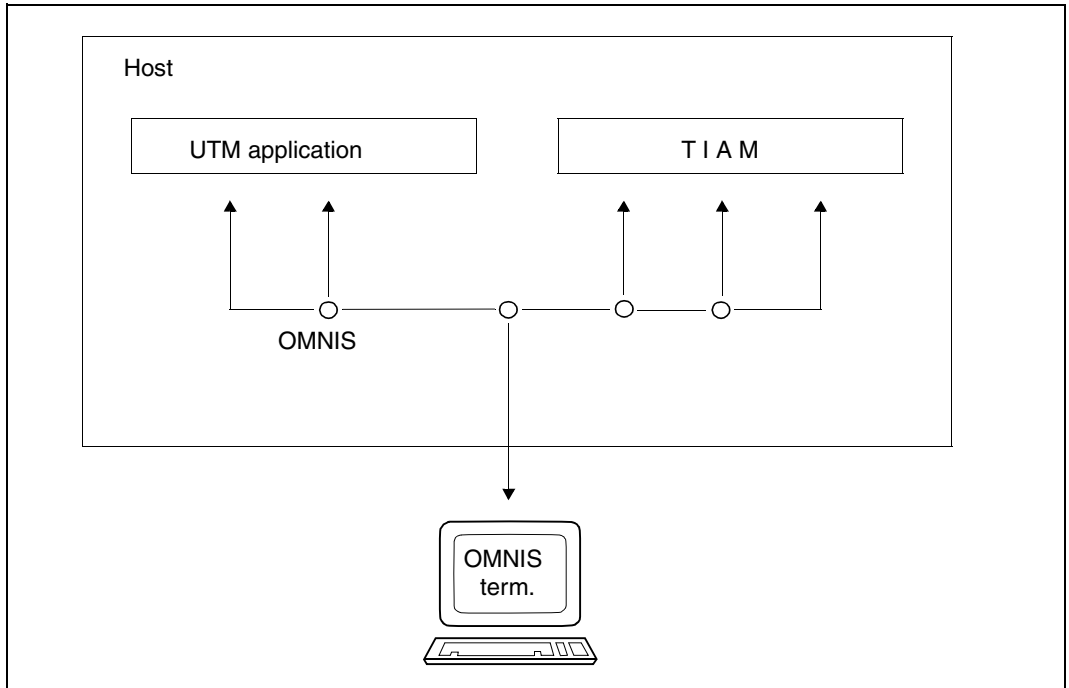


Figure 3: Multiple connection of one terminal to two partners via OMNIS

For the user, each connection is a partner which is identified by means of its own partner address code.

OMNIS (acting under orders from the terminal) can operate with many different partners at the same time. The partners for OMNIS can be:

- the \$DIALOG system application (timesharing mode, TIAM)
- the \$CONSOLE system application (operating, UCON)
- the \$VMCONS system application (virtual console, UCON)
- UTM applications
- DCAM and CMX applications

OMNIS supports all terminals supported by DCAM, and upward-compatible devices, to the same extent as DCAM. Terminals of the type 3270 are not supported.

OMNIS can be used by many terminals simultaneously and therefore usually has to be loaded only once (e.g. at system startup):

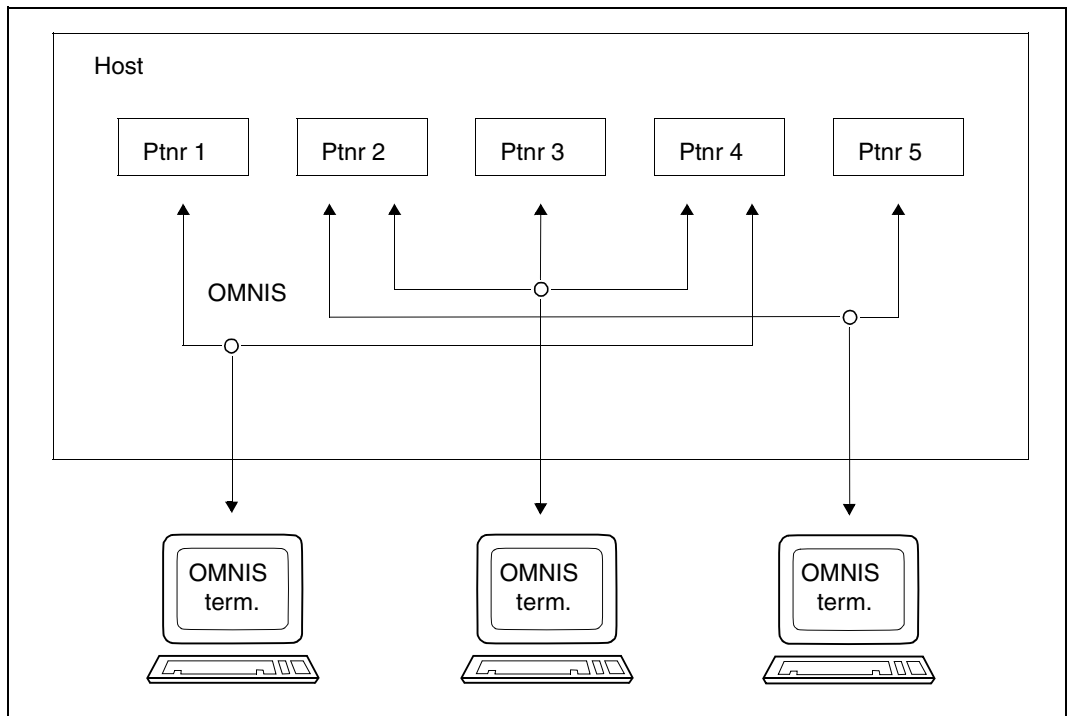


Figure 4: Connections between three terminals and five partners via OMNIS

The terminals connected concurrently to OMNIS are fully independent of one another: each terminal “thinks” it is the only OMNIS user. Full consideration is given to data protection requirements; no terminal can access another terminal's data.

OMNIS can be protected against unauthorized sign-on of terminals by means of a password.

A further protection facility is that the administrator can specify that connections are permitted only via specific terminals.

The various partners of a terminal do not all have to be in the same computer; each partner in a homogeneous network can be accessed via OMNIS, so long as the partner is known to the OMNIS computer:



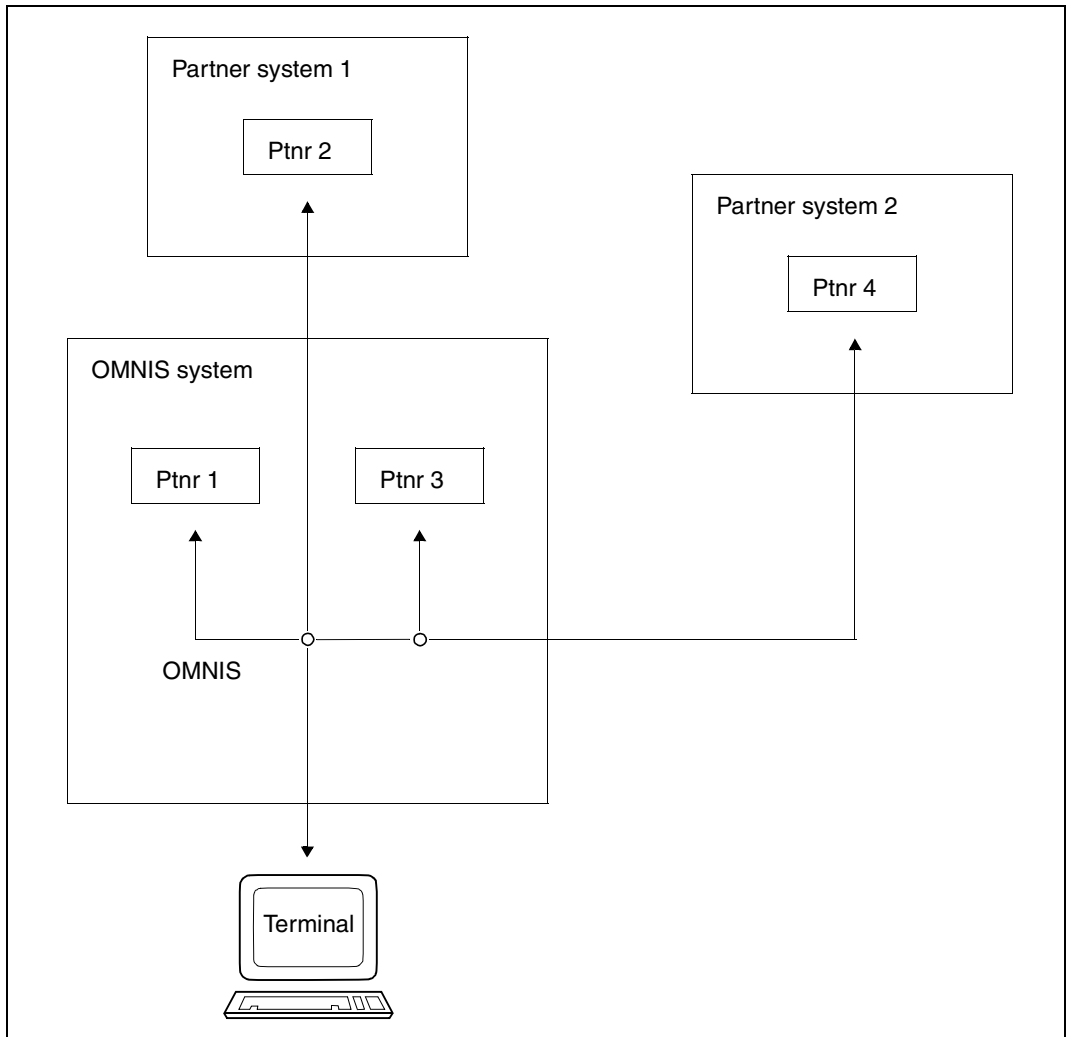


Figure 5: Communication between one terminal and a number of partners in different systems of a network via OMNIS

Because OMNIS acts as a message router between terminals and communication partners, it has to buffer, and sometimes also process, the messages before passing them on.

This is used by OMNIS to implement the functions listed below.

### **Establishing and clearing down connections to different partners**

Connections can be established from one terminal to partners of different types. Messages can be exchanged both between the terminal and the individual partners and between any of the partners themselves.

### **Switching between partners by means of K keys**

If connections to more than one partner have been set up from one terminal, the K keys can be used (even in formatted mode) to switch between the individual partners. In the case of UTM applications, the last message sent to the terminal is repeated. With any other type of partner, terminal deactivation and subsequent reactivation is simulated. How the partners respond to this depends on how the particular application concerned has been programmed. If the OMNIS extension OMNIS-MENU is used, the restart message can be defined specifically for each partner.

### **Partner groups**

A number of partners can be combined together into a group and addressed jointly. A message addressed to a partner group is automatically sent to every partner in the group; a command relating to a partner group is executed once for each partner.

### **Changing terminals during operation**

The assignment of a partner to a terminal can be altered. A user can begin a long dialog at one terminal and then, if necessary, change to another terminal from where he or she takes over the partner from the first terminal and continues the dialog with this partner.

After this the first terminal is free and can be used for other purposes.

### **Exchanging messages between terminals**

All terminals signed on to OMNIS can exchange messages with one another. A message can be sent either to another specific terminal, to all terminals (broadcast), or to all OMNIS administrators. Each recipient is informed as to the sender of the message so that a response is possible.

## OMNIS command mode

An operator terminal operates either in OMNIS command mode or in OMNIS-MENU mode. In OMNIS command mode messages from OMNIS-MENU are ignored. Which mode is currently active is shown only when switching from partner input (pac:) to OMNIS input (@:):

- OMNIS command mode announces itself with the input prompt @: /.
- OMNIS-MENU mode displays the “Application Menu” mask.

In OMNIS-MENU, whether switchover to OMNIS command mode is permitted can be defined for each user, see [page 138](#).

When the user switches from OMNIS command mode to OMNIS-MENU mode, in the case of a configured user (OMNISKD command *DECLARE-USER*) OMNIS-MENU resets the setting for message flow control to the value set for *DECLARE-USER*. See the [section “Message flow control” on page 106](#), *EXCLUSIVE-PARTNER* operand. Any setting made with the OMNIS command OPTION is overwritten.

## Information service

Users have access to a wide-ranging information service which can be used to query user-written or global resources. Only the OMNIS administrator can query the resources of other users

Users can have displayed on the screen a brief syntax description of every command, as well as detailed information on each of the messages.

## Information in the 25th line of the screen

If a partner outputs a message on the terminal, the 25th line (status line) of the screen can be used to display the most important characteristics of the partner, e.g. partner address code, partner type (UCON, TIAM, etc.), symbolic network address.

## Color control

With the 9752 Data Display Terminal, OMNIS supports four colors (red, white, yellow, green); with the 9763 Data Display Terminal, OMNIS supports seven colors (red, white, yellow, green, blue, cyan and magenta). It does this in both cases by inserting the corresponding color control characters in the original message.

You can define the colors according to the partner, i.e. each partner has his or her own color. UCON partners also have the option of deriving the color from the actual message contents (requests red, information messages yellow).

### Screen overflow control

The interface between OMNIS and the terminal incorporates an easy-to-use, switchable screen overflow control facility.

This stops the output of messages at the terminal when a screen becomes full or the next message would clear the screen.

In such cases OMNIS displays a message in the last line of the screen and waits for input (command to OMNIS, message for a partner, acknowledgment) before continuing with the message output.

When the screen overflow control facility is switched on inputs can be made at the terminal even where outputs follow one another in rapid succession and important messages do not disappear from the screen before they can be read.

The overflow control provided in timesharing mode (*MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=USER-ACKNOWLEDGE*) can be used under OMNIS only if *PROTOCOL=DSSIM* is specified.

### Screen and P key saving

Each user can have OMNIS save the current screen and / or the P key assignment on switching partners.

### Hardcopy log

Each user can have OMNIS send all or part of the output to a RSO printer. For RSO printers you can set a terminal-specific and user-/menu-specific printer by means of the *HCOPY* operand with the *FORM* parameter.

You use a K key to initiate a screen printout.

### **Exit routines (subroutine interface)**

The user has the option of transferring all messages to user-own exit routines before they are passed on to the addressee (partner, terminal or OMNIS).

These exit routines can

- modify the contents of a message
- block the forwarding of a message
- generate further messages from a message; these can in turn be sent to different recipients (e.g. a reply to the sender)
- intervene in the message stream between OMNIS and the logging or hardcopy function
- set up the connection to a terminal

### **Command files, startup file**

Frequently used command sequences can be stored in BS2000 files (SAM or ISAM) or type-S library members, and called up using an OMNIS command. The commands thus read from a command file are executed as if they had been entered at the terminal.

Both messages to partners and OMNIS commands from the command file can be read in.

The startup file is a special command file processed automatically by OMNIS immediately after loading. It is primarily intended for global settings and may contain only OMNIS commands.

### **Start sequences**

*LOGON/KDCSIGN* commands can be stored in a text file. Following a successful connection setup, OMNIS sends the commands from this file to the relevant partners.

### **Bulletin file**

The contents of the bulletin file are sent by OMNIS automatically to every terminal which signs on to OMNIS. The function of the file is the same as in timesharing mode.

### **Dialog logging file**

If required, OMNIS writes the entire dialog conducted at a terminal to a logging file.

The user can select at what time the messages are to be written to the logging file. This can take place either when messages are input/output at the terminal or when they are sent to/received from a partner or partners.

During operation the administrator can assign a new logging file to OMNIS so that the old one becomes available for off-line analysis or for backup.

### **Paging in the logging file**

Each user can page through the logging file of his or her terminal. Access to other users' messages is not permitted for reasons of security.

The user can page through the dialog with one partner, or the complete dialog for the terminal.

### **Printing the logging file**

After OMNIS has been terminated, or after a new logging file has been assigned, the (old) logging file can be used as documentation of the dialog or evaluated by programs in various ways.

With the aid of the OMNISLC utility routine, the entire logging file, the dialog with selected terminals, the dialog with selected partners, or when using OMNIS-MENU, also the dialog with selected users can be edited for printing. The messages (and also formats), including all passwords, are printed just as they normally appear on the screen.

### **Terminal as a BS2000 console (system operating)**

For reasons of data protection, terminals cannot be connected directly to the system application \$CONSOLE; a program is always needed as an intermediary. This intermediary can be OMNIS.

Using OMNIS and \$CONSOLE, a terminal can receive console messages. It is also possible to enter console commands at the terminal if authorization to do so exists.

### **Automatic answering of console messages, message suppression**

If required, OMNIS will check a console message before passing it on to the terminal in order to see whether it contains a specific message number and, if necessary, an additional character string whose position and contents are freely selectable (e.g. a variable field within the message).

If a match is established, the output of the message at the terminal is suppressed (message suppression) and/or a reply, also freely selectable, is returned to the sender of the message and/or output of a message is accompanied by an audible alarm at the terminal.

Any number of different messages with individual responses can be defined.

### **Reply suppression**

Using OMNIS you can specify for certain terminals that replies to UCON queries can only be sent from a terminal on which those queries were output.

Reply suppression for a terminal is enabled with *REPLY = RESTRICTED*.

### **BS2000 console as a terminal**

OMNIS includes a special command that enables a BS2000 console to interoperate with BS2000 applications as a terminal. This means, for example, that BS2000 user commands can be entered and programs called via the console.

### **Administrator interface**

Various functions are privileged and reserved for use by the OMNIS administrator. The administration can be either centralized (only one administrator) or, in certain cases, decentralized (more than one administrator, exceptionally all OMNIS users).

### **Data protection**

The level of data protection for OMNIS can be defined with the SECURITY start parameter when OMNIS is started.

### **Users**

When OMNIS-MENU is used, certain OMNIS commands can refer to users identified by name. If OMNIS-MENU is not used, there are no users identified by name.

## XHCS support

OMNIS supports XHCS (Extended Host Code Support). XHCS is **not** supported for consoles used as terminals.

Prerequisite for the use of XHCS is that the character sets must be defined on the computer on which OMNIS is running.

If the XHCS functionality is also to be supported for EXIT routines, these must be converted to VTSUCB.

The following restrictions must be noted when different character sets are used with XHCS:

- The output from the *TURN* command is always executed with the 7-bit standard character set.
- Input is always executed with the character set used for the last output.
- The CALL-KEY information (*CALL-INF=YES*) is always output in the standard character set.
- SDF in command mode cannot run in 8-bit mode via OMNIS.
- If an 8-bit partner is occupied with a 7-bit terminal, no further operations are possible with this partner.



## 2.2 Possible implementations

The following examples of possible implementations of OMNIS are intended to give you a first impression of the many ways in which OMNIS can be used.

### 2.2.1 Dialog control

Two areas of implementation are described here:

- dialog control for more than one application
- dialog control for heterogeneous applications

#### Dialog control for more than one application

##### *Problem 1*

A number of (UTM or DCAM) applications are executing simultaneously in a computer. A user would like to communicate with each of the individual applications in turn.

##### *Problem 2*

(UTM or DCAM) applications are executing on different computers in a computer network. Again the user needs to work with each of the individual applications in turn.

##### *Standard solution (for both problems)*

The user opens an emulation for each of the applications, signs on to the application and conducts the dialogs in parallel.

##### *OMNIS solution (for both problems)*

The user sets up connections to the individual applications via OMNIS and signs on to these applications. After this, he or she communicates with each application in turn and as required.

Switching between UTM applications can be done using the K keys. It is normally possible for the user to perform switching himself in an exit routine. (This does not only apply to UTM applications.)

An exit routine which lets you communicate with a number of applications via a menu-driven user interface is provided as standard with the product **OMNIS-MENU**. In particular, the setting up of connections to partners and switching between applications are made much easier thanks to explicit selection via K and P keys.

## Dialog control for heterogeneous applications

### *Problem*

The best way of implementing a function is to divide it up into several subfunctions (e.g. one UTM application and one interactive process). The user, however, is to remain unaware of this division (which is undertaken for technical reasons).

### *Standard solution*

The user writes his or her own dialog control routine (DCAM application) to analyze messages coming from the terminal and, depending on their contents, deliver them either to the UTM application or the interactive process.

### *OMNIS solution*

The user uses OMNIS for dialog control. The control logic must be accommodated in a user routine, but this involves much less programming effort than a DCAM application.

## 2.2.2 Administration of applications

Three such areas of implementation are described here:

- simultaneous administration of a number of applications
- administration of large computer centers
- administration of decentralized computer networks

### **Simultaneous administration of a number of applications**

#### *Problem*

In a host computer, more than one application is normally executing at the same time, e.g. two UDS database systems and three UTM applications.

Each of these applications may require administrative intervention.

#### *Standard solution*

There are two options: either a separate administration terminal is used for each application, or a single terminal is used, which is connected as and when required to the application to be administered.

*OMNIS solution*

All applications are administered from a single OMNIS terminal from which a connection has been established to each application.

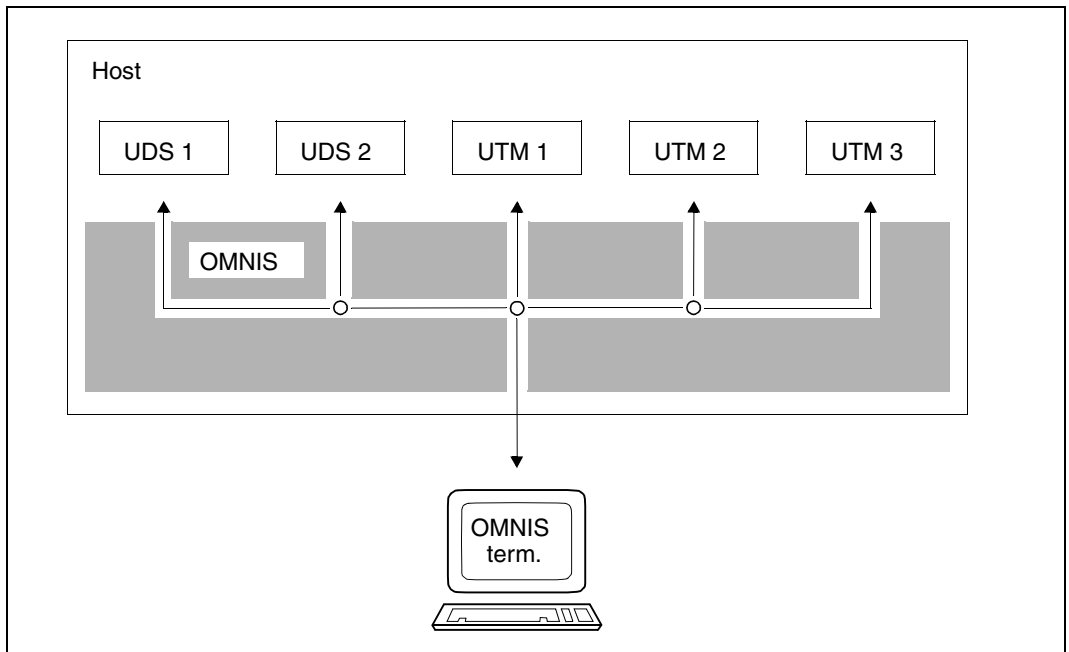


Figure 6: Administration of a number of applications from one terminal

### Administration of large computer centers

*Problem*

In a large computer center a number of computers are linked together. All the computers are to be under the control of a single operator or system administrator.

*Standard solution*

The system administrator manages each system from a separate console; it is necessary to have as many consoles as there are systems.

If all the systems are in one room, the operator has to move from one console to another and may miss important messages which have already been cleared from the screen.

*OMNIS solution*

The operator or system administrator uses OMNIS to connect to all systems to be administered. The operator functions can be handled via \$CONSOLE and the system administrator functions via \$DIALOG under privileged user IDs.

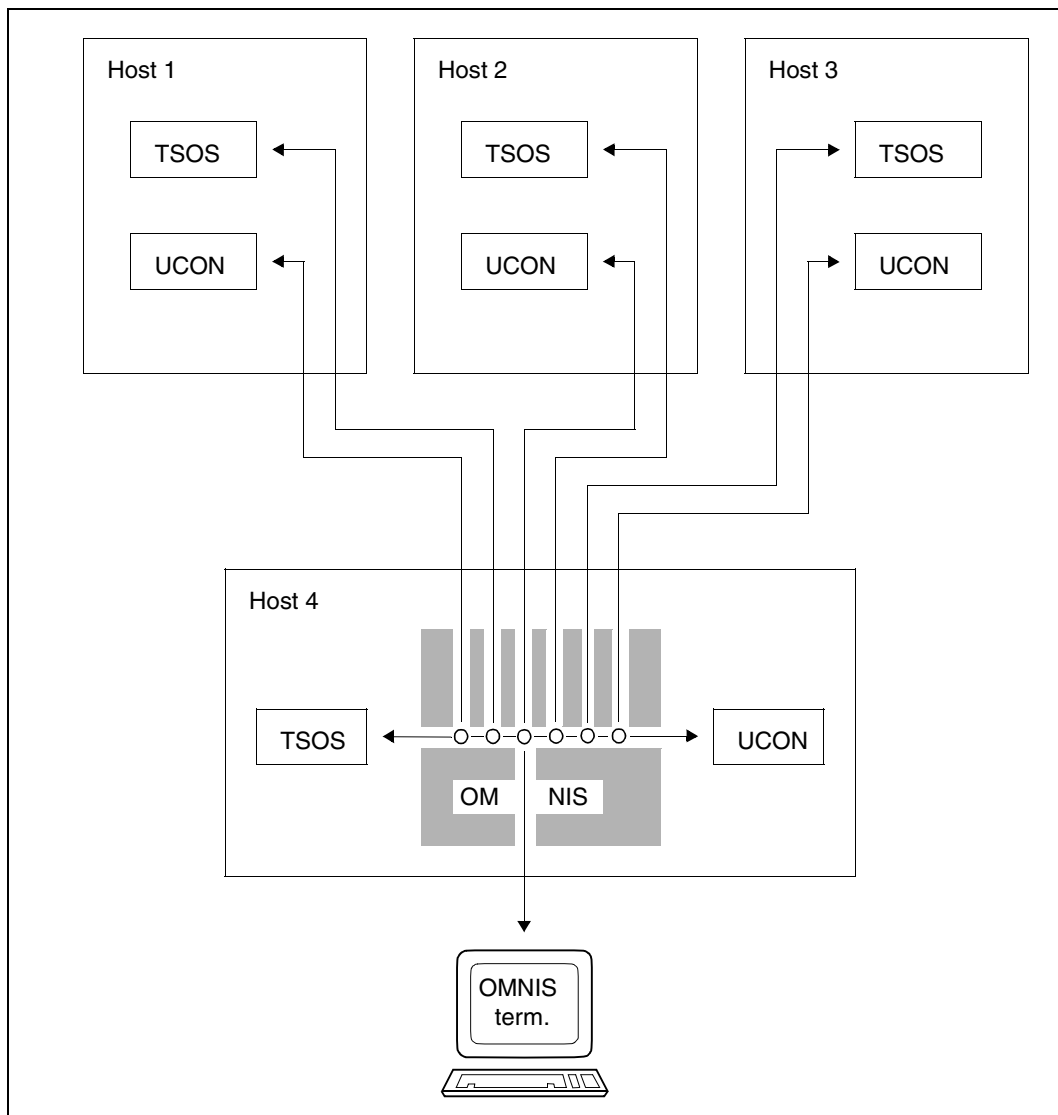


Figure 7: OMNIS in a large computer center

The work of the central operator is supported or to a large extent automated by the “automatic message answering”, “reply suppression” and “message suppression” functions.

### *Problem*

Normally one operator (or team of operators) looks after each system and are completely independent of each other.

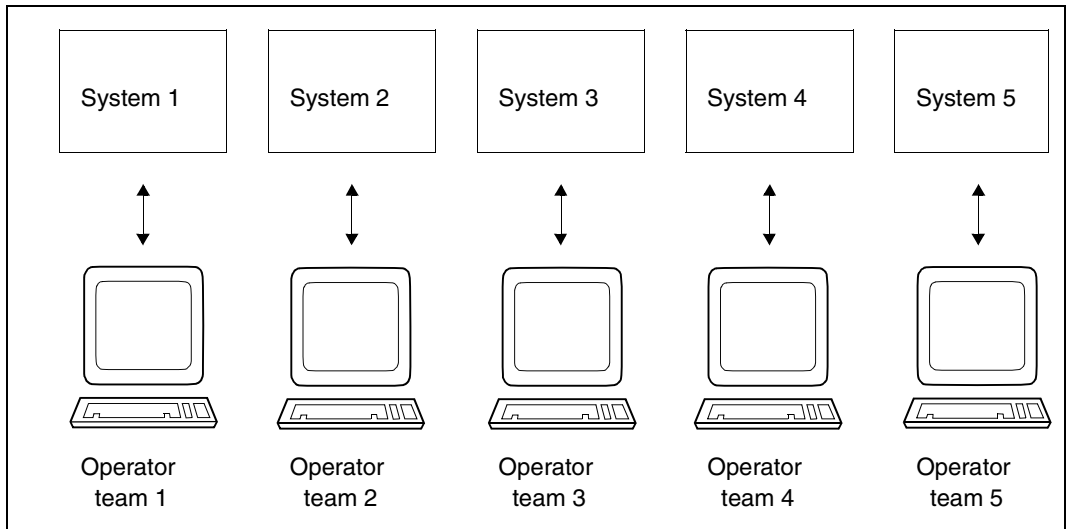


Figure 8: Operating by independent operator teams

This division is often unsatisfactory for the following reasons:

- one operator has to master the whole range of operator functions
- in many problems the operator must rely on support from other people (system administrator, archive administrator, shift leader)
- no operator has a global perspective allowing him or her to determine which system most urgently requires a device (e.g. laser printer, private disks) when there is only one in the computer center (multichannel switch)

*OMNIS solution*

Operating is divided up into several function groups using BS2000 routing codes (e.g. spoolin/spoolout, device management, system operating, job management, network administration). The operators work via OMNIS terminals, each of which is assigned to a particular function group. Each console is used to perform the operating of a particular function group for all systems.

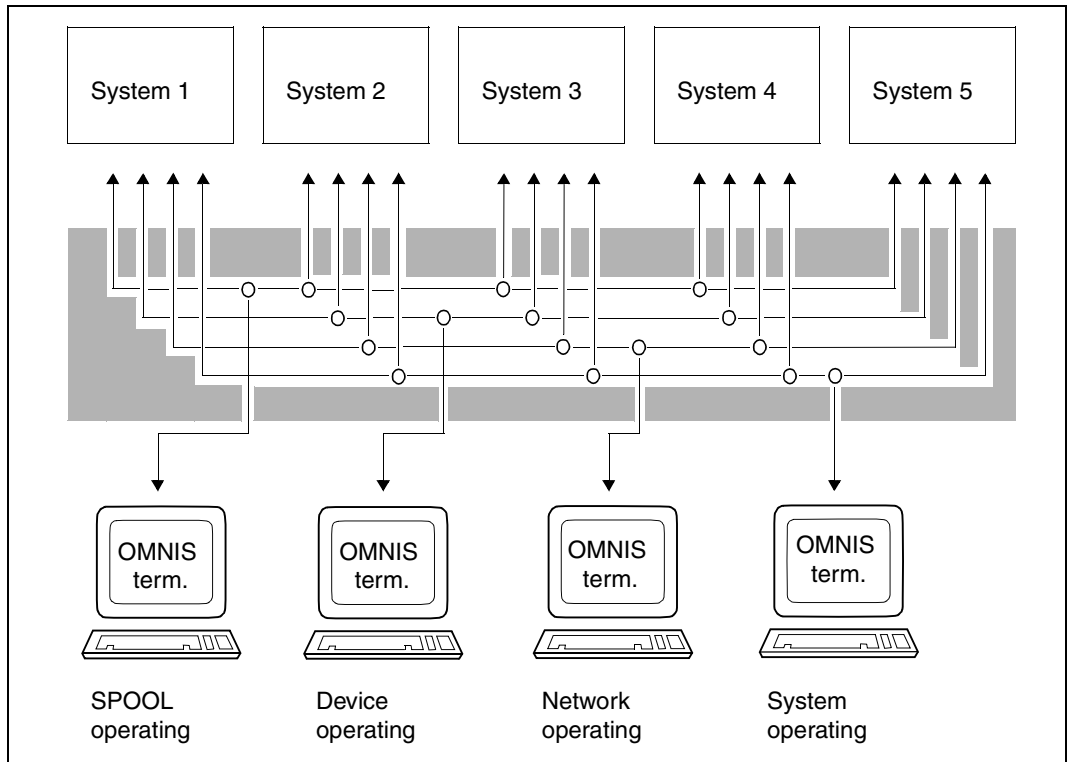


Figure 9: Operating by function group

For easy operating, OMNIS products can be used for the individual function groups as follows:

**OMNIS-PROP** offers a programming aid for routines designed to control and administer programmed operator actions, as well as ready-made, simple standard solutions for certain areas of operating (see the “OMNIS-PROP” manual [2]).

## **Administration of decentralized computer networks**

### *Problem*

A computer network with a large central system and many small DDP systems, spread over a wide geographical area, is to be administered.

It is probably not necessary to have a trained operator and/or system administrator for each DDP system; instead, you can have assistants, as long as they are backed up by fully trained staff at the central location.

These specialist staff members must, however, be able to intervene directly.

### *OMNIS solution*

OMNIS is loaded in the central system and in each DDP system. The central system has a fully trained computer center team; all administrative functions are carried out from there.

All connections are set up indirectly (see [page 35](#)). OMNIS monitors its own functional integrity and that of the network.

In the DDP systems, the team of administrators is supported by the local message suppression and automatic message answering functions, as well as the administration functions for standard situations implemented in OMNIS-PROP (see the “OMNIS-PROP” manual [2]). The latter ensures a certain degree of autonomy for the DDP systems in the event that the connection to the central system fails.

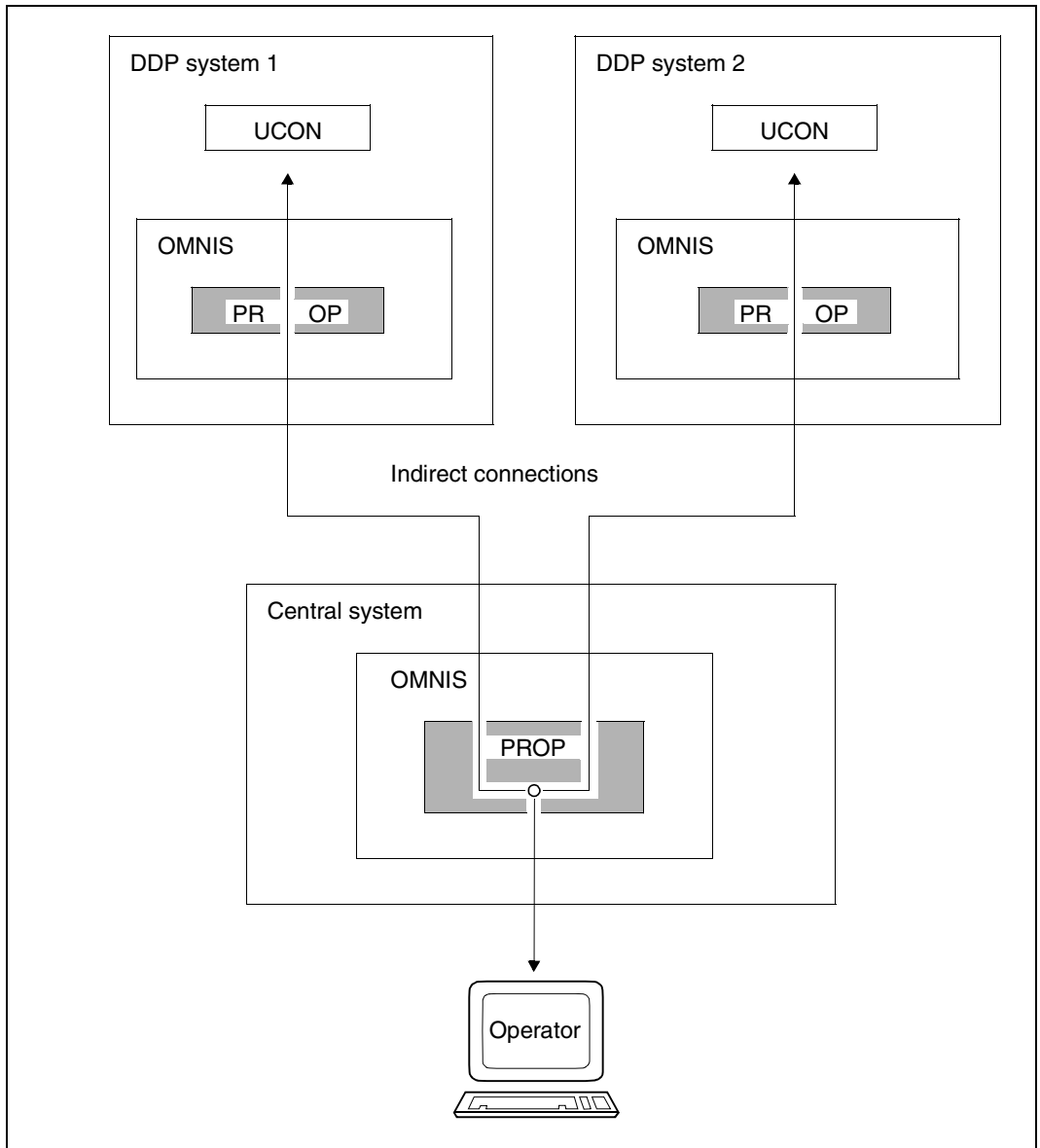


Figure 10: OMNIS in DDP systems



Partners of type UCON must be located in the same computer as the slave OMNIS.



## Administration of guest systems under VM2000

Operation of the individual BS2000 guest systems which run under VM2000 is the equivalent of operating a computer center with several computers. The solutions for operating in multiprocessor systems are similar.

Each guest system has a network connection and also its own host address. Each guest system in a network can thus also be addressed as a separate host. The OMNIS solutions for the operating of the BS2000 guest systems are thus similar to those shown in [figure 7 on page 28](#) and [figure 9 on page 30](#) if hosts 1 to 4 are regarded as guest systems 1 to 4 and systems 1 to 5 are regarded as guest systems 1 to 5.

In addition to BS2000 multiprocessor operation with central operating, there are further possibilities for the use of OMNIS under VM2000:

In the solutions described above ([figure 7 on page 28](#) and [figure 9 on page 30](#)) for OMNIS operation of several computers without VM2000, each system must have, in addition to the OMNIS consoles, a BS2000 console to permit the reception and answering of possible emergency messages.

Under VM2000, this also can be handled by the virtual console of VM2000. This virtual console is an emulation in the VM2000 monitor for the BS2000 consoles of the guest systems and is implemented by means of the DCAM application \$VMCONS in VM2000. A virtual console can be set up for each guest system.



The use of virtual consoles under VM2000 is no longer recommended.

For detailed information on connecting terminals to \$VMCONS via DCAM applications, such as OMNIS, please refer to the "VM2000" manual [9].

## 2.2.3 Implementation in product development

The principal implementation area is the **testing of multi-process systems**

### *Problem*

A program consists of several intercommunicating processes (eventing, DCAM, common memory, FITC).

A number of these processes are to execute simultaneously under the BS2000 debugging aid.

### *Standard solution*

More than one terminal is used at the same time by a single person entering commands from each of the terminals individually.

As a result, a number of terminals, which may not be in the same room (time wasted going from terminal to terminal), are blocked by a single user and the individual terminals are poorly utilized.

### *OMNIS solution*

A single OMNIS terminal is used and a number of connections to \$DIALOG are set up concurrently. Because all processes are managed from one terminal and no time is wasted going from terminal to terminal, the single terminal is better utilized.

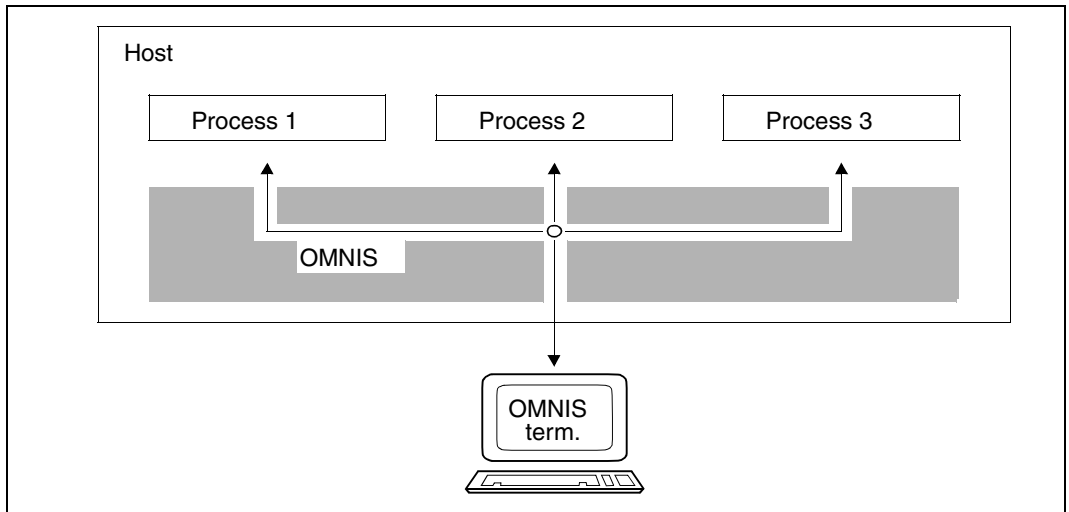


Figure 11: Testing multi-process systems

Another major advantage is the facility for page-turning in the BS2000 debugging aid log.

## 2.3 Fundamentals of operation with OMNIS

The following chapter provides an overview of the many types of connection possible via OMNIS and a summary of the files that OMNIS requires.

### 2.3.1 Configurations and connection options

A distinction is made between OMNIS connections to terminals, to partners, to printers, multiplex connections to UTM, and OMNIS-OMNIS connections.

#### Configuration

The configuration is the sum of all terminals, partners, printers and connections. Configurations can either be “dynamic” or “declared”. A dynamic configuration is created by chance as a result of connections being successively set up and cleared down without advance notice. A “declared” configuration is preplanned and defined with the utility routine OMNISKD.

#### *Dynamic configurations*

Dynamic configurations are used when the circle of users is not defined at the outset (OMNIS as an open-for-all system in development computer centers). Any user can sign on to OMNIS (specifying a password, if necessary) and set up connections to any partners, provided he or she is accepted by these partners.

The configuration changes dynamically, as required.

#### *Declared configurations*

If the circle of users is predefined (e.g. OMNIS as dialog control for a number of tasks in the personnel department), the configuration can be declared to OMNIS. It is then possible to limit the use of OMNIS to particular terminals and their partners, and establish connections automatically.

The administrator can change the declared configuration with the *CONFUPDATE* command while the system is running.

The figure below is an example of a configuration:

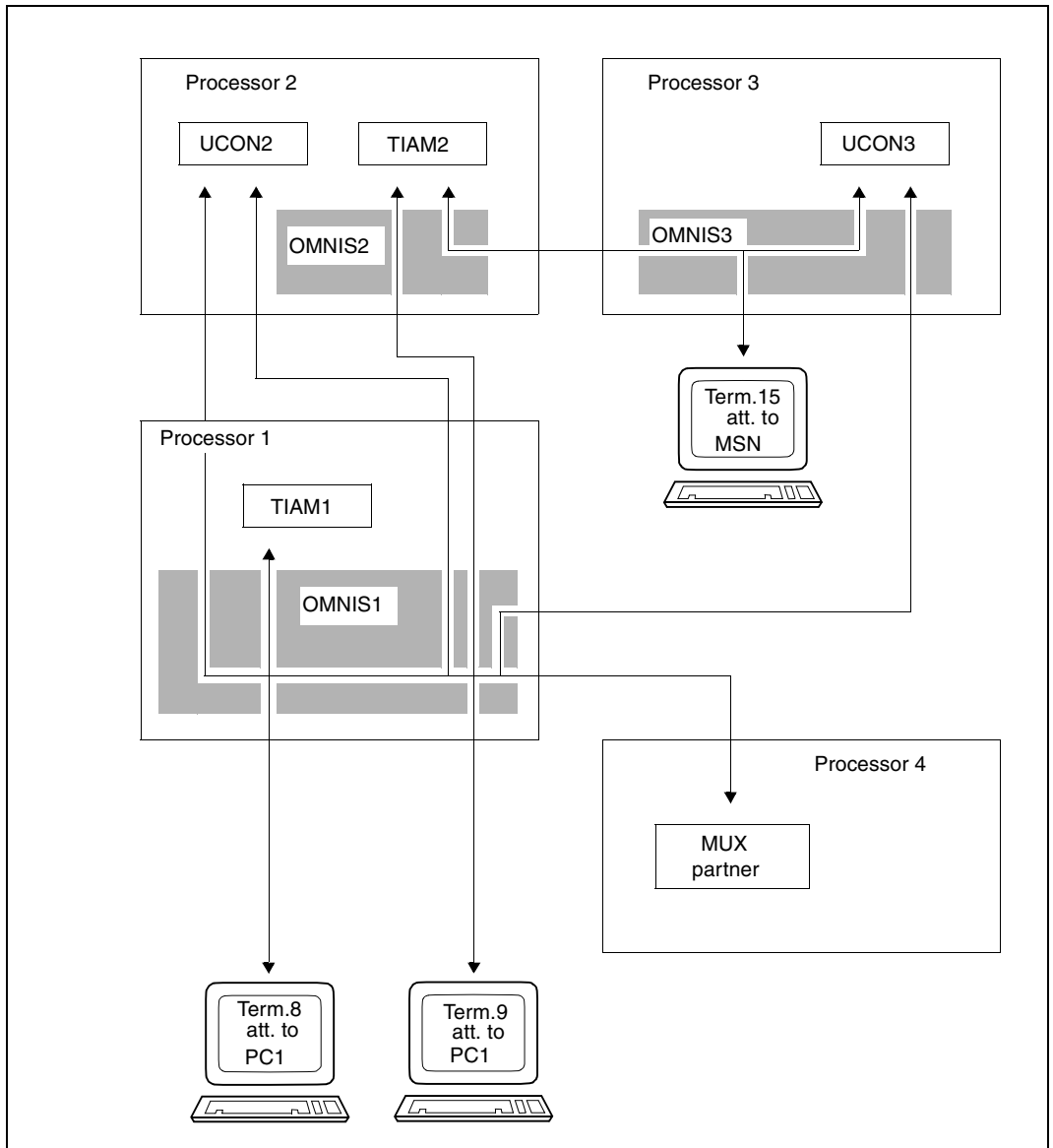


Figure 12: Example of a configuration

The above diagram will be referred to frequently in the text that follows, in order to explain by means of a concrete example the terms that are introduced.

## Connections to terminals

A terminal is connected to OMNIS in transaction mode. In the example on [page 36](#), Term.8 attached to FEP1 is connected to OMNIS1 in Processor 1.

### *Terminal types*

For OMNIS a “terminal” may be:

- a terminal or a terminal emulation
- a program (an DCAM/CMX/ISO application)
- a BS2000 console

## Connections to partners

OMNIS sets up connections to the partners with which communication is to take place from the terminal; connections to partners can be intra-computer or inter-computer.

Each partner is assigned a permanent partner address code (PAC). This can be freely selected by the user, consists of one through four alphanumeric characters and identifies the partner uniquely for a particular terminal. The partner address code can be assigned to the partner either statically when the partner is declared or dynamically at connection setup.

### *Partner types*

Via OMNIS connection setup and communication with partners of the following types is possible:

- TIAM (the partner is the system application \$DIALOG)
- UCON (the partners are the system applications \$CONSOLE or \$VMCONS as virtual console)
- DCAM (the partner is a DCAM/CMX application)
- UTM (the partner is a UTM application)

### *Partner groups*

Situations often occur in which actions are to be initiated which affect more than one partner in the same way, for example:

- a user wishes to set up three connections to \$DIALOG and start all three processes with the same LOGON command
- an operator wishes to know if there are any replies outstanding from any of the systems for which he or she is responsible

In order to simplify such actions, OMNIS provides the option of grouping a number of partners together to form a partner group and addressing all partners in the group with a common group address code (GAC). This can be freely selected by the user, consists of one through four alphanumeric characters and identifies the group uniquely for a particular terminal. The group address code can be assigned either statically when the group is declared (with *DECLARE-GROUP*) or dynamically with the *GROUP* command. These two commands are also used to specify which partners belong to the group.

OMNIS automatically sets up and administers default partner groups containing all partners of a specific partner type; see [page 89](#).

Inter-computer partner connections can be direct or indirect.

#### *Direct partner connections*

The connection to the partner is set up directly, i.e. the OMNIS to which the terminal is linked is connected directly to the communication partner in another computer.

In the example on [page 36](#), OMNIS1 in Processor 1 is connected directly to UCON2 in Processor 2 twice. The connections “belong to” terminals Term.8 and Term.9 attached to FEP1.

#### *Indirect partner connections*

The connection to the partner is set up indirectly via an OMNIS loaded in another computer. The OMNIS to which the terminal is linked is connected to an OMNIS in the remote computer via an OMNIS-OMNIS connection (see below), and the OMNIS in the remote computer is connected to the communication partner.

In the example on [page 36](#), terminal Term.9 attached to FEP1 is connected indirectly to partner UCON3 in Processor3 via OMNIS1 in Processor1 and OMNIS3 in Processor3.

The OMNIS to which the terminal is linked is known as the **master**, the OMNIS in the remote computer the **slave**. These terms refer only to a single indirect connection. One OMNIS can therefore be master for some indirect connections and slave for other indirect connections at the same time.

In the example on [page 36](#), OMNIS1 is both master (for [Term.15,TIAM2]) and slave (for [Term.9,UCON3]) at the same time.



Partners must be located in the same computer as the slave OMNIS.

## OMNIS-OMNIS connections

OMNIS-OMNIS connections are used by OMNIS for a number of purposes:

- Indirect connections

Messages can be exchanged between partners and terminals via indirect connections.

- Reciprocal monitoring

An exchange of messages via OMNIS-OMNIS connections (*ACK-ACK* traffic) takes place at regular intervals so that faults (in the network, in the computer, in OMNIS) can be recorded. Faulty OMNIS-OMNIS connections are reported to all terminals from which an indirect connection currently exists to the other OMNIS.

- Decentralized administration

Commands can be entered at one OMNIS for any other OMNIS if a connection to this OMNIS exists. The commands and messages are transported via an OMNIS-OMNIS connection.

OMNIS-OMNIS connections must be declared in the OMNIS configuration, but this does not apply to indirect connections. It is not possible to dynamically declare OMNIS-OMNIS connections, but it is possible to modify a declaration using the *CONFUPDATE* command (see [page 192](#)).

### *Comparison of direct and indirect connections*

If OMNIS is loaded in both computers and all messages between them are transported via the OMNIS-OMNIS connection, the following advantages ensue:

- Some of the processing functions (message suppression, message answering, exit routines) can be delegated to the slave (message suppression and answering take place locally, i.e. no unnecessary messages are transported through the network). In the case of direct connections, all functions are resident in the master.
- The connections from the slave to the partners are intra-computer; only the OMNIS-OMNIS connection is implemented via the network. A loss of this connection has no repercussions on the connection from slave to partner. Thus all functions delegated to the slave (message answering, exit routines for implementing a service with just a few operators) continue as usual. The remote computer is to a large extent autonomous.

The data traffic on an OMNIS-OMNIS connection is not backed up; messages arriving at the slave while the connection is down are lost.

In the case of indirect connections, it must be noted that all messages are transported via a single connection without flow control. This transport connection may become overloaded if messages arrive in rapid succession.

## Multiplex connections

A number of OMNIS-partner connections to a UTM application can be set up via a transport connection.

A **transport connection** is a connection between two programs or a program and a terminal via which messages are exchanged. A transport connection has a defined beginning (connection setup) and a defined end (connection cleardown), and is known to the transport system.

A **session** is one of several clearly distinguished data streams that are directed over one transport connection. A session has a defined start (session setup) and a defined end (session cleardown) and is not known to the transport system. In the specific case of OMNIS and UTM, a session is defined as a communication association between a UTM application and an OMNIS terminal which starts with the (logical) setup of the session and ends with the cleardown of the session.

Each multiplex connection must be declared in the OMNIS configuration. It is not possible to dynamically declare multiplex connections, but it is possible to modify a declaration using the *CONFUPDATE* command (see [page 192](#)).

### *Advantages of multiplex connections*

- The transport system only needs to recognize and administer a single connection. A number of sessions can, however, be conducted via this connection. This reduces the system resources (e.g. class 3 and class 4 memory, PCBs, etc.) and OMNIS resources (e.g. message buffers) required drastically.
- OMNIS and UTM exchange special information via multiplex connections, using their own protocol. This permits, for example, secure partner switching and the transfer of data to exit routines (see the “Administration and Programming” manual [1]).

It is also possible to maintain more than one parallel multiplex connection between OMNIS and UTM in order to distribute the load (see the “Administration and Programming” manual [1]).



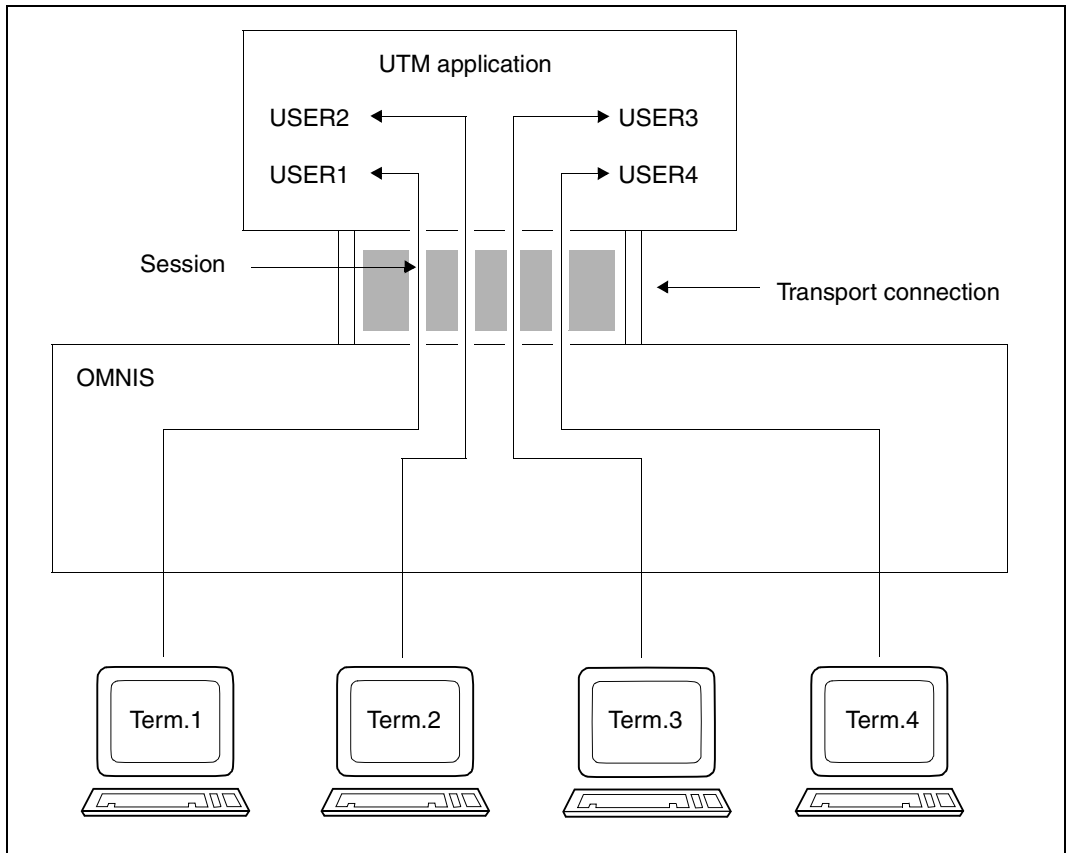


Figure 13: Schematic representation of a multiplex connection

### 2.3.2 Files for OMNIS

During execution, OMNIS requires a number of files which must either be made available by the administrator before the start or be generated by OMNIS itself.

With regard to files specifically for OMNIS-MENU, refer to the “Administration and Programming” manual [1].

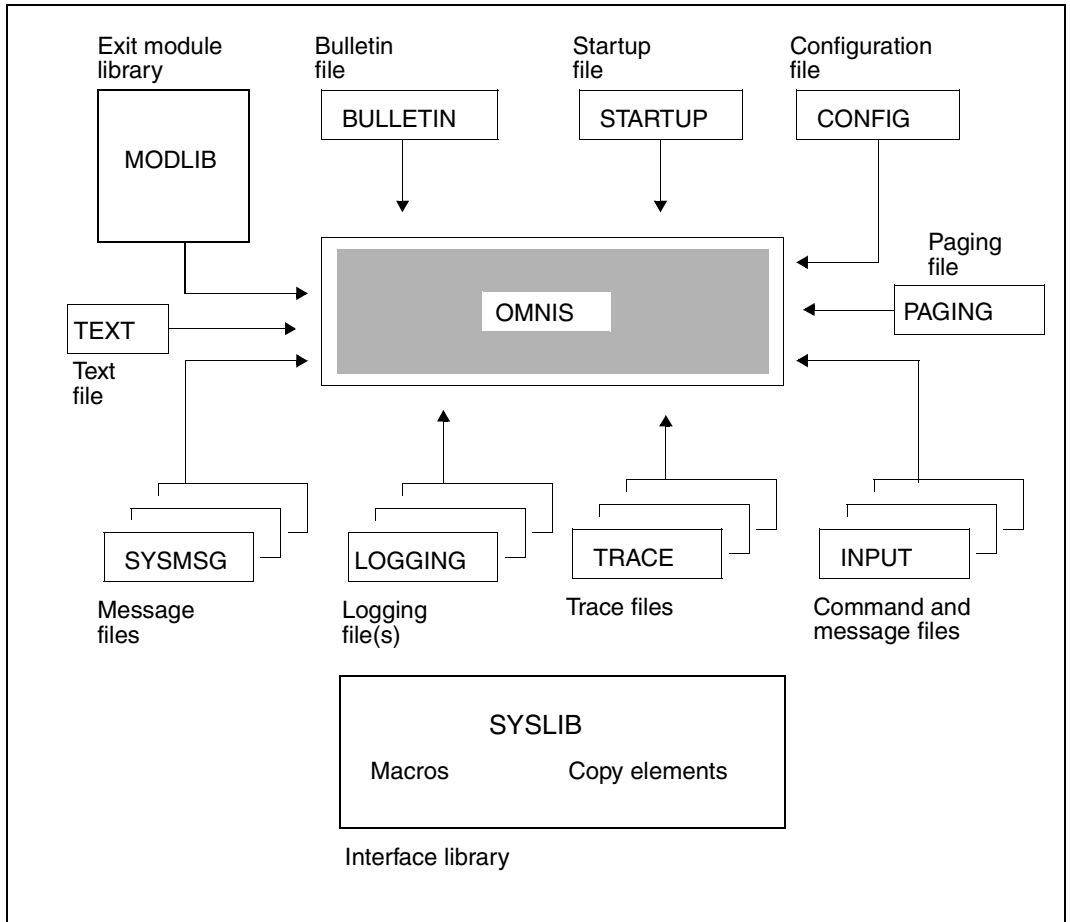


Figure 14: Files required by OMNIS

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

The **bulletin file** is a message file whose contents are displayed at each OMNIS terminal as it signs on to OMNIS.

The **logging file** is used when logging is switched on.

The **paging file** is used to buffer large amounts of data when there is a message bottleneck.

The **trace file** is used to save trace entries when the trace function is active.

The **configuration file** contains a configuration (generated using the utility routine OMNISKD) which is read in automatically by OMNIS when initiated, and set up if necessary.

The **command files** contain messages and user-specific OMNIS 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** contain the text of messages that 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 the OMNIS messages and HELP texts in German and English.



OMNIS as a rule denies access to files which have been migrated by HSMS (**H**ierarchical **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 startup phase of OMNIS, during which the system waits for files (configuration file, startup file) to be read in.

### 2.3.3 OMNIS diagnostics

A range of diagnostic options is available to the administrator for online 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.

#### *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 TRACE* command.

For more detailed information on tracing, please refer to the appropriate sections for the relevant commands in this manual and to the "Administration and Programming" manual [1].



---

## 3 Connection setup and cleardown

This chapter describes the setting up and clearing down of connections, loss of connection and taking over connections.

### 3.1 Connections to terminals

This section deals with connections to terminals of various types.

#### Setting up the connection

Connection setup between OMNIS and the terminal is dependent on the type of “terminal” (terminal, BS2000 console, DCAM/CMX application, ISO application) and whether the terminal is declared or not.

#### *Undeclared terminals*

Undeclared terminals must always sign on to OMNIS by predialog. OMNIS cannot set up the connection. The OMNIS administrator can prevent undeclared terminals from signing on to OMNIS by means of the command `SET OPNCON=DECLARED`.

In order to set up a connection, OMNIS needs to know whether the communication partners come from the NEA world or from the ISO world. OMNIS uses the NEA or ISO data according to what is specified in the *TRANSPORT-PROTOCOL* operand in the *DECLARE-TERMINAL* command. Therefore, for undeclared terminals where *TYP=APPL*, the call *OMEOPTRM* with *OMEROPAP* from EXIT routines is rejected; see the “Administration and Programming” manual [1].

*Declared terminals*

Declared terminals can either sign on to OMNIS by predialog or are connected automatically when OMNIS is activated. This is determined by the declaration:

```
DECLARE-TERMINAL CONNECT={LOGON/START}
```

*CONNECT=LOGON* means that the terminal signs on to OMNIS by predialog; with *CONNECT=START* the connection to the terminal is set up by OMNIS.

OMNIS sets up a connection as soon as it is activated. If the setup is unsuccessful (e.g. because the terminal is switched off), this is equivalent to a connection that is lost immediately after being established (see page [page 74](#)). The terminal must then sign on to OMNIS by predialog.

The terminals must, in the context of the NEA/ISO world, always be attached to the correct OMNIS, i.e. only ISO applications attached to ISO-OMNIS.

If the terminal is not “correctly attached”, then although the terminal can issue a connection request to ISO-OMNIS without an error being indicated, no exchange of messages is possible.

Parallel connections from and to ISO applications are rejected by OMNIS.

*Authorization check*

The OMNIS administrator can define a password in the ENTER file which must be entered at the terminals in the predialog when signing on to OMNIS or setting up a connection to DCAM/CMX programs.

*Acknowledging the connection setup*

OMNIS acknowledges that a terminal is connected by issuing the message OMS0001, outputs the contents of the bulletin file, and awaits input from the terminal.

*Terminal identifier (TID)*

Each terminal is assigned a sequence number when it signs on to OMNIS. This number, the terminal identifier (TID), identifies it uniquely “OMNIS-wide”. The TID is communicated to the terminal user in the connection acknowledgment, and must be specified in certain commands (it is optional for others). The TID thus has a similar function to the TSN in BS2000.



### Clearing down the connection

The connection to a terminal is cleared down with the *HALT* command:

$$\text{HALT [HOLD=}\left\{\begin{array}{l} \text{NO} \\ \text{YES, OPASS=}\left\{\begin{array}{l} \text{password} \\ ? \end{array}\right\} \end{array}\right\}. ]$$

If *HALT HOLD=NO* is specified, all existing connections to partners, regardless of their *HOLD* settings, are cleared down before the connection to the terminal itself is cleared down.

The *HOLD=YES/AUTO* specification prevents automatic clearing down of existing connections to partners: With *HALT HOLD=YES*, the connections to all partners for which *HOLD=YES/STD* is set are retained; with *HOLD=AUTO*, the connections to all partners are retained, regardless of their *HOLD* settings.

In the case of a **non-unique user** and *SECURITY LEVEL≠LOW*, the specification *HOLD=AUTO* is ignored; *HOLD=NO* always applies. The uniqueness of the user is checked by OMNIS-MENU.

In the case of a **unique user** and *SECURITY LEVEL≠LOW*, only the connections of Kerberos-authenticated partners (previously chip card) are cleared down.

### 3.1.1 BS2000 consoles used as terminals

BS2000 consoles used as terminals must always be declared with the *DECLARE-TERMINAL* command; the connection is always set up from OMNIS (immediately after OMNIS has been activated).

A BS2000 console can only interoperate with OMNIS in line mode since the console hardware does not allow formats to be output.

Messages and commands are entered using one of the special commands */OMNIS* or */O*, which must be taken into account when the system is generated (see the “Administration and Programming” manual [1]). Messages from partners and the system are output in the standard UCON format.

The following represents a dialog with OMNIS conducted at a console. *bnam* is the authorization name, which is defined in the *DECLARE-CONSOLE* statement using the *CMMSG* operand (see the “Administration and Programming” manual [1]).

```

/O OPNCON PTN1
%bnam- cc.hhmmss @: OMS0012 'OPNCON' COMMAND ACCEPTED ...
%bnam- cc.hhmmss @: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY ...
%bnam- cc.hhmmss PTN1:% JMS0150 ... PLEASE LOGON
/O PTN1:LOGON ...
%bnam- cc.hhmmss PTN1:% JMS0066 JOB ACCEPTED ...
/O PTN1:START-PROGRAM $LMS
%bnam- cc.hhmmss PTN1:% BLS0500 PROGRAM ... LOADED
/O LIB FILE= ...
.....
/O END
%bnam- cc.hhmmss PTN1:% LMS0311 ...
/O @:CLSCON PTN1
%bnam- cc.hhmmss @: OMS0055 CONNECTION TO PARTNER ... CLEARED DOWN ...

```

Since the maximum length of output to a console is 128 characters, OMNIS divides up longer outputs over several lines to enhance the clarity of the output. The points at which the output is divided are determined by the line length of the console and/or by any NL characters in the message:

```

%bnam- cc.hhmmss A: 1. THIS IS A VERY LONG 0
%bnam- cc.hhmmss A:UTPUT TEXT
%bnam- cc.hhmmss A: 2. SHORT TEXT
%bnam- cc.hhmmss A: 3. THIS IS ANOTHER VERY
%bnam- cc.hhmmss A:LONG TEXT WHICH APPEARS ON MORE
%bnam- cc.hhmmss A:THAN TWO LINES
%bnam- cc.hhmmss A: 213.

```

It is advisable to work with only one partner at a time from a BS2000 console and to deactivate partner address code output.

### 3.1.2 DCAM/CMX applications used as terminals

DCAM/CMX applications used as terminals must be declared with the *DECLARE-TERMINAL* statement. Connection requests for undeclared DCAM/CMX applications are rejected. Connections are set up and cleared down just as they are for declared terminals. For communication between a DCAM/CMX application and OMNIS, a specific protocol must be followed. The protocol used is determined with the operand *USER-PROTOCOL=NO|OMNIS|VTSUCB* in the *DECLARE-TERMINAL* statement.

#### Example of a dialog (USER-PROTOCOL=OMNIS)

Outputs from OMNIS:

consist of a header, which contains the four-byte edit options, and the message in the same format it has when output at the terminal.

```
|EDIT | Message |
```

```
|EDIT | LL+NBZ+NK | Message |
```

The format of the EDIT bytes is described in the CUPAB macro (see the “Executive Macros” manual [4]).

Inputs to OMNIS:

Line inputs must contain the function key code as the first characters:

```
|FK | Message |
```

In the case of format inputs, the message header must be preceded by a function key code X'FF':

```
|FF | NBZ+NK | Message |
```

### 3.1.3 ISO applications used as terminals

Applications on a network can also be connected with OMNIS via ISO transport connections.

Prerequisites for use of ISO applications as a terminals:

1. You open two OMNIS applications, one for the communication partners from the NEA world, one for those from the ISO world (see start parameters). The communication partners must sign on to the appropriate OMNIS application. OMNIS then sets the NEA or ISO data accordingly.  
If the terminal is not “correctly attached”, then although the terminal can issue a connection request to ISO-OMNIS without an error being indicated, no exchange of messages is possible.
2. In the case of the *DECLARE-TERMINAL* command, the administrator must request the ISO function by means of the operand *TRANSPORT-PROTOCOL=ISO*.
3. You can have the ISO/NEA attribute of a communication partner displayed by using the *INF* command.
4. The *MORE-DATA* function from the ISO world is not supported by OMNIS.
5. The ISO/NEA attribute is made visible at the exit interface.
6. Parallel connections from ISO applications from/to OMNIS are neither set up nor accepted.

## 3.2 Connections to partners

This section describes the special points to be observed when setting up and clearing down connections to partners of various types.

With OMNIS it is possible for users to prove their identity to applications using Kerberos-authentication (previously chip card).

### Setting up a connection

Connection setup between OMNIS and a partner is always initiated by OMNIS (OMNIS cannot accept connections requested by a partner). The time of connection setup depends on whether the partner is declared or not.

#### *Undeclared partners*

The connection to an undeclared partner is set up using the *OPNCON* command. All the necessary operands must be specified in this command (e.g. PAC, partner name and type).

For declared terminals, connections to undeclared partners can be prevented by declaring the terminal with `DECLARE-TERMINAL OPNCON=DECLARED`.

#### *Declared partners*

At what time the connection is set up to declared partners is determined by the declaration with `DECLARE-PARTNER pac, CONNECT=LOGON/START/OPNCON`.

For *CONNECT=OPNCON* the connection must be set up explicitly by the user, by means of the *OPNCON* command. Since all the necessary information is known as a result of the declaration, it is enough to indicate the partner by means of the appropriate partner address code; no further operands are needed. If other operands are nevertheless entered, they alter the specification of the declaration.

If the partner is declared with `DECLARE-PARTNER pac, CHANGE=NO`, the declared values cannot be changed.

Partners declared with *CONNECT=LOGON* are connected automatically once the connection to the terminal is (successfully) established (in the predialog or when OMNIS is activated, depending on how the terminal is declared).

Declaring the partner with *CONNECT=START* has the effect that the connection to the partner is set up automatically when OMNIS is activated.

If both the terminal and its partners are declared with *CONNECT=START*, the connections to the partners are set up once the connection to the terminal has been established. If connection setup to the terminal is unsuccessful, the connection to the partners can only be established if *HOLD=YES* (which also means that *SECURITY LEVEL=LOW* must have been set).

*Direct connections*

Direct connections are the default case, i.e. a connection to a partner is set up directly if no further specifications are made:

```
OPNCON          pac [,ROUTE=DIRECT] [,PRONAME=processor]
DECLARE-PARTNER pac [,ROUTE=DIRECT] [,PRONAME=processor]
```

If *PRONAME* is not specified, OMNIS assumes that the partner is in the same computer as OMNIS (intra-computer connection).

The connection is set up asynchronously in two stages:

1. OMNIS passes on the connection request to the data communication system, which either accepts (OMNIS issues the message OMS0012) or rejects it (OMNIS issues the message OMS0005)
2. The data communication system informs OMNIS whether the connection could be set up (OMNIS issues the message OMS0014) or not (OMNIS issues the message OMS0005 with the *dcam-errorcode*). The latter is the case when, for example, the partner has rejected a connection request or there is a disruption in the network (the various causes are encoded in *dcam-errorcode* and can be queried with *HELP OMY dcam-errorcode* (or *HELP YDD dcam-errorcode*).

A relatively long period may elapse between the two stages, especially in the case of long transmission paths between OMNIS and the partner. The terminal is not, however, blocked. Messages can be sent to other partners or received messages from them, or further OMNIS commands can be entered.

Until the connection is established positively, no messages can be sent to or received from the partner in question.

The following figure illustrates the individual steps involved in setting up a direct connection to a partner.

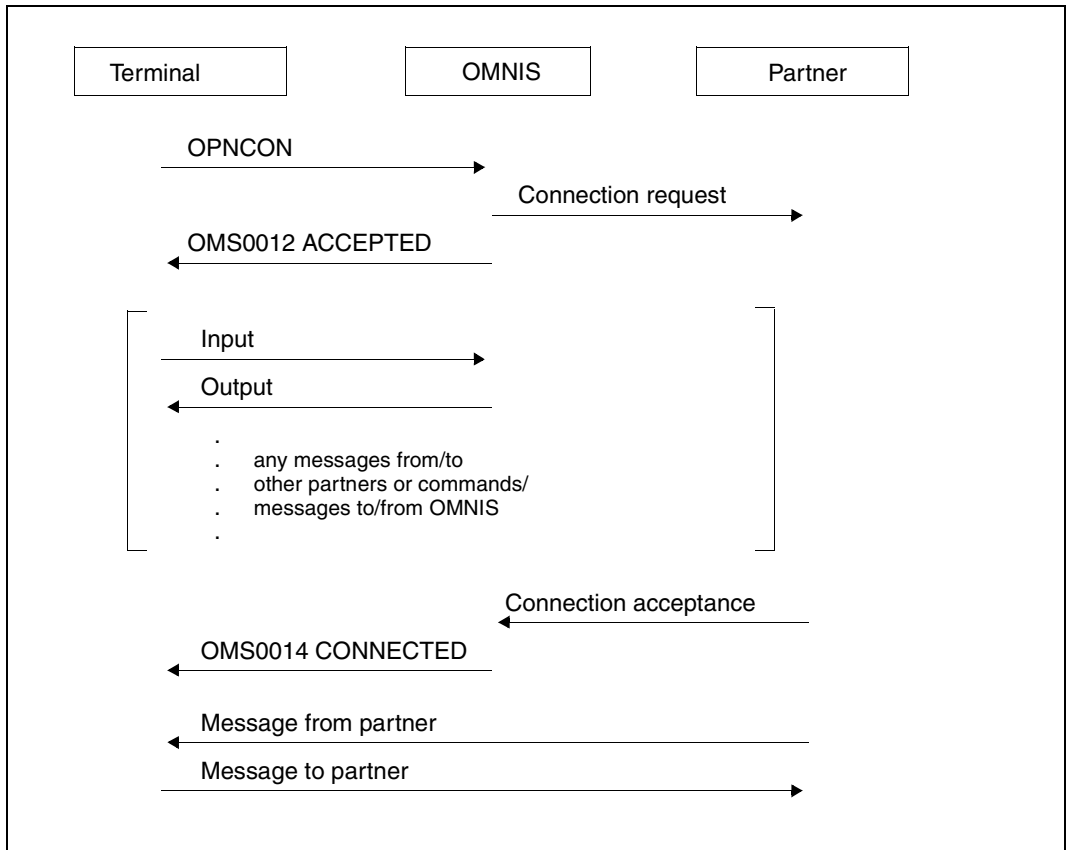


Figure 15: Setting up a direct connection

*Indirect connections*

If an indirect connection to a partner is desired, it must be specified explicitly:

```
OPNCON          pac, ROUTE=INDIRECT, PRONAME=processor  
DECLARE-PARTNER pac, ROUTE=INDIRECT, PRONAME=processor
```

Indirect connections can be set up both within one computer (intra-computer) and to another remote computer. The partner computer must be explicitly specified and an OMNIS-OMNIS connection to this partner must be declared and exist. Dynamic OMNIS-OMNIS connections are not possible.

An indirect connection is set up in several stages:

1. The master acknowledges the connection request with the message OMS0012.
2. If the OMNIS-OMNIS connection does not yet exist, it is established.
3. The desired partner must be located in the processor of the slave. The connection request is passed to the slave, which now sets up a connection to the desired partner.
4. The slave reports the result of connection setup (positive or negative) back to the master.

Until the connection is acknowledged by the slave as positive, no messages can be sent to or received from partners.

The terminal is not blocked between the acceptance and execution acknowledgments. A number of indirect connections can be set up at the same time. It is not necessary to wait for each execution acknowledgment.



The following figure illustrates the individual steps involved in setting up an indirect connection to a partner.

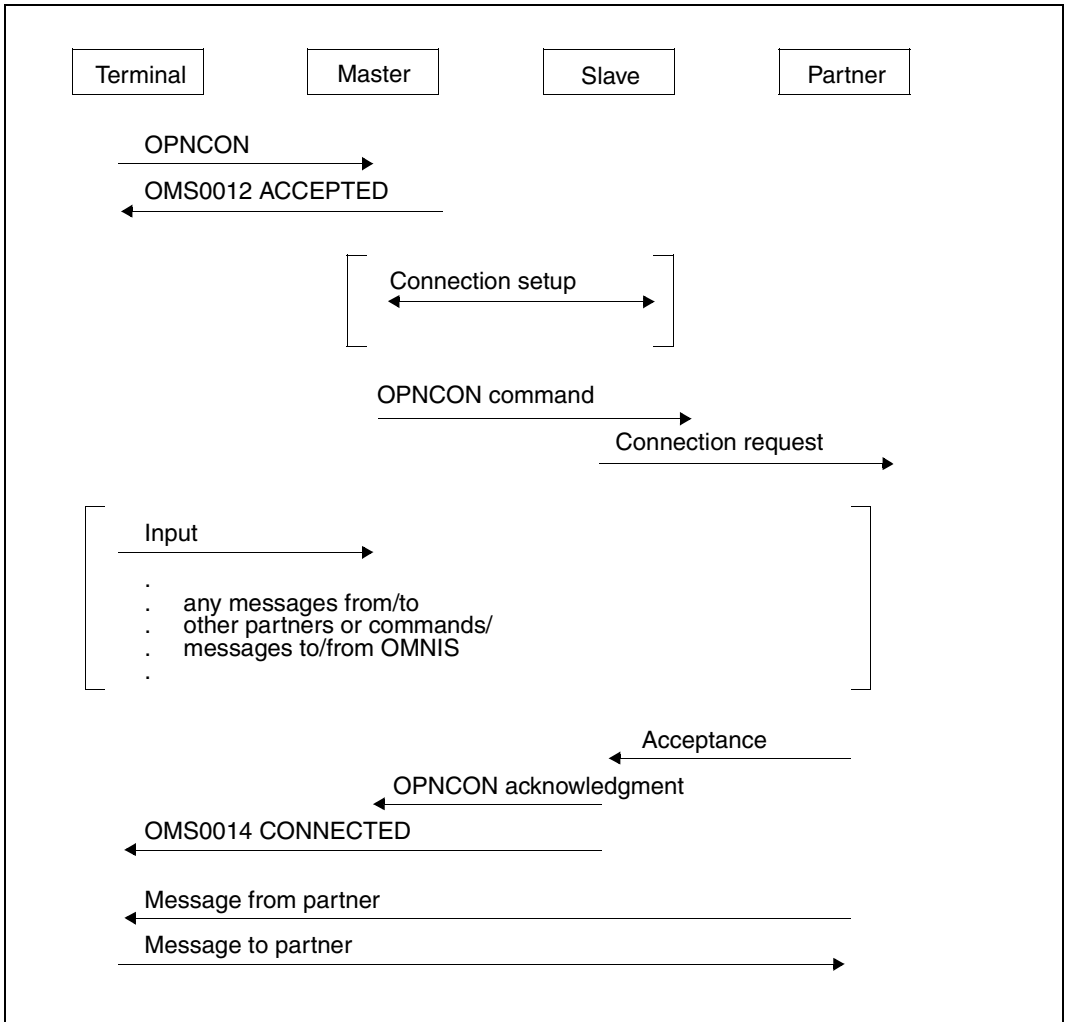


Figure 16: Setting up an indirect connection

Depending on the partner type, different specifications are necessary for successful connection setup. The specific entries for the individual partner types are described in detail on [page 59](#).

*Partner identifier (PID)*

OMNIS assigns each partner a sequence number, the PID, which uniquely identifies that partner throughout the OMNIS network. This PID is passed to the terminal operator in the message `OMS0012` and must be specified in certain commands (it is optional in others). The PID thus has a similar function to the TSN in BS2000.

*Authorization check*

In general, the communication partners provide their own protection; OMNIS does not circumvent the partners' authorization checks. In the case of declared configurations, it is also possible to define the partners with which a user can work from a certain terminal.

**Clearing down a connection**

The connection to a partner is cleared down using the command `CLSCON pac`.

The communication partner is informed of the loss of the connection. The individual partners react to this differently, as described in detail below for the various partner types.

The connection can also be cleared down by the partner itself (e.g. on `LOGOFF`, `KDCOFF`). From the point of view of OMNIS, this is equivalent to a loss of connection (see [page 76](#)).

*Undeclared partners*

All administration data is released following connection clear-down. The partner is no longer known to OMNIS and the partner address code is deleted.

*Declared partners*

The partner is set to "inactive", but the declared data is retained. The partner is still known to OMNIS. A new connection can be set up to this partner at any time using `OPNCON`.

### 3.2.1 TIAM partners

The partner is the system application \$DIALOG (timesharing mode).

OMNIS supports identification of the user to TIAM applications with the aid of Kerberos-authentication (previously chip card).

XHCS is not supported for the \$DIALOG application where *TYP=DCAM* and *PROTOCOL=DSSIM*.

#### Connection setup

It is sufficient to specify the partner address code in order to set up a connection to \$DIALOG (partner type *TIAM* is the default):

```
OPNCON          pac [,TYP=TIAM]
DECLARE-PARTNER pac [,TYP=TIAM]
```

The name of the partner (\$DIALOG) is implicit in the partner type *TIAM*; another name can, however, be set by specifying the *PTNNAME* operand.

With terminal type 9763, the partner for the next input is set automatically. In this case, it is thus not necessary to specify the partner address code.

#### Connection cleardown

If the connection is cleared down by OMNIS (*CLSCON* command), the *TIAM* partner is informed of the loss of the connection. Any interactive process which is running is aborted with the message *EXC0736*.

The connection can also be cleared down from *TIAM*. OMNIS reacts to the *TIAM* command *LOGOFF* by issuing the message *OMS0006* (connection lost). The entry *LOGOFF BUT* causes a new logon prompt (*PLEASE LOGON*) to be displayed.

#### Station name in outputs from the SHOW....STATUS command

The following commands output the name of the real terminal:

```
/SHOW-USER-STATUS INFORMATION=ENVIRONMENT
/SHOW-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=tsn)
```

The parameter *INFORMATION=ENVIRONMENT(ORIGINAL)* of these commands outputs the real station name, while the parameter *INFORMATION=ENVIRONMENT(APPLICATION)* displays the name of the auxiliary application. The name of the auxiliary application is assigned with the *APPNAME* operand (command *OPNCON*, *DECLARE-PARTNER*). If no assignment is made, OMNIS assigns a name itself (see start parameter *APPLICATION* in the "Administration and Programming" manual [1]).

**Dialog under SDF**

For a dialog via OMNIS, *GUIDANCE=EXPERT* is always set after *LOGON*, regardless of any other specification in the global information of the activated syntax files. When a user syntax file is activated or switched, *EXPERT* mode is also set. It is, however, possible to switch to another dialog with the *MODIFY-SDF-OPTIONS* command or statement.

**Maximum message length**

The maximum length of messages that can be sent from TIAM to OMNIS is about 4000 characters.

### 3.2.2 UCON partners

The UCON partner is either the system application \$CONSOLE (system operating) or the VM2000 application \$VMCONS (a virtual console).

Operator logon is supported only for dynamically authorized applications.

In the event of “reply suppression” a maximum of 1000 queries can be waiting for replies OMNIS-wide because no more space is available for entries for authorization checking. If the number of 1000 simultaneously open queries is exceeded, OMNIS notifies all administrator terminals of this with message OMS0238.

Until one of the open queries has been answered, any further queries that arrive can only be answered from the terminal for which *REPLY = ALL* applies.

#### Connection setup to the \$CONSOLE system application

UCON distinguishes, on the basis of the connection message, between statically and dynamically authorized applications. The connection message for dynamically authorized applications includes the prefix \$CONSOLE.

##### *Statically authorized application*

The partner type (UCON) and the authorization name must be specified explicitly in order to set up a connection to the system application \$CONSOLE using *OPNCON* or *DECLARE-PARAMETER*.

The *CMSG* connection message is used to check the authorization, and contains an authorization name and an authorization password:

OPNCON pac, TYP=UCON,

$$CMSG = \left\{ \begin{array}{l} \text{authorizationname,password[,DISCON=}\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \text{]} \\ ? \end{array} \right\}$$

DECLARE-PARTNER pac, TYP=UCON,

$$CMSG = \text{authorizationname,password[,DISCON=}\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \text{]}$$

The name of the partner (\$CONSOLE) is implicit in the partner type UCON; another name can, however, be set by specifying the operand *PTNNAME*.

If only a ? is specified for *CMSG*, the connection message is requested by OMNIS in blanking mode.

You can use *DISCON* to define how UCON should behave in the event of message congestion. Message congestion occurs when an application cannot receive message quickly enough.

- *DISCON=YES* is the default. UCON breaks the connection to the application in the event of message congestion. Undelivered messages are deleted. This setting must be used for all applications which cannot run properly without uninterrupted delivery of messages.
- *DISCON=NO* must be used only for applications (such as console emulators) which can continue to run properly even without uninterrupted message delivery. UCON does not break the connection to the application in the event of message congestion. Undelivered messages are deleted, however, and UCON sends a message to that effect to the application.

OMNIS in some case modifies the connection message specified with the *CMSG/LMSG* operand by appending additional operands (such as *PROTVERS*). This is primarily relevant for system exits (also refer to the “System Exits” manual [5]).

See the “Administration and Programming” [1] manual for more details on authorization checks.

#### *Dynamically authorized application*

The following functions are available for dynamically authorized applications only:

- operator logon
- identification by Kerberos-authentication (previously chip card) with a UCON partner

For this type of connection setup, the connection message *CMSG* must contain the prefix *\$CONSOLE*.

OPNCON pac, TYP=UCON,

$$CMSG = \left\{ \begin{array}{l} \$CONSOLE [ , OPERID=name ] [ , password ] [ , DISCON = \left\{ \begin{array}{l} YES \\ NO \end{array} \right\} ] \\ ? \end{array} \right\}$$

DECLARE-PARTNER pac, TYP=UCON,

$$CMSG = \$CONSOLE [ , OPERID=name ] [ , password ] [ , DISCON = \left\{ \begin{array}{l} YES \\ NO \end{array} \right\} ]$$

The *OPERID* operand does not correspond to any of the names in the authorization name table; instead, it identifies the administrator who is running the user program which is logging on (dialog connection).

If *OPERID* is not specified, UCON issues its *PLEASE LOGON* prompt, expecting one of the following responses:

```
/SET-LOGON-PARAMETERS USER-ID=name [,PASSWORD=C'password']
/LOGON name [ , ,C'password']
```

If you enter the operator identification later on in a *LOGON* command, you must also include a password if one is required. The password specified with the *CMSG* operand is ignored.

All other operands must be specified as described above (for statically authorized applications).

### Connection setup to the \$VMCONS VM2000 application

\$CONSOLE is the default partner name for partner type UCON. A different partner can be set by specifying the *PTNNAME* operand.

This means that the partner type UCON (*TYP=UCON*) and the partner name (*PTNNAME=\$VMCONS*) must be explicitly specified for connection setup to \$VMCONS with *OPNCON* or *DECLARE-PARTNER*.

Connection setup to \$VMCONS via the UCON partner type is worthwhile only for one or more VC dialogs (operating guest systems via virtual console). The subsequent dialog should therefore be started with the command */BEGIN-VM-DIALOG ... ,TYPE=\*VC*. In this case, the UCON-specific functions for message tables and color control for the virtual console are available via the UCON partner type.

For an admin dialog (administering virtual machines), the connection setup to \$VMCONS should be made with *OPNCON* or *DECLARE-PARTNER* with the partner type DCAM (*TYP=DCAM*) and partner name (*PTNNAME=\$VMCONS*). In an admin dialog, the message traffic (command input/output, message output) is conducted without a UCON-specific header. Connection setup with the UCON partner type would therefore cause wrong interpretations in the color control and in the message tables.

If several VC dialogs are conducted via a connection with the UCON partner type, all connection messages from a virtual console must carry the prefix of the relevant console so that the message can be uniquely assigned. This is the only way of assigning a message to a specific guest system. OMNIS also needs this prefix to decide which guest system to send the next input to when answers arrive from message tables.



For the reasons mentioned above, the prefix of a virtual console must never be deactivated if several VC dialogs are conducted via a connection with the UCON partner type. The dialog with the virtual console must therefore be started with the VM2000 command */BEGIN-VM-DIALOG ... ,TYPE=\*VC(OUTPUT-PREFIX=\*YES)*.

To make sure that there is a clear virtual console / guest system assignment, you are recommended to only ever conduct one VC dialog over a connection with the the UCON partner type.

A connection password must be specified with *CPASS* if a *\$VMCONS* password has been defined:

```
OPNCON pac, TYP=UCON, PTNNAME=$VMCONS, CPASS={ password }
                ?
```

```
DECLARE-PARTNER pac, TYP=UCON, PTNNAME=$VMCONS, CPASS=password
```

If only a *?* has been specified for *CPASS* in the *OPNCON* command, OMNIS prompts the password in blanking mode.

See the “Administration and Programming” [1] manual for more details on authorization checks.

*Example of a VC dialog via \$VMCONS:*

```
@:/o vm1,typ=ucan,ptn=$vmcons,pro=g133h600,cp='ssss'
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=VM1, PID= 68, APPNAME=(OML00068/
G133H600))
@:/
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY
(PARTNER=($VMCONS/G133H600), PAC=VM1)
vm1:/begin-vm-dialog vm-id=4,pass='ssss',type=*vc
VM1: % VMS3000 VM2000 COMMAND /BEGIN-VM-DIALOG FOR VIRTUAL MACHINE (4,DVA13)
PROCESSED SUCCESSFULLY
VM1: % NBR0740 COMMAND COMPLETED 'BEGIN-VM-DIALOG'; (RESULT: SC2=000,
SC1=000, MC=CMD0001); DATE: <date>/sta msg
VM1:GS04: + UCO-000.132025 % EXC0031 NO MESSAGE OUTSTANDING ON THE CONSOLE
VM1:GS04: ! UCO-000.132025 % NBR0740 COMMAND COMPLETED 'STA' ; (RESULT:
SC2=000, SC1=000, MC=CMD0001)
VM1:GS04: %1GGM-000.132036 % JMS0154 'OMSW' LOGGED ON FOR 'PC4315 /DST1'.
JOB NAME 'MINERAL'. CALLER '(NONE)'. TID 004C0071
VM1:GS04: %1GGM-000.132041 % EXC0420 /LOGOFF PROCESSED. CPU TIME USED:
0.0409
SEC, USER ID: OMSW, TASK ID: 004C0071 , SERVICE UNITS: 000000000000918
```



### Connection cleardown

If the connection is cleared down by OMNIS (*CLSCON* command), the UCON partner is informed of the loss of the connection. This corresponds to normal dialog termination both for \$CONSOLE and for \$VMCONS.

The UCON partner can also clear down the connection with the *BCDISCON* command. OMNIS reacts to this by issuing the message OMS0006 (connection lost). CODE indicates why the connection was lost.

If a terminal to which *REPLY = RESTRICTED* applies loses or clears down the connection to a UCON partner, the terminal can no longer reply to any queries from the partner which are still open. Even if the connection to the same partner is restored later, earlier queries can no longer be answered.

#### *Exception*

The setting of *HOLD* for the terminal is *{YES / AUTO}*. In this case open queries can be answered again once the terminal has logged back on.

This also applies if the connection is taken over by another terminal with the aid of an *OCCUPY* command. This other terminal is now also able to reply to open queries directed to the initial terminal.

### 3.2.3 DCAM partners

The partner is a DCAM or CMX application.

#### Connection setup

The type *DCAM* and the name of the DCAM/CMX application must be specified in order to set up a connection to a DCAM/CMX application.

```
OPNCON          pac
                ,TYP=DCAM
                ,PTNNAME=dcam-application-name

                [,PROT[OCOL] = { O[MNIS]
                                D[SSIM] } ]

                [,CODE= { SYSCODE
                           BINARY } ]

                [,CPASS= { password
                           ? } ]

                [,MSG= { connection-message
                         ? } ]

DECLARE-PARTNER pac
                ,TYP=DCAM
                ,PTNNAME=dcam-application-name

                [,PROT[OCOL] = { O[MNIS]
                                D[SSIM] } ]

                [,CODE= { SYSCODE
                           BINARY } ]

                [,CPASS=password]
                [,MSG=connection-message]
```

The other operands depend on the particular application, and must be agreed upon with the person responsible for the application.

With *PROTOCOL=OMNIS*, OMNIS can interoperate only with DCAM/CMX applications that accept a program's connection request. Applications that can only set up connections actively, or accept connection requests only from terminals, can work with OMNIS only if *PROTOCOL=DSSIM* is set.

*PROTOCOL* is used to set the behavior of OMNIS with respect to a DCAM application. If *OMNIS* is specified, OMNIS behaves like an application; if *DSSIM* is specified, OMNIS behaves like a real terminal.

*PROTOCOL=DSSIM* results in the following differences compared with other types of partners:

- No PAC is displayed on the screen.
- The DSSIM partner is set automatically if an output from the DSSIM partner is output on the terminal.
- Switching away from partners with *PROTOCOL=DSSIM* is possible only with the K keys.
- The *TURN* command and the CALL key are not supported.
- Acknowledgments are handled by the partner (e.g. \$DIALOG).
- Start sequences are not supported.
- If asynchronous messages can arrive at the terminal from a different partner, the operand *EXCLUSIVE-PARTNER=YES* must be set for the DSSIM partner with the *SET* command or with *DECLARE-TERMINAL*.

For DCAM partners which work only in a strict dialog with formatted messages or which do not accept connection requests from programs, *PROTOCOL=DSSIM* should be used.

*CODE* is used to specify the code (EBCDIC or binary) in which data is to be sent to OMNIS.

Here again, *CPASS=?* and *CMSG=?* indicate that, for reasons of data protection, *CPASS* and *CMSG* should be requested by OMNIS in blanking mode.

### Connection cleardown

If the connection is cleared down by OMNIS (*CLSCON* command), the DCAM partner is informed of the loss of the connection. The reaction to this is program-specific.

If the connection is cleared down from the DCAM partner, OMNIS reacts with the message OMS0006 (connection lost).

### 3.2.4 UTM partners

**Case 1: The partner is a UTM application, without multiplex connection.**

#### Connection setup

The type UTM and the name of the UTM application must be specified in order to set up a connection to a UTM application:

```
OPNCON pac, TYP=UTM,PTNNAME=utm-application-name
        [,APPNAME=pterm-name] [,CPASS=password/?]
DECLARE-PARTNER pac, TYP=UTM,PTNNAME=utm-application-name
        [,APPNAME=pterm-name] [,CPASS=password]
```

The user must have been defined with *pterm-name* in the UTM application named *utm-application-name* (see “Administration and Programming” manual [1]).

The connection password is specified with *CPASS* if one was defined when the UTM application was activated. If *CPASS=?* is specified, the password is requested by OMNIS in blanking mode.

#### Dialog with the UTM partner

During a dialog with a UTM partner, the input of several (short) messages immediately after each other may result in violation of the strict dialog, since OMNIS accepts the SEND input from the terminal before the response to the preceding K1 input is sent to the terminal.

A type 9763 terminal may, in some circumstances, be unable to set up a connection to a UTM partner with *ROUTE=[IN]DIRECT*, since OMNIS does not recognize the status messages as such.

#### Connection cleardown

If the connection is cleared down by OMNIS (*CLSCON* command), the UTM partner is informed of the loss of the connection. The partner then reacts to this by resetting the current transaction.

If the connection is cleared down from the UTM partner (*KDCOFF* command), OMNIS reacts by issuing the message *OMS0006* (connection lost). If *KDCOFF BUT* is specified, the user is requested to sign on again.

## Case 2: The partner is a UTM application, with multiplex connection.

On multiplex connections, it is possible for users to prove their identity to UTM applications using Kerberos-authentication (previously chip card).

Each multiplex connection must be declared with *DECLARE-MUX*. It is not possible to enter multiplex connections dynamically, but it is possible to modify a configuration during operation by using the *CONFUPDATE* command (see [page 192](#)).

Using MUX connections, OMNIS supports applications (or formats) which work with the extended character set.

K keys always cause a REPEAT. In other cases (with multiplex connections), message traffic is governed by the right to send.

UTM transfers asynchronous messages to OMNIS over a MUX connection without fetching with *KDCOUT*.

All asynchronous messages are fetched if the terminal is currently engaged in a dialog with this UTM application. If you are currently engaged in a dialog with another partner, OMNIS announces an asynchronous message in the 25th screen line. If you switch back to the UTM application, then OMNIS automatically fetches all outstanding asynchronous messages.

### Connection setup

If a multiplex connection is desired, this must be specified explicitly at connection setup with *ROUTE=MUX*:

```
OPNCON pac, TYP=UTM,PTNNAME=partner-name
          [, PRONAME=processor-name], ROUTE=MUX
DECLARE-PARTNER pac, TYP=UTM,PTNNAME=partner-name
                  [, PRONAME=processor-name], ROUTE=MUX
```

The partner (*PTNNAME*) must be specified explicitly and must be declared in the configuration file as a multiplex connection.

Indirect connections and multiplex connections are mutually exclusive.

A multiplex connection is set up in several stages:

1. OMNIS checks whether the multiplex connection is declared in the configuration. If so, the message *OMS0012* is output.
2. If the OMNIS multiplex connection is not active, it is set up.
3. After successful setup of the transport connection, a request to set up a session for this *OPNCON* command is sent to the UTM application.

4. OMNIS receives a positive or negative acknowledgment of session setup from the UTM application. If the acknowledgment is positive, the message OMS0014 is output at the terminal; if the acknowledgment is negative, an error message is output.

Messages cannot be exchanged between the terminal and the UTM application until session setup has been positively acknowledged

The following figure illustrates the individual steps involved in setting up a multiplex connection:

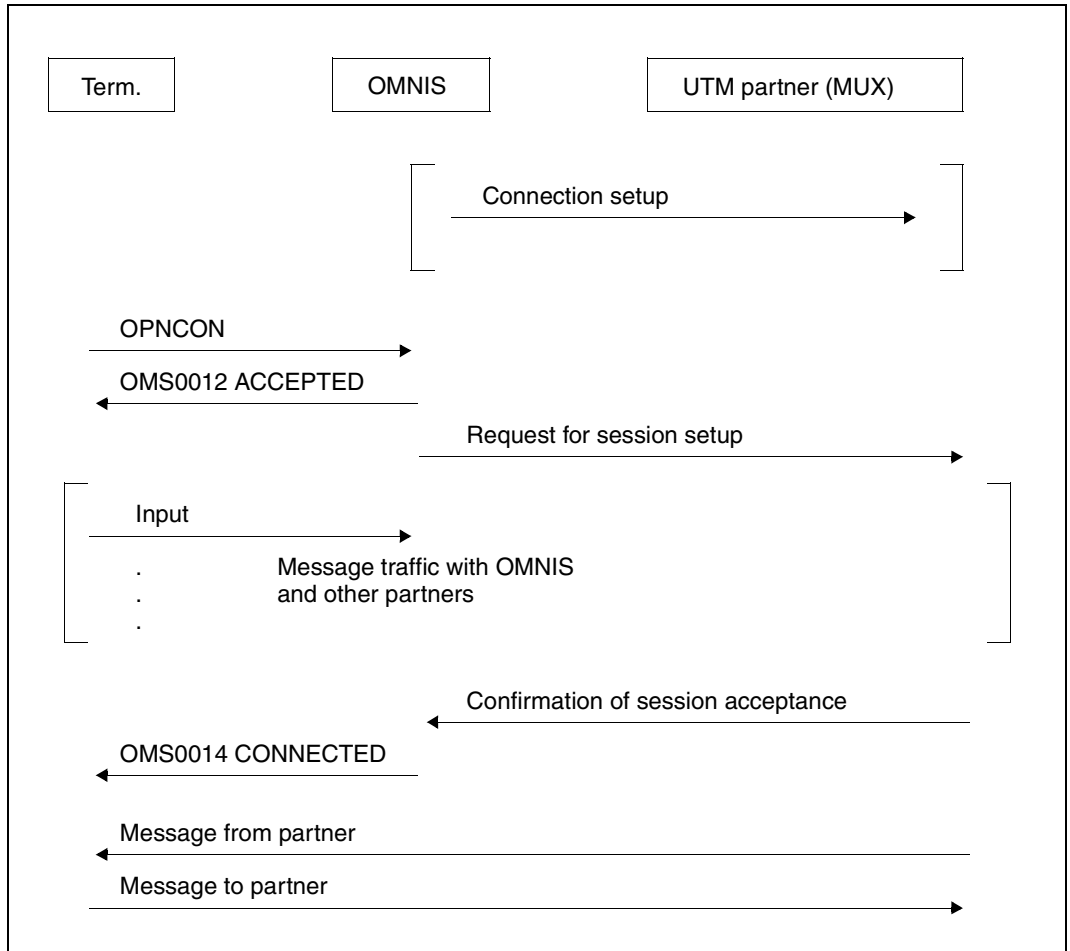


Figure 17: Setting up a multiplex connection

**Error situations**

OMNIS reacts as follows to errors which may occur in the interaction between OMNIS and UTM:

If UTM does not respond to a request to set up or clear down a session within 5 minutes, OMNIS releases the partner again. You can now initiate a new session setup with the *OPNCON* command.

The timeout period of 5 minutes is preset and cannot be changed.

The figure below illustrates what happens during session setup:

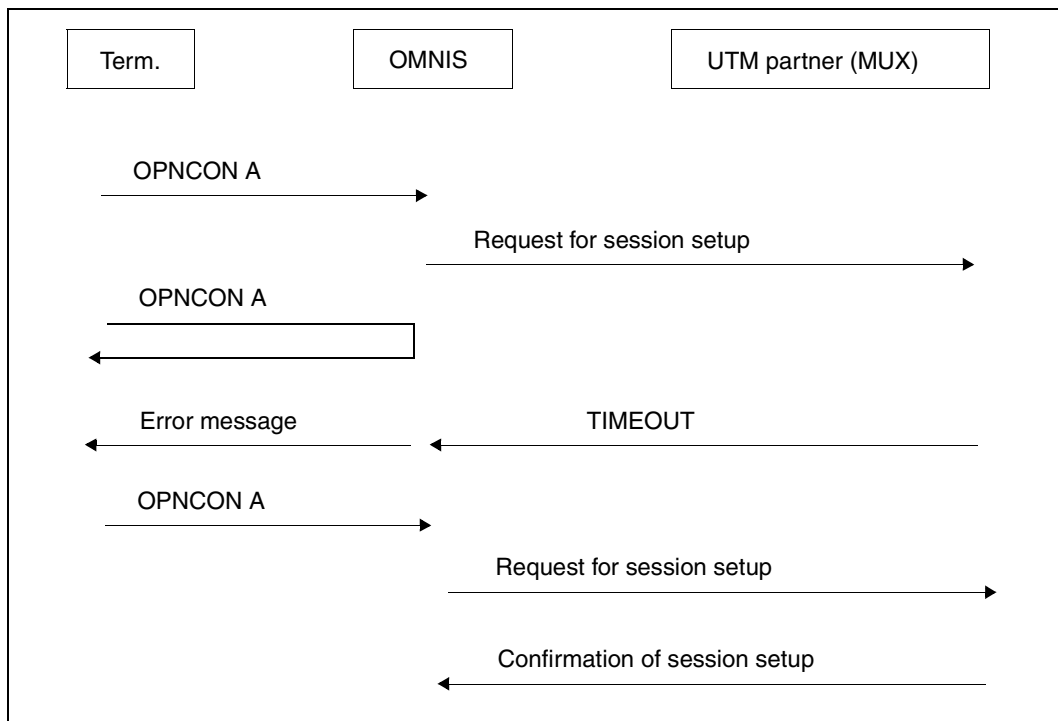


Figure 18: Session setup with *TIMEOUT*

### Connection cleardown

If the connection is cleared down by OMNIS (*CLSCON* command), the UTM application is informed of this by a separate message (a separate protocol element). The request for cleardown is confirmed. The UTM application reacts to this by resetting the current transaction.

The connection can be cleared down from UTM with the command *KDCOFF*. In the case of a multiplex connection, specifying the operand *BUT* has no effect: it is not possible to sign on again and OMNIS reacts to any such attempt by issuing the message `OMS0006` (connection lost).

The figure below illustrates the individual steps in the cleardown of a multiplex connection:

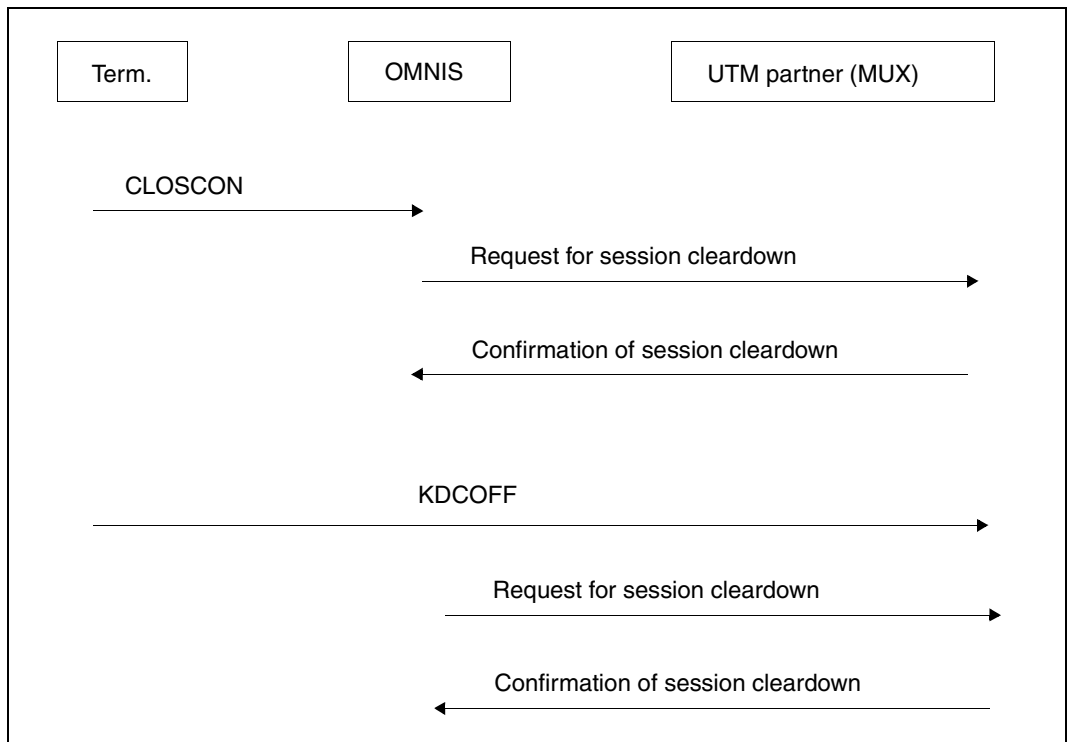


Figure 19: Cleardown of a multiplex connection



### 3.3 Start sequences

OMNIS permits *LOGON* or *KDCSIGN* commands to be sent to the partner immediately after a successful connection setup. The start sequences which contain the appropriate command strings are stored in the text file, which is an ISAM file. The text file is managed by the OMNIS administrator, who also assigns the sequence numbers. These can be specified with the operand *START-SEQUENCE* in the commands *OPNCON* and *DECLARE-PARTNER*:

```
OPNCON          pac, START-SEQUENCE=number (if SET OPNSTART=YES)
DECLARE-PARTNER pac, START-SEQUENCE=number
```

With TIAM partners, the operand *AUTO-LOGOFF* can also be specified in the *DECLARE-PARTNER* or *DECLARE-TERMINAL* statement to achieve an automatic *LOGOFF* after the start sequence has been processed.

```
DECLARE-PARTNER pac, START-SEQUENCE=number, AUTO-LOGOFF={YES/NO}
DECLARE-TERMINAL PRONAME=praname, PTNNAME=ptnname, AUTO-LOGOFF={YES/NO}
```

OMNIS then does the following:

While the start sequence is being processed, *N* is automatically given as the response to bulletin CONTINUEs occurring during screen output. The next record in the sequence is sent as input for each input request (*/* or *\**). If no more records exist in the sequence and *AUTO-LOGOFF=YES*, a *LOGOFF* is sent to the partner in response to the next system input request (*/*) received.

This allows TIAM partners to process procedures themselves and *ENTER* jobs to be started without the user having to send commands to TIAM.

The output of a format interrupts the processing of a start sequence. Processing of the start sequence is resumed, however, after the next input request (*/* or *\**) in line mode.

Similarly, OMNIS interrupts the processing of a start sequence if a blanked input from the terminal is requested. Passwords must not be contained in files. You can therefore demand, for example, that the password be entered blanked at the terminal.

If a start sequence for a UTM partner is to be processed, OMNIS reads the next record from the text file and sends it to UTM each time UTM relinquishes the right to send.

### 3.4 Loss of connection to terminals and partners

#### Loss of connections to terminals

A connection to a terminal is lost in the following situations:

- The user clears down the connection with the network command *CLSCON* (default is *::C*)
- The terminal is switched off
- The connection is explicitly cleared down by the operator or network administrator (*/BCDISCON*)

The actions taken by OMNIS after a loss of connection can be defined by the user via the *HOLD* operand.

The following commands define, for a partner, a terminal or globally, how OMNIS is to respond to a loss of connection to the terminal:

SET OPTION	}	HOLD=	$\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{AUTO} \\ \text{STD} \end{array} \right\}$	[ ,OPASS=	$\left. \begin{array}{l} \text{password} \\ ? \end{array} \right\}$	]
DECLARE-TERMINAL		HOLD=	$\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{AUTO} \\ \text{STD} \end{array} \right\}$	[ ,OPASS=password]		
PARAM OPNCON	}	pac, HOLD=	$\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{STD} \end{array} \right\}$	[ ,OPASS=]	$\left. \begin{array}{l} \text{password} \\ ? \end{array} \right\}$	]
DECLARE-PARTNER		pac, HOLD=	$\left. \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{STD} \end{array} \right\}$	[ ,OPASS=password]		

The scope of the above commands is described in [chapter “OMNIS commands” on page 167](#); the priority for *HOLD=AUTO* can be changed using the *PRIORITY* command.

#### *Loss of connection when HOLD=NO*

The connection to a partner, for which *HOLD=NO* has been defined, is cleared down. The effect is the same as for operation without OMNIS (when the terminal is connected directly to the partner).

*Loss of connection when HOLD=YES*

The connection to a partner, for which *HOLD=YES* has been defined, is not cleared down but remains in effect. When a connection is lost the terminal assumes HOLD status, i.e. connections to partners exist although no terminal is connected to OMNIS for these partners. Messages from this partner arriving in the meantime are written to the logging file.

If a terminal signs on to OMNIS again after the loss of connection, the partners are not automatically assigned to it again but have to be specifically made available again using the *OCCUPY* command (the HOLD status is canceled only by *OCCUPY*). The messages written to the logging file while the connection was down are not output automatically, but are now only available in page-turning mode for one partner at a time.

It is possible that a few messages may be lost in the process. These are messages sent by OMNIS before the connection was lost, but which could not be displayed at the terminal.

*HOLD=YES* is effective only for partners to which OMNIS can assign an *OPASS*.

*Loss of connection when HOLD=AUTO*

If *HOLD=AUTO* was specified, all partners are again available to the terminal as soon as the connection to OMNIS is reestablished. It is not necessary that an *OPASS* be defined or that the *OCCUPY* command be issued.

In the case of a **non-unique user** and *SECURITY LEVEL≠LOW*, the specification *HOLD=AUTO* is ignored; *HOLD=NO* always applies. The uniqueness of the user is checked by OMNIS-MENU.

In the case of a **unique user** and *SECURITY LEVEL≠LOW*, only the connections of Kerberos-authenticated partners (previously chip card) are cleared down.

The administrator can cancel the effect of a user-specified *HOLD=AUTO* operand with the *PRIORITY* command (see [page 274](#)).

*Taking over a connection from an indirect partner*

If indirect partners are not taken over automatically with *HOLD=AUTO* or with the *OCCUPY* command, it is necessary to check the OMNIS-to-OMNIS connection and, if necessary, re-establish it with the *RESTART* command. Indirect partners which still exist on both sides are activated and can be taken over with the *OCCUPY* command. Indirect partners which no longer exist on one side are eliminated. A connection can only be established to these indirect partners by logging off the terminal with *HOLD=NO* and then logging the terminal on again.

**Loss of connections to partners**

A connection to a partner is lost in the following situations:

- if the partner clears down the connection (e.g. after *LOGOFF*, *KDCOFF*);
- if a fault occurs in the network (failure of a communication computer or line; only in the case of direct inter-computer connections);
- when the connection is explicitly cleared down by the operator or network administrator;
- when the application has been terminated.

OMNIS is informed by the data communication system of the loss of connection. OMNIS then issues the message `OMS0006` (connection lost).

The partner remains preset in OMNIS, which means that any further entries still go to the already inactive partner, producing the message `OMS0007` (partner not found) or `OMS0082` (partner not active).

Otherwise, OMNIS behaves exactly as if the connection had been cleared down by the user with the *CLSCON* command.

### 3.5 Taking over connections from terminals and partners

A partner is always assigned to a terminal, i.e. there is always a terminal that he or she “belongs to”. This ownership can be altered using the *OCCUPY* command:

```
OCCUPY TID=tid,OPASS=password
OCCUPY PID=pid,OPASS=password
```

If *?* is entered instead of the password (*OCCUPY TID=tid, OPASS=?*) the *OCCUPY* password is requested in blanking mode with the message OMS0137. When OMNIS-MENU is used, *USER* can also be specified instead of *TID*.

The *OCCUPY* password is assigned to terminals and partners with one of the following commands:

```
OPTION          }
PARAM   pac,    } OPASS={ password }
HALT HOLD=YES, }      {      ?      }
```

```
DECLARE-TERMINAL }
DECLARE-PARTNER  } OPASS=password
pac,              }
```

The OMNIS administrator can also take over partners without knowing the *OCCUPY* password. For this, he or she must explicitly specify the administration password:

```
OCCUPY { TID=tid } , APASS={ password }
      { PID=pid }      {      ?      }
```

If *PID* is specified, only the partner addressed by *pid* is taken over. If *TID* is specified, all partners of the terminal addressed with *tid* are taken over.

The old “owner” may be an active terminal or a terminal in the HOLD state. In the latter case, the connection to the terminal has been aborted or explicitly cleared down with the command *HALT HOLD=YES*. If the terminal from which a partner has been taken over is active, a suitable message is displayed on this terminal.

A user who is identified by name can access all connections under his or her name. If multiple connections meet the search criteria you have entered, OMNIS rejects the *OCCUPY* command.

In the case of a **non-unique user** and *SECURITY LEVEL≠LOW*, the *OCCUPY* command is rejected. Any specification of *OPASS* is ignored without comment; *HOLD=NO* always applies.

*Example*

The combinations resulting from the settings

```
OPTION   OPASS='X'
PARAM A, OPASS='X'
PARAM B, OPASS='Y'
PARAM C, OPASS='Z'
```

have the following results:

```
OCCUPY TID=tid,   OPASS='X'   A, B and C are taken over
OCCUPY TID=tid,   OPASS='Y'   A, B and C are not taken over
OCCUPY TID=tid,   OPASS='Z'   A, B and C are not taken over
OCCUPY PID=A,     OPASS='X'   A is taken over, B and C not
OCCUPY PID=B,     OPASS='Y'   B is taken over, A and C not
OCCUPY PID=C,     OPASS='Z'   C is taken over, A and B not
```

**Taking over a connection after connection cleardown**

When a connection to a terminal is cleared down with the *HALT* command, the automatic cleardown of existing connections to partners can be prevented by specifying *HOLD=YES/AUTO*:

- If the connection to the terminal is cleared down with *HALT HOLD=YES*, all partners for which *HOLD=YES/STD* is specified are placed in the HOLD state and can be taken over with *OCCUPY* after the terminal signs on to OMNIS again or by another terminal.
- If the connection to the terminal is cleared down with *HALT HOLD=AUTO*, all partners, regardless of their *HOLD* settings, are placed in the HOLD state and are automatically taken over (without *OCCUPY*) when the terminal signs on to OMNIS again.

In the case of a **non-unique user** and *SECURITY LEVEL≠LOW*, the specification *HOLD* is ignored; *HOLD=NO* always applies. The uniqueness of the user is checked by OMNIS-MENU.

In the case of a **unique user** and *SECURITY LEVEL≠LOW*, only the connections of Kerberos-authenticated partners (previously chip card) are cleared down.

**Taking over a connection from an indirect partner**

If indirect partners are not taken over automatically with *HOLD=AUTO* or with the *OCCUPY* command, it is necessary to check the OMNIS-to-OMNIS connection and, if necessary, re-establish it with the *RESTART* command. Indirect partners which still exist on both sides are activated and can be taken over with the *OCCUPY* command. Indirect partners which no longer exist on one side are eliminated. A connection can only be established to these indirect partners by logging off the terminal with *HOLD=NO* and then logging the terminal on again.

### Taking over connections after the start of OMNIS

If a terminal is connected after OMNIS has been activated, the following applies for the takeover of partners declared with *CONNECT=START*:

- `DECLARE-TERMINAL, HOLD=AUTO`  
`DECLARE-PARTNER pac, CONNECT=START`

If the terminal is declared with *HOLD=AUTO*, all (active) partners are automatically taken over (and remain active). *OCCUPY* does not need to be entered. If indirect partners are to be taken over, the terminal must be declared with *HOLD=AUTO* on the remote OMNIS (SLAVE) or *HOLD=AUTO* must be set OMNIS-wide.

- `DECLARE-TERMINAL, HOLD={YES/NO}`  
`DECLARE-PARTNER pac, CONNECT=START, OPASS=password`

The terminal is in *HOLD* state and cannot be taken over, as *OPASS* was not defined in the declaration. All partners for which the *OPASS* was specified can be taken over using `OCCUPY PID=pid, OPASS=password`.

- `DECLARE-TERMINAL, OPASS=password, HOLD={YES/NO}`  
`DECLARE-PARTNER pac, CONNECT=START`

The terminal and all its (active) partners can be taken over using `OCCUPY TID=tid, OPASS=password`.

## 3.6 OMNIS-OMNIS connections

### Setting up the connection

OMNIS-OMNIS connections must be declared. Only one OMNIS-OMNIS connection is possible within one computer or between two computers; see also [“OMNIS address code” on page 90](#). There are no OMNIS-OMNIS connections between three or more computers.

OMNIS-OMNIS connections can be set up explicitly or implicitly.

### Explicit connection setup

An OMNIS-OMNIS connection can be established explicitly using the *RESTART* command. This command is not privileged.

### Implicit connection setup

An OMNIS-OMNIS connection is set up implicitly when OMNIS is activated if the connection is declared with *CONNECT=START*.

If an indirect connection is to be established to a partner in a computer to which no OMNIS-OMNIS connection yet exists, this is also set up implicitly.

Normally users need not concern themselves with setting up the OMNIS-OMNIS connection.

### Connection cleardown

An OMNIS-OMNIS connection can be cleared down from OMNIS using the *CANCEL* command.

### Loss of connection

A connection is lost in the following situations:

- if a fault occurs in the network (failure of a communication computer or line);
- when the connection is explicitly cleared down by the operator or network administrator;
- if OMNIS has been terminated.

OMNIS is informed by the data communication system of the loss of connection. OMNIS then informs all those affected (i.e. all users to whom an indirect connection existed via the interrupted OMNIS-OMNIS connection), and sets the OMNIS-OMNIS connection to “inactive”.



The indirect connections remain in their current status until the OMNIS-OMNIS connection has been reestablished.

Messages and command inputs relating to an inactive OMNIS-OMNIS connection are rejected (exceptions: *CLSCON* and *RESTART*).

### Restart

An OMNIS-OMNIS connection can be restarted explicitly or implicitly.

#### Explicit restart

An inactive OMNIS-OMNIS connection can be reestablished by using the *RESTART* command. OMNIS sets up all indirect connections again, including the partner to which *CONNECT=START* applies.

Once the connection has been successfully reestablished, each OMNIS involved checks which indirect connections still exist and which no longer exist and informs the other OMNIS in the connection of this (on the one hand, the user may in the meantime have cleared down indirect connections using *CLSCON* or *HALT*; on the other hand, the connection to the partner may have been lost).

Indirect connections which still exist on both sides are activated and can be used again. Indirect connections which no longer exist on one side are eliminated.

Each user is informed by a message as to which connections are available and which have been eliminated.

#### Implicit restart

If an OMNIS-OMNIS connection has been declared with *CONNECT=START* on both sides, the OMNIS-OMNIS connection is restarted automatically in the following instances:

- when the other OMNIS is activated
- in response to an *OPNCON* command from an indirect partner via this OMNIS-OMNIS connection

The effect is the same as with a *RESTART* command entered at the earliest time possible.

### 3.7 Connections to printers

All messages displayed at the terminal can be logged for a particular terminal or a particular partner on a printer. The printer is declared as a hardcopy unit or a separate station.

Hardcopy output is controlled by means of the commands *SET*, *OPTION* and *DECLARE-TERMINAL* or *PARAM*, *OPNCON* and *DECLARE-PARTNER*:

$$\begin{array}{l}
 \text{PARAM} \\
 \text{OPNCON} \\
 \text{DECLARE-PARTNER}
 \end{array}
 \begin{array}{l}
 \text{pac,} \\
 \text{pac,} \\
 \text{pac,}
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{PARAM} \\ \text{OPNCON} \\ \text{DECLARE-PARTNER} \end{array}} \right\}
 \text{HCOPY} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{STD} \\ \text{hac} \end{array} \right\}$$
  

$$\begin{array}{l}
 \text{SET} \\
 \text{OPTION} \\
 \text{DECLARE-TERMINAL}
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{SET} \\ \text{OPTION} \\ \text{DECLARE-TERMINAL} \end{array}} \right\}
 \text{HCOPY} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{STD} \\ \text{hac} \\ (\text{D[DEVICE]}=\text{devname}, \text{K[EY]}=\text{num} \\ , \text{F[FORM]}=\text{STD} | \text{form-name}) \end{array} \right\}$$

In the case of the commands *SET*, *OPTION* and *DECLARE-TERMINAL*, you can use *DEVICE=devname, K[EY]=number* to define a K key with which you can initiate a screen dump and simultaneously initiate printing on the specified printer *devname* with the *form-name* specified in *FORM*.

This hardcopy key and also the request for the screen contents are not handled via EXIT routines.

Printing takes place via the BS2000 macro *PRNT*.

The differences in printer output depend on whether *HCOPY=YES* or *HCOPY=hac* is specified and on the basic generation (see the "Administration and Programming" manual [1]). These differences are summarized in the following table.

<b>HCOPY=YES</b>	<b>HCOPY=hac</b>
<p>Depending on the printer connection, a distinction is made between</p> <ul style="list-style-type: none"> <li>- a local hardcopy unit, and</li> <li>- a central hardcopy unit.</li> </ul> <p>Only outputs in line mode are logged. In the case of formats and physically edited messages</p> <ul style="list-style-type: none"> <li>- HCOPY=Y has no effect</li> <li>- output takes place on the hardcopy unit and is program-controlled (by the partner).</li> </ul> <p>Terminal output is coupled with printer output in the case of a local hardcopy unit. In the case of a central hardcopy unit, terminal output is largely independent of printer output.</p> <p>Blocking of outputs is not possible.</p> <p>If a printer fails, the message "Printer error" is displayed in the status line each time an output to terminal occurs.</p>	<p>The printer is defined as a separate station with DECLARE-HARDCOPY or with the HCOPY command.</p> <p>Input, output and formats are logged.</p> <p>The output of messages to printer is completely independent of messages traffic with the terminal. OMNIS assumes responsibility for administration. Output can then be accelerated by blocking outputs.</p> <p>If a printer fails, automatic switchover to a predefined alternate device takes place, if such a device has been defined.</p> <p>If a printer is no longer needed, the connection is automatically cleared down, i.e. the printer is no longer assigned to a terminal or partner.</p>

### 3.8 OMNIS and HIPLEX AF

OMNIS (-MENU) supports automatic switchover of applications between BS2000 systems as offered by HIPLEX AF, see the “HIPLEX AF” manual [7] (Highly Integrated System Complex Availability Facility).

With HIPLEX, “switching to a system” means: “Restarting on an intact system”.

The description in this section applies to both OMNIS and OMNIS-MENU.

OMNIS uses the virtual host name offered by BCAM and, from HIPLEX AF, the switchover of UTM applications between virtual hosts. OMNIS itself can run on a virtual host. The valid host name is that specified in */SET-DCAM-APPLICATION-LINK*.

OMNIS supports the switchover function only for multiplex connections to UTM.

With OMNIS, a configuration file is always created with the OMNISKD utility. The *DECLARE MUX* statement associated with OMNISKD has now been extended for interworking with HIPLEX AF by the addition of the *AVAILABILITY* operand.

The *AVAILABILITY* operand can be assigned the value *NO* or *YES*.

*AVAILABILITY = NO*

This setting corresponds to the OMNIS behavior to date.

*AVAILABILITY = YES*

With this setting OMNIS saves the multiplex transport connection to the relevant UTM application, together with all the sessions running on it.

This has the following result in the event of a system failure:

All the sessions which are affected and the MUX connection go on hold (*HOLD* status). OMNIS first notifies the user of the failure by means of message *OMS0224* (connection no longer available).

HIPLEX AF initiates rapid “switchover” of the UTM application to another computer system with precisely the same application environment. As soon as the UTM application contacts OMNIS again, OMNIS ensures that the sessions are signed on again with the same settings as before. Once this has been achieved, OMNIS issues message *OMS0226* (connection now available).

The end user can then work with the UTM application as though nothing had happened. (However, UTM resets a transaction to its status at the last synchronization point.)

If in the meantime the user has attempted to send a message to the application concerned, OMNIS responds by issuing message *OMS0225* (connection in hold).

In this phase no message traffic is possible for the sessions concerned. Any new *OPNCON* command is also rejected. The *CANCEL* or *CLSCON* command is only effective if the *KILL* operand is specified. However, existing sessions can be taken over with *OCCUPY*.

### **Saved MUX partners at a connection takeover**

When the connection to a terminal is taken over automatically, saved MUX partners in *HOLD* status are not logged in again immediately, but only when the MUX connection is available again. If the corresponding terminal is in *HOLD* status, a saved MUX partner is not logged in immediately when the MUX connection is re-established, but only when the terminal is active again.

The saved sessions in *HOLD* status can, however, also be taken over with *OCCUPY*. These are logged in again immediately the MUX connection is active.

## 3.9 Load distribution

Load distribution on the basis of a variable concept is now possible in addition to load distribution using HIPLEX MSCF.

### 3.9.1 Load distribution with HIPLEX MSCF

In OMNIS, it is also possible to distribute the required computational load to multiple BS2000 systems. This "load balancing" feature minimizes the response time for the user, while maximizing the throughput of the overall system.

The load distribution to multiple hosts by OMNIS is offered for TIAM and DCAM partner types. It is not possible for indirect partners.

The equivalence of the systems with respect to an application must be ensured by the system administrator.

#### Using load balancing

Load balancing by OMNIS is defined at the time of setting up a connection to the partner. This can be done by specifying *PRONAME=\*ANY* in the *OPNCON* command (or in the *DECLARE-PARTNER* and *DECLARE-APPLICATION* statements).

```
OPNCON <pac>, PRONAME=*ANY
DECLARE-APPLICATION . . . , PRONAME=*ANY
DECLARE-PARTNER . . . , PRONAME=*ANY
```

The precise command syntax can be found under the description of *OPNCON* on [page 240](#) or under *DECLARE-PARTNER* and *DECLARE-APPLICATION* in the "Administration and Programming" manual [1].

*\*ANY* causes OMNIS to determine the two processors with the lowest load in the default job class (by a JMS system call).

OMNIS then sets up a connection to the processor with the lowest load.

If the connection setup fails, OMNIS tries to set up a connection with the second host processor and sends a corresponding message to the console. If both attempts are rejected, the user also receives an error message.

If no HIPLEX group exists, OMNIS will likewise issue an error message.

### 3.9.2 Load distribution on the basis of a variable concept

As well as the option already described of distributing load by specifying the host name *\*ANY*, i.e. selecting the "best" computer in a HIPLEX-MSCF cluster, there is the alternative option of distributing load by specifying an S variable or job variable as the host name.

```
DECLARE-APPLICATION ...,PRONAME=&variable  
DECLARE-PARTNER ...,PRONAME=&variable  
OPNCON ...,PRONAME=&variable
```

You can find detailed information on specifying and substituting variables in the "Administration and Programming" manual [1].

The current values for the S variables and job variables are substituted when OMNIS is executed. The scope of the variables is as follows, depending on the variables used:

- The OMNIS start procedure
- The OMNIS task
- The ID OMNIS was started under





---

## 4 Dialog with OMNIS

Dialog with and via OMNIS is described in detail in the following.

### 4.1 Address codes and identifiers

The address codes and identifiers described below are used to uniquely identify partners, terminals, hardcopy units, etc. during dialogs with or via OMNIS.

#### Partner address code

Each partner is assigned a partner address code (PAC). The partner address code can be freely selected by the user, consists of one through four alphanumeric characters and identifies the partner uniquely for a particular terminal. The partner address code can be assigned statically when the partner is declared or dynamically when the connection is set up:

```
OPNCON          pac
DECLARE-PARTNER pac
```

#### Group address code

A number of partners can be combined into a group and addressed jointly with a common group address code (GAC). The group address code can be freely selected by the user, consists of one through four alphanumeric characters and identifies the group uniquely for a particular terminal.

A group address code is defined using one of the following commands which at the same time define the members of the group:

```
GROUP          gac [ ,ADD=(pac,...) ]
DECLARE-GROUP  gac [ ,ADD=(pac,...) ]
```

Partner and group address codes must be unique for a particular terminal.

The individual groups may be defined before the connections to the group members are set up.

The same formal rules apply to both the group address code and the partner address code; a group address code may be specified with *PARAM*, *OPNCON* and *CLSCON*.

A message entered with a group address code is sent to all active partners belonging to the group. A group address code in a command causes the command to be executed once for each partner of the partner group. Exceptions to this are the *OPNCON* and *CLSCON* commands: *OPNCON* is executed only for the partners which are inactive, *CLSCON* only for those which are active.

In addition to the partner groups specified with the *GROUP* and *DECLARE-GROUP* commands, there are some standard partner groups which are automatically generated and managed by OMNIS:

- \* is a partner group containing all partners currently assigned to the terminal.
- \*C is a partner group containing all UCON partners.
- \*D is a partner group containing all DCAM partners.
- \*T is a partner group containing all TIAM partners.
- \*U is a partner group containing all UTM partners (including those with *ROUTE=MUX*).

The group address codes \*, \*C, \*D, \*T and \*U can be used just like all other group address codes.

### OMNIS address code

The OMNIS address code (OAC) identifies OMNIS. It can be freely selected by the administrator, consists of the symbol @ followed by up to three alphanumeric characters and identifies OMNIS uniquely (i.e. all OMNIS address codes must be unique “OMNIS-wide”). The OMNIS address code is assigned to the OMNIS statically when an OMNIS-OMNIS connection is declared:

```
DECLARE-OMNIS oac
```

The OMNIS address code for the local OMNIS is predefined and consists of the character @ alone.

Commands are entered after the corresponding OMNIS address code; see also [page 172](#). OMNIS issues messages identified by their OMNIS address code.

### Hardcopy address code

A terminal printer generated as a station must either be declared in the configuration file or be defined with the *HCOPI* command; in both these cases, it is assigned a hardcopy address code (HAC):

```
DECLARE-HARDCOPY hac
HCOPI             hac
```

The hardcopy address code consists of one through four alphanumeric characters and identifies the printer uniquely, i.e. the hardcopy address codes must be unique “OMNIS-wide”.

By means of this hardcopy address code, the printer can be assigned to a partner, a terminal or globally. It can be assigned using the *PARAM*, *OPTION*, *SET*, *DECLARE-PARTNER* and *DECLARE-TERMINAL* commands.

### Exit address code

An exit group must be either declared in the configuration file or defined with the *EXIT* command; in both these cases, it is assigned an exit address code (EAC):

```
DECLARE-EXIT eac
EXIT         eac
```

The exit address code consists of one through four alphanumeric characters and identifies the exit group uniquely, i.e. the exit address codes of all exit groups must be unique “OMNIS-wide”.

By means of this exit address code, the exit group can be assigned to a partner, a terminal or globally. It can be assigned using the *PARAM*, *OPTION*, *SET*, *DECLARE-PARTNER* and *DECLARE-TERMINAL* commands.

### Partner identifier (PID)

OMNIS assigns each partner a sequence number. This number, the partner identifier (PID), identifies the partner uniquely “OMNIS-wide”. The PID is passed to the terminal user in the message *OMS0012* and must be specified in certain commands (it is optional for others). The PID thus has a similar function to the TSN in BS2000.

### Terminal identifier (TID)

Each terminal is assigned a sequence number when it signs on to OMNIS. This number, the terminal identifier (TID), identifies the terminal uniquely “OMNIS-wide”. The TID is communicated to the terminal user in the subsequent message *OMS0001* and must be specified in certain commands (it is optional for others). The TID thus has a similar function to the TSN in BS2000.

**Hardcopy identifier (HID)**

OMNIS assigns each hardcopy unit a sequence number (HID) which identifies the hardcopy unit uniquely “OMNIS-wide”. For declared hardcopy units, the HID is assigned when OMNIS is activated; for undeclared hardcopy units, it is assigned during operation when the HCOPY command is executed. The HID must be specified in certain commands (it is optional for others). The HID thus has similar function to the TSN in BS2000.

**OMNIS identifier (OID)**

Each remote OMNIS to which a connection is declared from the local OMNIS is assigned a sequence number, the OMNIS identifier (OID), when the local OMNIS is started. This number identifies the remote OMNIS uniquely. The OID must be specified in certain commands (it is optional for others) and thus has a similar function to the TSN in BS2000.

**Multiplex identifier (MID)**

OMNIS assigns each multiplex connection a sequence number (MID) which identifies the multiplex connection uniquely “OMNIS-wide”. The MID is assigned when OMNIS is started. It must be specified in certain commands (it is optional for others) and thus has a similar function to the TSN in BS2000.

## 4.2 Connection of terminals to OMNIS

After the terminal  $\longleftrightarrow$  computer connection has been established (dialing up of computer; SEND, if necessary), BCAM responds with PLEASE ENTER NET COMMAND.

If OMNIS is running, the connection to OMNIS can be set up by entering:

```
0 omnis-name [,processor-name] [,PW=password] [,...] or
0 omnis-name [,pp/rrr] [,PW=password]] [,...]
```

Entry of a password is required only if one has been defined by the administrator (see the “Administration and Programming” manual [1]).

If the terminal is connected successfully to OMNIS, the following messages are issued:

```
CONNECTED WITH omnis,processor-name;IND=C'::'
```

```
@: OMS0001 USER IS CONNECTED TO OMNIS (TERMINAL=
      station-name/processor-name,TID=tid).
```

```
[ @: line 1 of the bulletin file
  @: line 2 of the bulletin file
  @: line 3 of the bulletin file
  .
  .
  .
  .
  @: line n of the bulletin file ]
```

```
@: OMS0002 OMNIS VERSION <ver> READY
```

```
@:/
```

Connection of the terminal to OMNIS is thus completed and the user can now start a dialog.

Connection of a terminal to OMNIS is not possible if

- the maximum number of terminals from which OMNIS may be used at the same time has been reached. In this case, the message OMS0057 is then issued.
- the terminal is not declared and the administrator has specified that only declared terminals may be connected. The message OMS0063 is then issued.
- exit routines which prevent the connection of the terminal are being used. Exit routines can also suppress or redirect the output of messages.

The terminals must either support network dialogs or must be assigned permanently to the partner *omnis-name*, where *omnis-name* is the name for OMNIS (default “OMNIS”) defined in the ENTER file (see the “Administration and Programming” manual [1]).

If a terminal cannot be connected to OMNIS, you should contact the administrator.

### 4.3 Operation in line mode

A message is sent to a partner by prefixing it with the partner address code assigned to the partner:

```
pac:input-message
```

OMNIS removes *pac*: from the message and transmits only *input-message* to the partner addressed by *pac*.

The partner address code is always separated from the actual message by the character `:` (colon).

With the following commands the user can decide on the message output style: style 1 (*PAC=PREFIX*), style 2 (*PAC=LINE*), or style 3 (*PAC=NO*).

```
OPNCON          }
OPTION          } PAC= { PREFIX }
PARAM    pac,  }      { LINE   }
                   }      { NO    }
```

The administrator can use the following commands for this purpose:

```
SET              }
DECLARE-TERMINAL } PAC= { PREFIX }
DECLARE-PARTNER  }      { LINE   }
pac,             }      { NO    }
```

In the output direction, OMNIS places the partner address code in front of messages arriving from the partners so that each sender of a message can be uniquely identified:

Style 1: `pac:output-message`

Style 2: `pac: partner-name/processor-name (partner-type) output-message`

Style 3: `output-message`

A message is always output at the terminal from which the connection was set up, unless the terminal (or partner) was taken over from another terminal with the *OCCUPY* command. In the example below a user is communicating simultaneously via a terminal with three (TIAM) partners which have been assigned the partner address codes A, B and C:

```
A: /A: START-PROGRAM FROM-FILE=PROG1
A:% BLS0500 PROGRAM 'PROG1' ... LOADED.
B: SHOW-FILE-ATTRIBUTES INFORMATION=SPACE-SUMMARY
C: COPY-FILE FROM-FILE=UVW, TO-FILE=XYZ, PROTECTION=SAME
B: TOTAL PUBLIC PAGES ALLOCATED .....
B: /
A: PROGRAM PROG1 READY
A: *
C: /B: /START-PROGRAM FROM-FILE=PROG2
B:% BLS0500 PROGRAM 'PROG2' ... LOADED.
```

In a typical application the terminal is connected to two partners. Communication with one of the partners is extensive, but only a small number of messages are sent to the other partner or only system messages are received from the other partner.

In such cases communication is facilitated further in that the partner address code can sometimes be omitted, i.e. once specified, a partner address code remains valid until a new partner address code is entered. If a message is entered without a partner address code, it is forwarded to the same partner to which the previous input message was delivered:

```

B: /A: START-LMS
A: % BLS0500 PROGRAM 'LMS' ... LOADED.
CTL=(CMD) ...
$PAR LOG=MED
A: $LIB FILE=DAT,USAGE=BOTH,STATE=NEW
A: LIBRARY IS CLEARED AND PREPARED
$B: START-PERCON
B: % BLS0500 PROGRAM 'PERCON' ... LOADED.
B: *A: END
A: LMS-TERM-MSG : NONE
**** E N D   O F   R U N   ****   LMS (BS2000) VERSION <ver>
A: /

```

Care is advised with input data which is preceded by a character string which could be interpreted by OMNIS as a partner address code, e.g. the EDT entry @:# for reassigning the command character (@ is a predefined OMNIS address code). In such cases it is imperative that the partner address code of the partner is specified: A:@:# (where A is the partner address code of the partner).

When a dialog is conducted with a partner via OMNIS, the outputs on the screen may in some cases differ slightly from direct outputs (without OMNIS). Screen overflow control, if set, has no effect.

OMNIS can also output partner-specific information (PAC, PIC, partner/processor name, partner type, route, K key, user ID) in the 25th line of the screen (see *LINE25* in *OPTION* and *SET*). This may, however, result in conflicts with the partner if the partner itself writes messages to or deletes them from the system status line.

## 4.4 Operation in formatted mode

Communication with partners using partner address codes only functions smoothly with line-oriented messages (for example, operating system messages and commands).

If, however, a format is output, OMNIS cannot display the sender's partner address code on the screen without destroying the format. The user cannot prefix the input message with a partner address code. There is a danger that the format will be destroyed by a message arriving from another partner.

For this reason OMNIS offers the following aids in formatted mode:

- As the default option, OMNIS does not indicate the sender when formats are output, i.e. the partner address code is not displayed. It is left to the user to ascertain the identity of the sender from the contents of the format.  
Alternatively, OMNIS can output partner-specific information (PAC, PID, partner/processor name, partner type, route, K key, user ID) in the 25th line of the screen (see *LINE25* under *OPTION* and *SET*). This may, however, result in conflicts with the partner if the partner also writes messages to, or deletes messages from, the system status line.
- A response to a format output is automatically sent to the sender of the format with no opportunity of specifying the partner address code and irrespective of the partner to which the previous message was addressed. This means that the current partner setting is changed to the sender of the format.

*MUX partners are an exception to this rule:*

If an asynchronous message was sent from a MUX partner, the partner is not set automatically in this way. The partner setting remains unchanged and the response to the format output is sent to the partner to whom the previous message was sent.

- OMNIS prevents a format from being destroyed by other messages by blocking the terminal for outputs from other partners after a format has been output until a response (completed format or actuation of a K key assigned to a partner or OMNIS with *KPAC*) has arrived from the terminal. Messages arriving in the interim from other partners are buffered by OMNIS and output in sequence at the terminal after the format response is entered.



**Example**

A terminal communicates with four partners, who are working partly in formatted mode and partly in line mode.

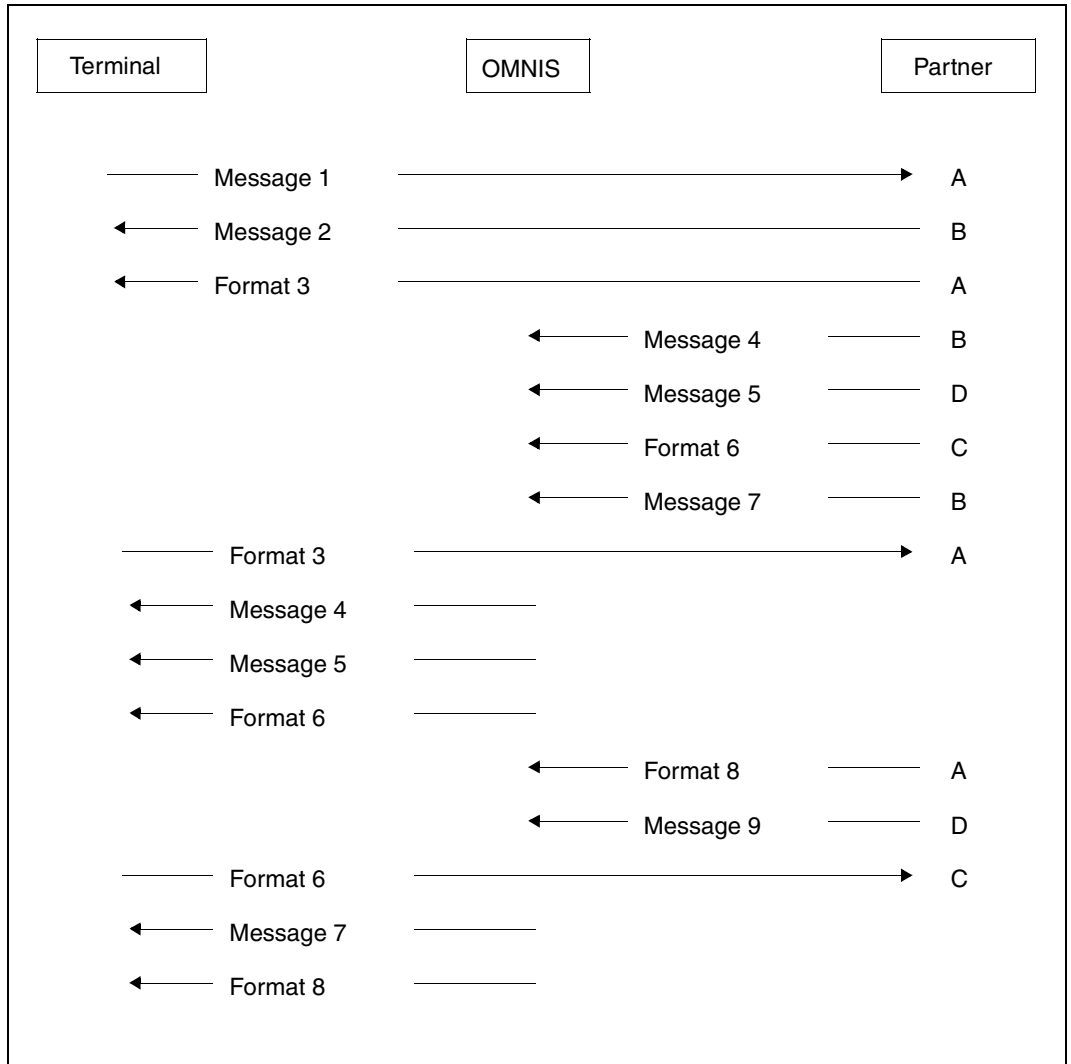


Figure 20: Communication between a terminal and partners in formatted and line mode

## 4.5 Switching between multiple partners in formatted mode

If you wish to communicate with multiple partners from one terminal, you can switch from one application to another using K keys.

For this to be possible, a K key must be allocated to each partner to which a switch is to be made. This can be achieved by declaration or dynamically using the following commands:

```

DECLARE-PARTNER pac, KPAC=number,
OPNCON          pac, KPAC=number,
PARAM           pac, KPAC=number
  } TYP={
  } TYP={
  } TYP={
        TIAM
        DCAM
        UTM
  }

```

OMNIS checks all inputs from the terminal to see whether one of the defined K keys has been pressed. If this is the case, a message is sent to the partner selected by that key.

- The UTM command *KDCDISP* or, in the case of a multiplex connection (*ROUTE=MUX*), the protocol element *REPEAT* (which has the same effect as *KDCLAST*) is sent to the UTM partner. The UTM partner then repeats its last output, which OMNIS passes on to the terminal.
- Other types of partners (non-UTM partners, i.e. DCAM or TIAM partners) are sent the same message which is sent when the terminal is switched off and on again and the K3 key pressed (change message).  
If the partner is working with FHS, a return code to the effect that the screen has been destroyed is issued by FHS, and it is up to the program to respond to this return code in an appropriate manner.  
Other partners (e.g. EDT and SDF) interpret K3 as a request to restore the screen contents.

The following must be noted when using the facility to switch partners by means of K keys:

- Each K key must uniquely identify an OMNIS partner. It is therefore not possible to assign one K key to more than one partner.
- A K key used for addressing an OMNIS partner must not already be assigned a different function (switchover to OMNIS command mode, *CALL*, *BREAK*).
- Inputs made with K keys already assigned a function in OMNIS are not passed on to partners. It is therefore recommended that you use K keys which are not interpreted by partners. K keys K4 ... K13 are available as ESC sequences which may be loaded into P keys if required. Avoid key K14 because the message it generates is produced by the ID card reader when an ID card is removed.

- UTM partners with format exit must be set up in such a way that they are able to process screen restarts correctly (the format exit is called with *RESTART* in this case).
- Other OMNIS partners with formatted mode must be set up in such a way that they are able to process the switch message and respond accordingly.
- If a K key is to be assigned to an indirect UTM partner with the *PARAM* command, the following should be noted:

The *PARAM* command must not be sent to the local OMNIS, but rather to the remote OMNIS, specifying the OMNIS address code:

```
oac: PARAM pac,kpac=n
```

The specified K key must not have a different function assigned to it, either in the local OMNIS or in the remote OMNIS.

**Example**

A user opens two connections, to the DCAM application DCA1 and to the UTM application UTM2, and assigns to them the keys K11 and K12 respectively:

```
OPNCON U1,TYP=DCAM,PTNNAME=DCA1,KPAC=11,APPNAME=DCAMTERM1
OPNCON U2,TYP=UTM,PTNNAME=UTM2,KPAC=12,APPNAME=UTMTERM2
```

The following dialog may ensue:

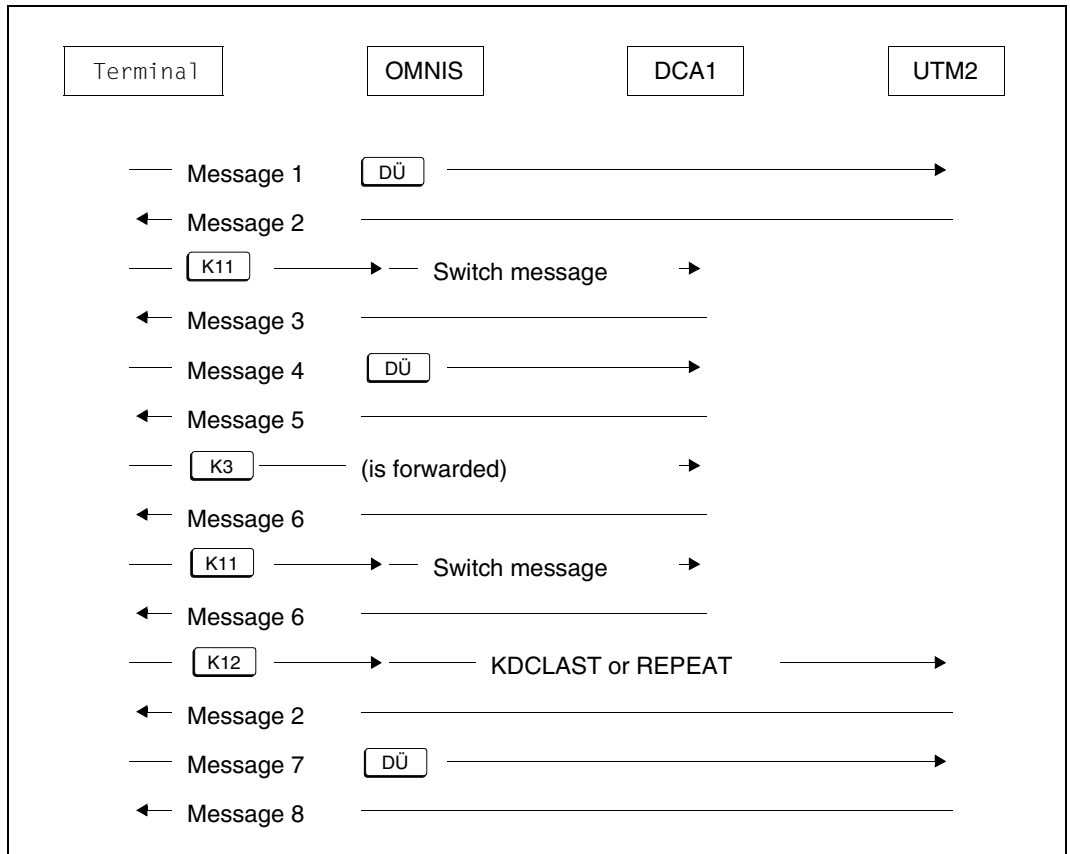


Figure 21: Switching between partners in formatted mode with K keys

## 4.6 Switching from formats to OMNIS command mode

As long as a format is on the screen, OMNIS will not interpret any partner address code input. It is therefore not possible for the user to switch to another partner by entering a partner address code. OMNIS insists on a strict dialog.

However, it is possible to switch from formatted mode to the OMNIS (line-oriented) command mode. The format on the screen is destroyed in the process.

The switch to OMNIS command mode is made by a freely selectable K key. This key is defined using one of the commands

```
DECLARE-TERMINAL }
OPTION            } KPAC=number
SET               }
```

The following must be noted when using the facility to switch from formats to OMNIS command mode:

- The K key used for changing to OMNIS command mode must not already be assigned a different function (OMNIS partner addressing, CALL key, BREAK key).
- Inputs made with K keys already assigned a function in OMNIS are not passed on to partners. It is therefore recommended that you use the K keys which are not interpreted by partners. K keys K4 ... K13 are available as ESC sequences which may be loaded into P keys if required. Avoid key K14 because the message it generates is produced by the ID card reader when an ID card is removed and the partner may later clear down the connection if he or she checks for an inserted ID card.
- Except in the case of UTM partners, there is no guarantee that the format will be reconstructed on the screen after switching from a format to OMNIS command mode; this depends to a large extent on how the partner application is programmed.

With UTM applications, it is possible to switch back from OMNIS command mode to the UTM format.

## Example

A terminal communicates with two partners, who are working partly in formatted mode and partly in line mode (key K12 is reserved for switching to OMNIS command).

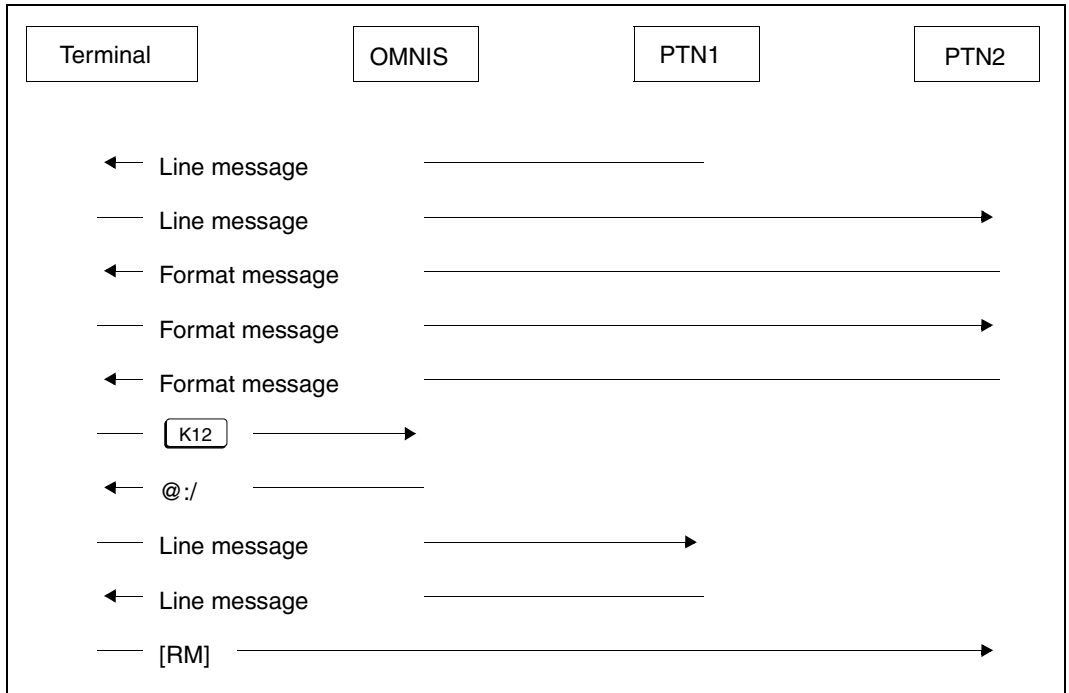


Figure 22: Switching from formatted mode to OMNIS command mode with a K key

The message to restore the PTNR2 screen (represented by [RM] in the figure) depends on the product, and cannot be specified for a general case. The OMNIS extension OMNIS-MENU permits the restart message to be defined specifically for each partner.

If a K key assigned to the user with *KPAC* is pressed, "RM" corresponds to a short message which signals loss of connection to the terminal.

- The UTM command *KDCDISP* or, in the case of a multiplex connection (*ROUTE=MUX*), the protocol element *REPEAT* is sent to a UTM partner. The UTM partner then repeats its last output, and OMNIS passes this on to the terminal.
- Other types of partners (non-UTM partners, i.e. DCAM or TIAM partners) are sent the same message which is sent when the terminal is switched off and on again and the K3 key pressed (change message).

If the partner is working with FHS, a return code to the effect that the screen has been destroyed is issued by FHS, and it is up to the program to respond to this return code in an appropriate manner.

Other partners (e.g. EDT and SDF) interpret K3 as a request to restore the screen contents.

### Switching partners via the OMNIS command mode

It is possible to switch between partners in formatted mode indirectly via OMNIS command mode using the K keys. It is not, however, possible to switch from command mode to the appropriate partner application by entering *pac*:

For TIAM partners, *pac*: in line mode acts like *BREAK* (see [page 110](#)); for DCAM and UCON partners, entering *pac*: results in the message OMS0122 (message of length 0 ignored).

The following figure illustrates various possibilities for switching between partners, where:

OPT KPAC=4

PARAM PTN1 ,KPAC=5

PARAM PTN2 ,KPAC=6

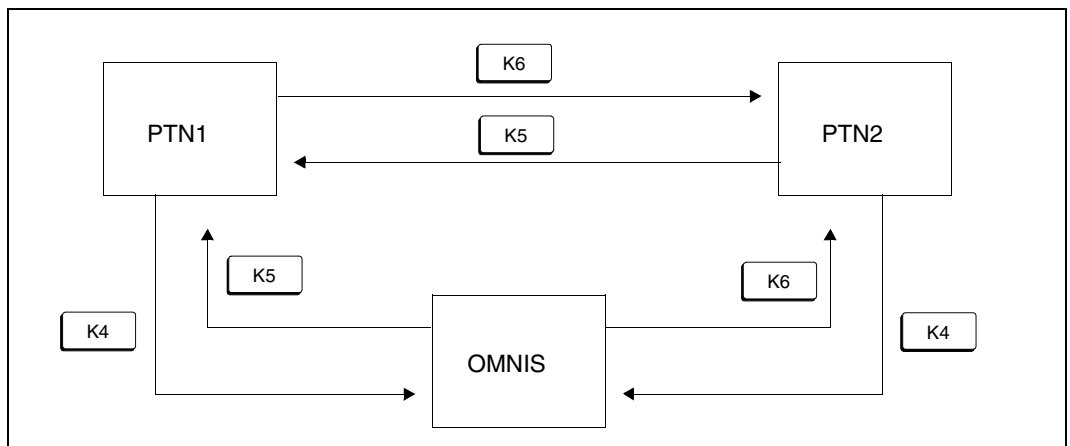


Figure 23: Switching between partners directly and via OMNIS command mode

## 4.7 Special points relating to the use of K keys in OMNIS

As OMNIS interprets K keys internally, the following points must be observed when sending K keys to partners:

- If no CALL KEY has been defined, no K keys are forwarded to the partner in line mode; instead, they are treated as a call (i.e. each K key initiates a call).
- If a CALL KEY has been defined (*OPTION* or *SET* command), all other K keys not assigned other functions by commands are forwarded to the partner.
- K keys assigned by means of the *SET*, *OPTION*, *PARAM*, etc. commands are not forwarded to the partner.
- In formatted mode, all K keys that have not been assigned are forwarded.



## 4.8 Output control

If an OMNIS terminal has opened connections to a number of partners simultaneously, OMNIS outputs the messages from these partners to the terminal spontaneously in the order in which they arrive.

Under normal circumstances this is a desirable feature, but it can lead to problems when there is a large amount of output from the partners. If the outputs occur in rapid succession, the terminal user may not have the opportunity to

- enter his or her own messages for a partner or send commands to OMNIS
- read all the messages before they disappear from the screen in roll-up mode
- read a message from a partner if the output of the message is followed immediately by another output which clears the screen (e.g. change from line mode to formatted mode)

If there are a large number of connections existing at the same time, it may also be difficult to place messages in their proper context if they are being output by a number of different partners.

To solve these problems, OMNIS offers three functions that are entirely independent of each other:

- message flow control
- screen overflow control
- interrupt facility (CALL)

Further assistance is provided in the form of

- output to a hardcopy unit
- the logging file, in which the user can page

## 4.8.1 Message flow control

OMNIS message flow control provides you with the option of checking the (output side) message flow from each partner to the terminal.

OMNIS offers you two variants for this:

- control with the *EXCLUSIVE-PARTNER* operand
- control with the *CLASS* operand.

Since the two variants affect one another, you cannot use both at the same time.

With *EXCLUSIVE-PARTNER=YES*, all partners behave as in the case of *CLASS=OUTPUT*, regardless of their *CLASS* setting.

### Variant 1: Control with the *EXCLUSIVE-PARTNER* operand

With the commands *SET* and *OPTION*, or *DECLARE-TERMINAL* or *DECLARE-USER* (in the case of OMNIS-MENU), you can control the message flow by setting *EXCLUSIVE-PARTNER=YES*.

While you are engaged in a dialog with one partner, all messages from all other partners will then be buffered. You must switch to a different partner by entering the PAC or using a K key before OMNIS will output that partner's messages - and then it outputs only that partner's messages.

Each message requiring inputs is announced through message OMS0215 in screen line 25. In the case of pure output messages, only the first message for each partner waiting in the background is announced (by means of message OMS0214).

With groups, only the messages for the partner standing in first position in the group are output, all messages from all other partners are buffered.

#### *Effects on a partner with format output*

On switching to this partner, the previously announced message is output, but is only visible for a brief period, as it is immediately overwritten by the screen refresh carried out as a result of the restart message sent to the partner.

If screen saving is set for this partner (*SAVE=SCREEN*), no restart message is sent, and the announced message remains visible.

## Variante 2: Control with the CLASS operand

The message streams from the partners are divided into three classes:

- OUTPUT      The messages from partners assigned to the *OUTPUT* class are passed on to the terminal without delay.
- SAVE        Messages from partners in the *SAVE* class are saved by OMNIS and not output immediately to the terminal.
- DELETE     Messages from partners in the *DELETE* class are neither output to the terminal nor buffered, which means that they are lost.

Each partner is assigned to just one of these classes at any given time. The class assignment can be changed at any time using the *PARAM* or *OPNCON* commands with the *CLASS* operand.

For example, the following commands remove the partners with partner address codes *pac1*, *pac2* and *pac3* from their previous classes and assign *pac1* to the *OUTPUT* class, *pac2* to the *SAVE* class and *pac3* to the *DELETE* class:

```
PARAM pac1,CLASS=OUTPUT  
PARAM pac2,CLASS=SAVE  
PARAM pac3,CLASS=DELETE
```

After the connection to a partner has been set up with *OPNCON* without the *CLASS* operand or a partner has been taken over with *OCCUPY*, the class assignment is retained.

If a partner is transferred from the *SAVE* class to the class *OUTPUT*, the buffered messages are output to the terminal in partner-specific paging mode.

## 4.8.2 Screen overflow control

Screen overflow control causes the spontaneous output of messages to the terminal to be halted if

- a screen becomes full in roll-up mode
- a message which would clear the screen is to be output
- the CALL key is pressed (see [page 109](#))
- you enter an empty message without a CALL key having been defined

The output of messages is continued once an entry is made by the terminal user. This entry can be

Readiness for input is indicated by OMNIS with the short message @:ACK.

This input may be:

- a command to OMNIS
- a message intended for a partner
- a K key (except the call key)
- an empty message (SEND)



If you switch partners at @:ACK, then all the *SAVE* functions have no effect for this switch at the acknowledgment point.

ACK and the subsequent acknowledgment are not recorded in the logging file.

The screen overflow control is controlled with the ACK operand of one of the commands SET, OPTION or DECLARE-TERMINAL:

$$\left. \begin{array}{l} \text{SET} \\ \text{OPTION} \\ \text{DECLARE-TERMINAL} \end{array} \right\} \text{ACK} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{STD} \\ \text{LINE} \end{array} \right\}$$

Note that an OMNIS command which answers an acknowledgment may also redefine the message flow control. If, as a result of a programming error, a partner begins to send uncontrolled messages, output of the messages can be deactivated with *PARAM pac*, *CLASS=DELETE*.

### 4.8.3 Halting the output

Unsolicited message output at the terminal means that the user sometimes has no opportunity to enter messages without the input being disrupted by a further output.

Where output activity is very brisk, the screen overflow control generates “respites” in which any amount of time is available to enter longer messages.

Where output activity is less brisk, it is possible to halt output to the terminal briefly by pressing the CALL key (interrupt), which has to be defined beforehand; the CALL key thus has a similar function to the CALL key on a real console.

The CALL key is defined with the operand *CALL-KEY*.

```
SET
OPTION
DECLARE-TERMINAL } CALL-KEY=number
```

If no CALL key has been defined, any empty message (i.e. any K key not already defined elsewhere or EM-SEND) will produce a *CALL*.

A *CALL* is possible, however, only if there is sufficient time to press the K key (escape sequence) between two outputs. As of type 9750 terminals this can always be achieved by programming the K key into a P key and pressing the P key for the call.

In all cases, however, the *CALL* becomes effective only when the BCAM message pool has been emptied. This pool contains messages which have already been sent to the terminals by OMNIS but which have not yet been output at the terminals by the teleprocessing network. After the *CALL* input it is thus possible that a few messages will still be output.

If the screen overflow control is active and the output has stopped at an ACK point, the CALL key or any empty message is not interpreted as *CALL* but merely as an acknowledgment which has no other effect.

The K key programmed as the CALL key may not have any other function assigned to it (partner addressing, BREAK key or switch to OMNIS command mode).

Inputs made with K keys already assigned a function in OMNIS are no longer passed on to partners. It is therefore recommended to use K keys which are not interpreted by partners. K keys K4 ... K13 are available as ESC sequences which may be loaded into P keys if required. Avoid using key K14 because the message it generates is produced by the ID card reader when an ID card is removed.

Whether or not *CALL* is active is not normally indicated by a specific message, but only by the absence of message outputs. If desired, the terminal user can have the active *CALL* indicated by the short message @:CALL (*pac*) where *pac* is the address code of the partner to which, by default, the next input is sent.

If a format is output, the K key assignment defined by *CALL-KEY* is passed on to the partner.

The *CALL* indicator is controlled with the *CALL-INF* operand:

SET	}	CALL-INF=	{	YES
OPTION				NO
DECLARE-TERMINAL				STD

#### 4.8.4 Interrupting the dialog

If a terminal is being used to work together with TIAM without OMNIS, the short message key K2 and all inputs violating the strict dialog are interpreted as *BREAK*, i.e.:

- a program is interrupted (switch from user mode to system mode)
- a procedure is interrupted: DO YOU WANT TO INSERT COMMANDS (Y/N)?

During a dialog via OMNIS, a *BREAK* is achieved by

- pressing the *BREAK* key
- an “empty” message, i.e. a message consisting only of a partner address code (line mode only)
- the short message key K2 (formatted mode only)
- all inputs violating strict dialog with the TIAM partner

The *BREAK* key is defined with the *BREAK-KEY* operand:

SET	}	BREAK-KEY=	{	number
OPTION				
DECLARE-TERMINAL				

The K key programmed as the *BREAK* key may not have any other function assigned to it (UTM partner addressing, *CALL* key, or switching to OMNIS command mode).

Inputs made with K keys already assigned a function in OMNIS are no longer passed on to partners. It is therefore recommended to use K keys which are not interpreted by partners. K keys K4 ... K13 are available as ESC sequences which may be loaded into P keys if required. Avoid using key K14 because the message it generates is produced by the ID card reader when an ID card is removed.

## 4.9 Automatic system operating through message tables

The main task of an operator is to respond to specific events by performing specific actions (possibly the same actions repeatedly). The events can be divided into two categories:

- events which require “mechanical intervention” by the operator
- events which only require a specific console input

The occurrence of an event (in either category) is almost always indicated by an output (a message) at the console; OMNIS can receive such console messages via the UCON interface.

With the functions for automatic operating incorporated into OMNIS, the actions required in response to events in the second category can to a large extent be automated, so that fully automatic or semi-automatic operating is possible.

Fully automatic operating means that all operator activities can be automated to such an extent that an operator is no longer required. Semi-automatic operating is when only certain standard situations can be automated, but operator intervention is required for special situations.

For fully automatic or semi-automatic operating, OMNIS provides the functions of automatic answering of messages, message suppression and OMNIS-PROP (or other exit routines). All three mechanisms can be implemented in parallel; it should be noted, however, that they do affect each other.

The software product OMNIS-PROP is described in the “OMNIS-PROP” manual [2]).

### Message answering and message suppression

The basic function of fully automatic or semi-automatic operating is that messages are answered automatically with predefined responses. Responses in this case are not just “true” responses (format: `tsn.text`) but also console commands which are intended to take account of a particular situation.

Message suppression allows the output of particular messages to be suppressed. It is thus possible to eliminate unimportant messages.

Message suppression and message answering are independent of each other, i.e. messages are answered even if their output at the terminal is suppressed. Conversely, where message output is allowed, the (automatically generated) reply is also output at the terminal.

It is also possible to trigger an audible alarm at the terminal for specific messages.

## Declaration of message tables

Each message to which there is to be a response and/or whose output is to be suppressed must be made known to OMNIS, and the standard response(s) and/or a “suppressed” flag specified at the same time (message declaration).

A number of message declarations are combined to form a message table. A number of message tables can be administered at the same time, but only one of these will be current for one partner at any one time.

This makes for very flexible operation:

- each system can be administered via a separate message table; it is, however, also possible to have one message table for all systems
- the current message table can be switched over at any point, thus allowing automatic operating to be introduced at suitable times or separate tables to be used for fully automatic and semi-automatic operating.

## Creating message tables

A message table is created using the command `MDEF table, ACTION=CREATE`, see [page 229](#). To make a message table available to all OMNIS users, you must assign it to the partners or to the virtual consoles (see [page 116](#) for details).

## Generating/altering message table entries

Message table entries are generated or altered using the `MDEF table, ACTION=CREATE` command, see [page 229](#).

Each entry in a message table describes a message and the appropriate responses to it. A message is uniquely identified by

- the BS2000 message number (*MSG* operand), and
- in some cases an additional string. This string is defined either by the *ID* operand or by the *INSERT* operand. The *INSERT* operand is not supported for virtual consoles.

The *MSG* operand can be specified more precisely by means of *position*. In this case, the specified message is ignored unless it is located at the position defined by *position*.

The response to a message can be a direct reply to the message originator (format: *.text*) and/or a sequence of any operator commands (format: */command*) (*REPLY* operand).

Neither *text* nor *command* may contain the ampersand character `&`, which is reserved for OMNIS.



*Examples*

1. If message DMS0715 begins precisely at position 35, the reply *.YES* is to be used.  
MDEF TAB1,MSG=(DMS0715,35),REPLY='.YES'
2. The reply to a *MOUNT* message must always be *.NO* (unattended night-time operation):  
MDEF TAB1,MSG=NKVT013,REPLY='.NO'
3. All program load messages are to be suppressed:  
MDEF TAB1,MSG=BLS0519,DISPLAY=NO  
  
The *DISPLAY* operand controls the message suppression function. A message for which *DISPLAY=NO* is specified is not displayed on the terminal.
4. The *LOGON* message from OMNIS is to be output on the terminal with an audible alarm.  
MDEF TAB1,MSG=JMS0154,INSERT=('OMNIS',0),BELL=YES  
  
The *BELL* operand controls the audible alarm at the terminal. A message specified with *DISPLAY=YES* and *BELL=YES* is accompanied by an audible alarm when output at the terminal.
5. All *MOUNT* messages are explicitly to attract the operator's attention by means of an audible alarm:  
MDEF TAB1,MSG=NKVT013,DISPLAY=YES,BELL=YES

The message answering and message suppression functions can be used in any combination.

A message table entry can be altered at any time by entering a further *MDEF* command with the same message number and the same *INSERT* or the same *ID* (if any).

**Character substitution &TSN, &VMP**

It is necessary in many instances to take action to influence a particular process, e.g. by means of */CANCEL-JOB tsn* or */SEND-MSG tsn*.

To this end, it is necessary to insert the relevant current TSN into the reply defined by the *REPLY* operand of the *MDEF* command.

This is achieved by specifying the dummy parameter *&TSN* at the points at which the current TSN is to be located. OMNIS replaces each *&TSN* that occurs with the current TSN of the process which generated the message.

Also the dummy parameter *&VMP* can be used to specify a virtual console.

*Example*

As before, the reply to *MOUNT* messages must always be *.NO*. In addition, however, the message originator is to be informed that there is no operator on hand:

```
MDEF TAB1,MSG=NKVT013,REPLY=(' .NO' , '/MES &TSN, ' '*** NO OPERATOR *** ' ' ' )
```

## Character substitution &MSGADDR

If the message originator specifies a message job ID equal to “000” (column 7-9 in the message output format), an answer will automatically be sent to his/her TSN. If the message originator specifies a message job ID not equal to “000”, this is no longer possible. The originator is then identified via the dummy parameter &MSGADDR. &MSGADDR has the content of column 2-9 of the console message, which is “sender-message job ID” (see the “Introduction to System Administration” [6]). &MSGADDR can also be used as a dummy parameter in the *REPLY* operand.

### Examples

```
MDEF MSTD, ACTION=CREATE
MDEF MSTD, MSG=MSG0001, REPLY=' .TEXT1 '
MDEF MSTD, MSG=MSG0002, REPLY=' &TSN. TEXT2 '
MDEF MSTD, MSG=MSG0003, REPLY=' &MSGADDR. TEXT3 '
MDEF MSTD, MSG=MSG0004, REPLY=' &VMP: &MSGADDR. TEXT4 '

```

## Sequence of MDEF commands

A message table is a list in the following form:

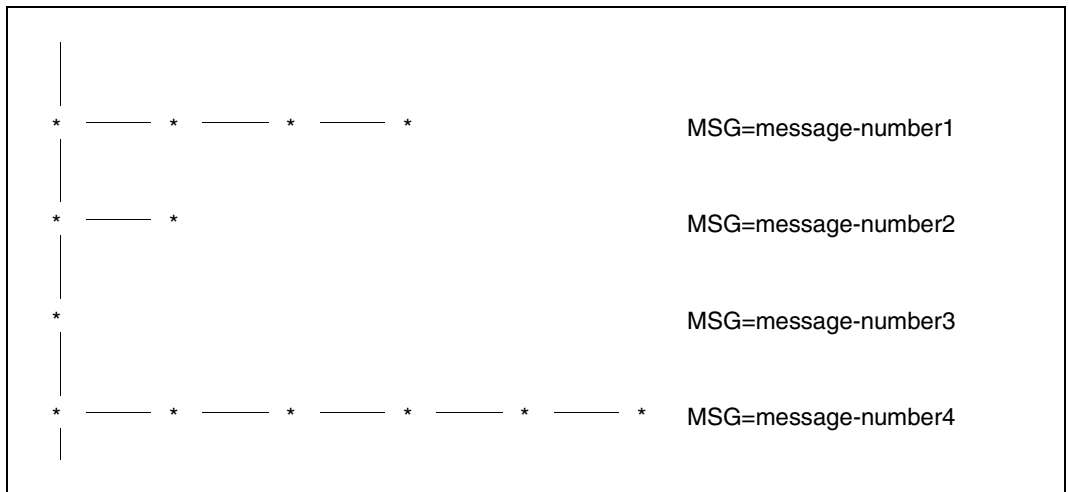


Figure 24: Structure of a message table

All entries in a horizontal branch have the same message number and are distinguished by the *ID* operand; all entries in the vertical branch have different message numbers.

The message table is constructed from top to bottom and from left to right; this is the search sequence used to ascertain whether a message is defined in the table or not. Thus, when a message which is not defined is searched for, only the vertical branch is searched; the message variants in the horizontal branches have no effect on runtime.

The search algorithm used dictates that messages should be defined in order of frequency. The frequency of the messages depends on the operations executed on a system; on a software development system, for example, the program load message EXC0560 will be one of the most frequent messages.

Different definition sequences for the individual variants can be used to achieve different effects. The following statement sequence causes all *LOGON* messages except those for *TSOS* and *SERVICE* to be suppressed:

```
MDEF 1, ACTION=CREATE
MDEF 1, MSG=JMS0154, ID='TSOS', DISPLAY=YES
MDEF 1, MSG=JMS0154, BELL=YES, ID='SERVICE', DISPLAY=YES
MDEF 1, MSG=JMS0154, DISPLAY=NO
```

The following statement sequence is meaningless since the second definition can never be reached because the first definition will always be interpreted as a valid hit:

```
MDEF 1, ACTION=CREATE
MDEF 1, MSG=JMS0154, DISPLAY=NO
MDEF 1, MSG=JMS0154, ID='SERVICE'
```

The following statement sequence has the effect of defining printer L1 as the standard printer and printer L2 as the laser printer and providing an automatic reply to the corresponding form requests. Other form requests are output at the terminal and require a reply from the operator:

```
MDEF 1, ACTION=CREATE
MDEF 1, MSG=SPS0200, ID='STD FORM L1', REPLY='.Y', DISPLAY=NO
MDEF 1, MSG=SPS0200, ID='LASER FORM L2', REPLY='.Y', DISPLAY=NO
```

### Assigning message tables to partners

The message tables are normally fixed in their definition and are constructed when OMNIS is activated (startup file). They can, however, also be generated or modified while OMNIS is running.

The assignment of message tables is carried out with one of the commands *SET*, *OPTION*, *PARAM*, *OPCON*, *DECLARE-TERMINAL* or *DECLARE-PARTNER* using the *MTAB=table* operand.

Message tables can be assigned only to partners of type UCON.

The assignment can be modified or canceled at any time.

### Assigning message tables to virtual consoles

Message tables are assigned to virtual consoles in the same way as for any other partner of the type UCON (see above). The assignment can also be modified or canceled at any time (even during operation).

The special feature in the assignment of message tables to virtual consoles relates to the prefix of the virtual console. Connection setup to \$VMCONS via the partner type UCON is worthwhile only for one or more VC dialogs (operating guest systems via virtual console). The subsequent dialog should therefore begin with the command `/BEGIN-VM-DIALOG ..., TYPE=*VC`. In this case, the UCON-specific functions for message tables and color control for the virtual console are available via the UCON partner type. For the reasons mentioned above, the prefix of a virtual console must never be deactivated if several VC dialogs are conducted via a connection with the UCON partner type. The dialog with the virtual console must therefore be started with the command `/BEGIN-VM-DIALOG ...,TYPE=*VC(OUTPUT-PREFIX=*YES)`.

To make sure that there is a clear virtual console / guest system assignment, you are recommended to only ever conduct one VC dialog over a connection with the the UCON partner type. You can then also work with `OUTPUT-PREFIX=*NO`.

If the prefix is activated, the following statement can be used to ensure that the message MSG001 with the string `IDI` sends the command `MES` to virtual console GS05:

```
MDEF TAB,MSG=MSG0001,D=N,ID=' ID1 ',REPLY=' GS05:/MES &TSN.HELLO'
```

The following statement can be used to ensure that the message `SELF` always sends the command `STA` to the virtual console from which the message came:

```
MDEF TAB,MSG=SELF,D=N,REPLY='&VMP:/STA &TSN'
```

### Deleting message tables

A message table can be deleted using the command `MDEF table, ACTION=DELETE`.

## 4.10 Screen and P key saving

It is not possible in the case of all partners, when switching back, for you to return to the current dialog. OMNIS therefore offers you the capability to save the screen.

You control the function “Save screen” and also the function “Save P keys” by means of the *SAVE* operands in the commands *DECLARE-PARTNER*, *DECLARE-TERMINAL*, *OPNCON*, *OPTION*, *PARAM* and *SET*.

In the case of “Save screen”, before switching partners OMNIS reads in the current screen from the terminal. As soon as you switch back to the old partner, OMNIS outputs the saved screen again on the terminal.

In this way you are able to continue working without problems following a partner switch.



If you switch partners at *@:ACK*, then all the *SAVE* functions have no effect for this switch at the acknowledgment point.

OMNIS always saves the entire screen. It is advisable, therefore, to employ message flow control at the same time as otherwise messages from other partners which happen to be currently on the screen will also be saved on switching partners.

If you make entries in *LINE* mode after outputting the saved screen, then OMNIS clears the screen on the next output.



### Caution!

For each partner for which you activate the “Save screen” function, OMNIS requires storage space the length *4096*. If this length is exceeded (due to a large number of control elements), the screen is not saved. Shorter screen contents are subsequently saved again correctly.

Just as it is possible to save the screen, you can also save the P keys for each partner. You control the function “Save P keys” by means of the *SAVE* operands in the commands *DECLARE-PARTNER*, *DECLARE-TERMINAL*, *OPNCON*, *OPTION*, *PARAM* and *SET*.

With data display terminals of type 9763 the P keys are read in by OMNIS in the same way as the screen. It is thus possible to save and read in again manually changed P keys.

With all other data display terminals types, the P keys cannot be read in; each message must therefore be checked for P keys. You should remember therefore that the function “Save P keys” can have a negative effect on the performance of OMNIS. Because the P keys cannot be read in, with these terminal types OMNIS is also unable to recognize manually changed P keys.

## 4.11 Color control

OMNIS supports only color terminals of the types 9752, 9754 and 9763. You can insert color control characters in messages to be output.

Data display terminals 9752 and 9754 support four colors: red, green, yellow and white. Data display terminal 9763 supports seven colors: red, green, yellow, white, blue, cyan and magenta.

### Suggested applications

- Each partner has its own color. OMNIS then outputs a partner's message in “its” color. The same color can also be assigned to more than one partner.
- In the case of UCON partners, OMNIS supports the system conventions by outputting information messages in yellow and requests in red.
- The user can also mix the two above-mentioned applications. In this case, the UCON prefix is depicted in the color of the partner and the rest of the message in the color defined by the system conventions.

### Inserting color control characters

You define the colors of partner messages with the *COLOR* operand of the *PARAM* or *DECLARE-PARTNER* command. You define the color of the OMNIS messages with the *COLOR* operand of the *OPTION* or *DECLARE-TERMINAL* command:

OPTION PARAM pac,	}	COLOR =	{	NO RED GREEN WHITE YELLOW BLUE CYAN MAGENTA	}
DECLARE-TERMINAL DECLARE-PARTNER pac,	}				

*COLOR=NO* means that OMNIS inserts no color control characters in messages.

Inputs are always displayed in green.

The output of UCON messages and the control of the colors for virtual console output is defined with the *DISPLAY-MODE* operand:

$$\left. \begin{array}{l} \text{OPTION} \\ \text{SET} \\ \text{DECLARE-TERMINAL} \end{array} \right\} \text{ DISPLAY-MODE} = \left\{ \begin{array}{l} \text{STD} \\ \text{OMNIS} \\ \text{SYSTEM} \\ \text{MIXED} \end{array} \right\}$$

On 9754 Data Display Terminals, “white” appears as “light pink”, and “yellow” as “light blue”.

With monochrome terminals (such as the 9750), the messages are output in various display modes. “Bright”, “normal”, “italics” and “underlined” here correspond to “red”, “green”, “yellow” and “white”. The operands *BLUE*, *CYAN* and *MAGENTA* have no effect.

## 4.12 Displaying a dialog on another terminal

A dialog can be output on another terminal as well as on a printer. This typically allows you to show an administrator what you are doing. This function is referred to as “listening”.

The listening function and output to printer (even by means of a K key) are mutually exclusive.

### Activation

To designate the terminal that is to be used for listening you either define it dynamically with the *HCOPY* command or declare it with *DECLARE-HARDCOPY*.

You can activate output of your dialog on this terminal by using *OPTION LISTENING = hac*, or alternatively listening can be specified with *DECLARE-TERMINAL LISTENING = hac*. OMNIS then establishes the necessary connections and displays start message *OMS0227* on both terminals. If the two terminals are not of the same type, OMNIS reports the fact by issuing message *OMS0230*.

### Operation

A terminal can only ever be connected to one listening terminal, and conversely a terminal can only listen to one other terminal.

A listening terminal must not have any other connection to OMNIS, as otherwise the dialog output on it could be interfered with by output from partners, exit routines and so on. An additional connection to which *HOLD=AUTO* applies is allowed. The *HOLD=AUTO* argument is not evaluated.

If a connection fails, the terminal which is generating the output can reactivate it with *OPTION LISTENING = hac*.

Differences between the output on the originating terminal and the listening terminal:

- Status, screen and P key prompts are not output on the listening terminal
- Overflow control is inoperative on the listening terminal
- If input is sent without the end marker (EM), the screen on the listening terminal is not cleared. The screen on the originating terminal is cleared before the output appears.
- LINE input is written on a new line on the listening terminal

All other output is identical to the extent allowed by the two terminal types.

You can call up information with *INF HCOPY*.

If the connection to the originating terminal is lost, OMNIS also breaks the connection to the listening terminal.



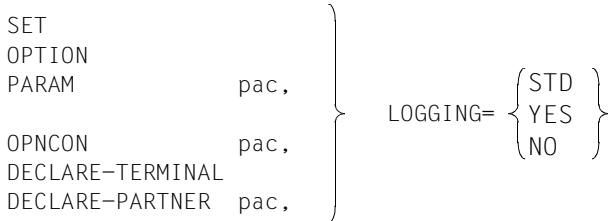
**Termination**

The listening function can be terminated in the following ways:

- You can enter *OPTION LISTENING=NO* on the originating terminal.
- You can break the connection to the listening terminal with *CANCEL*.
- If the *DECLARE-TERMINAL LISTENING = hac* assignment has been cancelled, you can clear down the connection.

### 4.13 Logging

OMNIS provides the option of writing the messages from specific partners or from all partners into a logging file. Logging is controlled by the following commands:



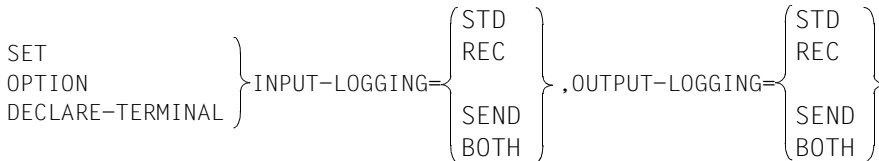
As the default option, OMNIS commands and messages are always logged.

- With *SECURITY LEVEL≠LOW*, OMNIS commands and messages are always logged even if *LOGGING=NO* is set.
- With *SECURITY LEVEL=LOW*, you can turn off the logging function by means of *SET LOGGING=NO* and *LOGGING-PRIORITY=SET-PAR-OPT*.

#### Scope and time of logging

Users can specify whether messages are logged upon input and/or upon output at the terminal. In addition, they can determine whether messages are to be logged when transmitted to and/or received from the partner.

Users are thus able to check how exit routines influence data traffic. The scope of logging and time at which it is performed are defined using the following commands:



*INPUT-LOGGING* specifies that messages entered at the terminal are to be logged.

*OUTPUT-LOGGING* specifies that messages arriving from the partner are to be logged.

*LOGGING=YES* is a prerequisite for *INPUT-LOGGING* and *OUTPUT-LOGGING*.

The following diagram shows the points at which logging takes place when the above operands are used.

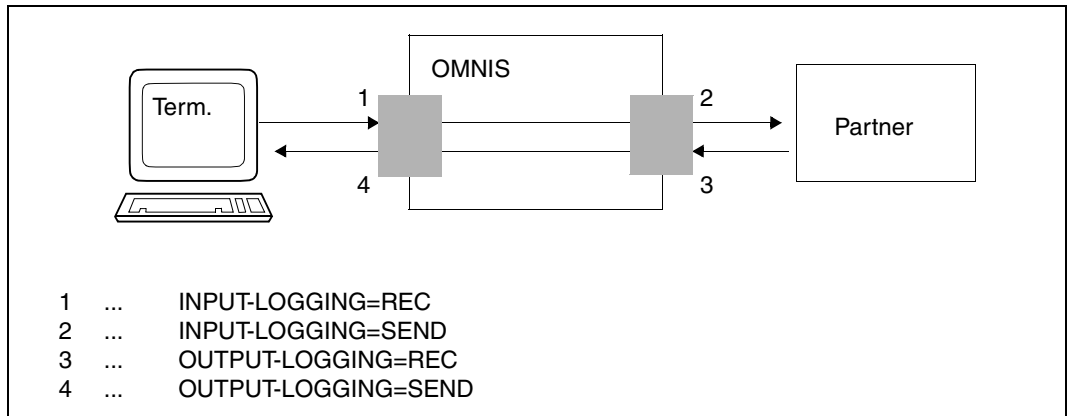


Figure 25: Effects of the various LOGGING operands

### Switching the logging file

The administrator can switch the logging file during operation without terminating OMNIS. The old logging file can then be analyzed offline.

The logging file is changed using the command `CHANGELOG [APASS=password/?]`.

For details of how to name the new logging file, see the “Administration and Programming” manual [1]. Since the *APASS* operand is rejected with a message if *SECURITYLEVEL=HIGH* is set, the *CHANGELOG* command can be entered only at the BS2000 console if *SECURITYLEVEL=HIGH*.

### Paging in the logging file

The user can only have his or her own message log displayed on the screen. This log can be paged through at any time by activating the paging mode with the *TURN* command:

```
TURN
TURN pac
```

If a partner address code is specified, only the messages of this partner are paged through; all other messages are ignored (“partner-specific paging mode”). If no partner address code is specified, the user may page through the entire message traffic associated with the terminal.

From a terminal it is only possible to page through the messages of partners which the terminal (still) “owns”. Messages that were written to the logging file before a partner was taken over with *OCCUPY* are accessible only by paging through the messages of that particular partner.

Only messages which have been logged since the last *CHANGELOG* command can be accessed.

The terminal is blocked while paging mode is active, i.e. no current messages received from the partners are displayed. OMNIS buffers these messages and displays them once paging mode has been terminated.

If the logging file is switched while paging mode is active, OMNIS automatically switches to command mode; the buffered messages are no longer displayed, but they are still available to the administrator during evaluation of the logging file.

If paging mode is active, the logging file can be paged using the commands `+`, `+n`, `++`, `-`, `-n` and `--`. Entering `0` terminates paging mode.

Formatted or physical messages are not output.

### **Blanked fields in the logging file**

If a message which is not in line mode contains blanked fields, it is replaced with an appropriate OMNIS message when written to the logging file or output to the HCOPY printer.

This functionality can be deactivated only by the OMNIS administrator for test purposes via *SECURITY LEVEL≠HIGH* and active test mode (*OPTION TESTMODE=YES*).

If the messages containing blanked fields are output to another terminal, i.e. the listening function is active, the blanked fields are also output to this terminal. Users can therefore decide themselves whether their blanked messages are to be read by others.

### **Analyzing the logging file**

The logging file is a PAM file with a special internal format and cannot be printed directly. The OMNISLC utility routine (OMNIS Logging Converter, see "Administration and Programming" manual [1]) converts the logging file into a form that can be easily printed or processed by other programs (such as EDT).

---

## 5 Functionality of OMNIS-MENU

The main purpose of OMNIS-MENU, through the provision of a menu-driven interface, is to make it easier for users to **set up and clear down connections** to their applications and switch between the applications. The values preset by the administrator mean that users no longer need to prove their identity to each individual application but just once when setting up a connection to OMNIS-MENU. Because of these **preset values**, you do not have to concern yourself with the data required for setting up a connection via OMNIS and thus no longer have to login and logout for each application.

OMNIS-MENU lets you **switch conveniently between applications**.

You can switch from one application to another: (see also [page 139](#))

- using a K or F key (and where appropriate using a P key)
- by entering *switch character* and *PAC*,  
in line mode also just by entering *PAC* :
- by implicit application switching

OMNIS-MENU fully satisfies **data protection** requirements. Access to OMNIS-MENU can be protected by means of a password and/or an ID reader. Users can/must change their own password directly. In OMNIS-MENU, each user can access only those applications allowed and assigned to him or her by the administrator. New users must be entered by the OMNIS-MENU administrator. Data security can be further enhanced by means of the option for declaring specific terminals for operation with OMNIS-MENU.

For each user a **customized user menu** can be created, which can contain up to 20 applications of any desired type.

The menu-driven user interface of **OMNISMD** facilitates creation and maintenance by the administrator of the files required by OMNIS-MENU. This task can also be performed offline with the **OMNISKD** utility. The **OMNISFC** File Conversion program makes it possible to use files from earlier OMNIS-MENU versions.

The basic functionality of OMNIS-MENU is illustrated by the next two diagrams:

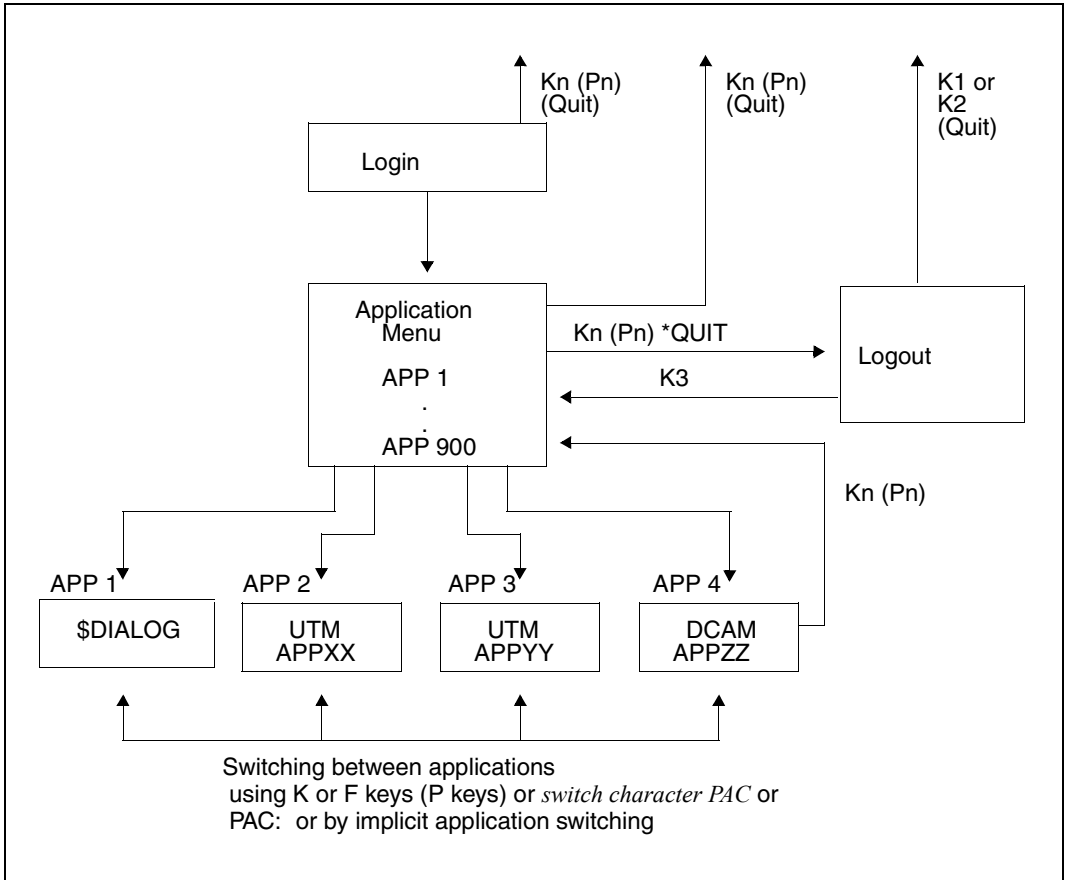


Figure 26: Functionality of OMNIS-MENU

After **setting up a connection to OMNIS-MENU**, users are presented with the *Login* mask and must identify themselves by entering their name, a password (if one is defined) and, if appropriate, inserting an ID card. If the login is completed successfully, the *Application Menu* assigned to a particular user by the administrator is displayed.

You **select the desired application** by entering the PAC or by pressing the K or F key (P key) assigned to the application. The connection to the selected application is then opened. If the activated application has a P key set assigned to it, this is loaded.

You can switch between the individual applications using K or F keys (P keys) or *switch character PAC* or, in line mode, just with *PAC:* (see also [page 139](#)) without having to call the *Application Menu*. With each **switchover** to a new application the P key set of the selected application is loaded automatically.

During operation, the individual applications can be recognized either by their screen layout or from the information in screen line 25. The latter is only possible if *LINE25 = YES* has been specified for the terminal in OMNIS (with *DECLARE-TERMINAL* in the configuration file or *SET* in the startup file; also refer to the “Administration and Programming” manual [1]).

You can return to the *Application Menu* using a predefined K key. The K key (P key) assigned to this exit function is indicated in the help line on the *Application Menu* mask (e.g: *Quit: K13/P20*).

At generation time, the administrator can specify whether the connection should be cleared down or retained when the application is exited.

If the connection to an application is interrupted while working with the application, the *Application Menu* is displayed. The interrupted connection is indicated by a flashing display of the PAC of the application affected.

Users can quit OMNIS-MENU from the *Login* mask or *Application Menu* by pressing the **logout key**. If you identified yourself with an ID card, you can also exit OMNIS-MENU by either removing the card or pressing the K14 key, regardless of where you are currently located.

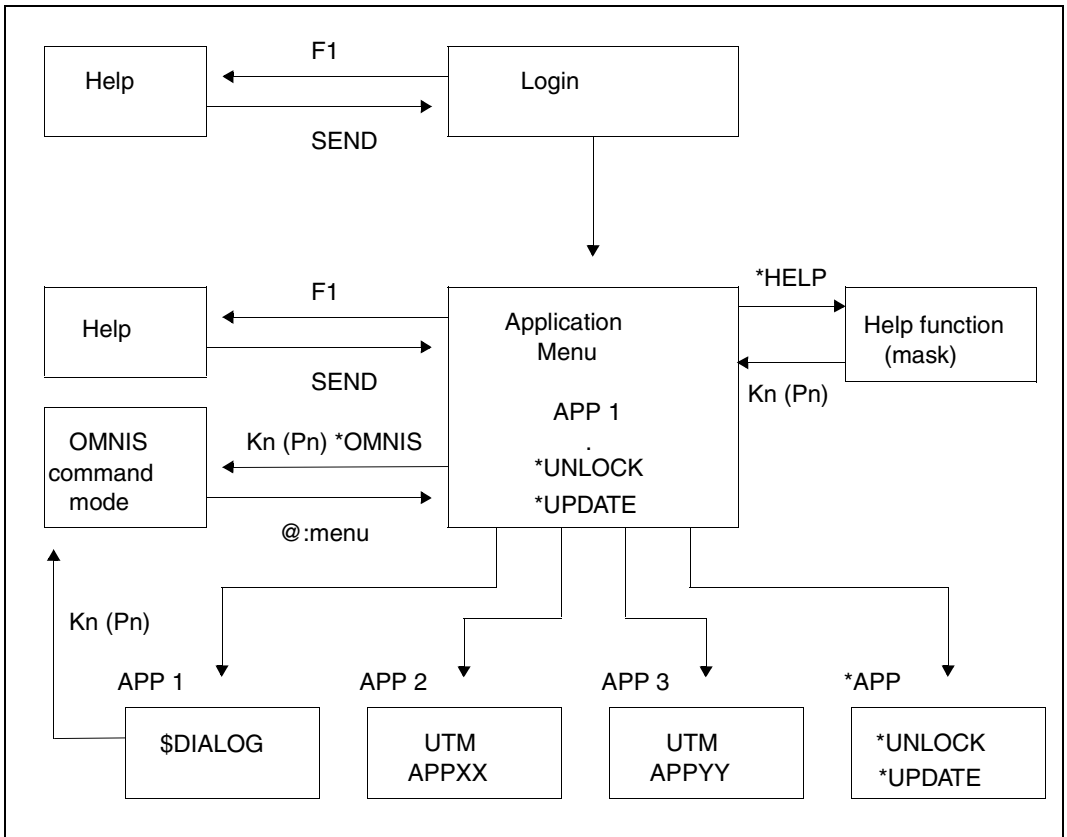


Figure 27: Functionality of OMNIS-MENU

The F1 key can be used to call a **help screen** from the OMNIS-MENU masks. Pressing the *SEND* key takes you back to the current menu. Help functions in individual applications must be implemented by the applications themselves.

A separate help mask for reading the help messages of a dialog with OMNIS-MENU is available from the *Application Menu*. A certain message can also be specified by its message number. The mask is called by entering *\*HELP* or *\*HELP <message-number>* in the command line of the *Application Menu*.

Provided the user is authorized to do so, he or she can reach **OMNIS command mode** directly by entering *\*O[MNIS]* in the *Application Menu*. In OMNIS command mode all OMNIS functions are available to the user. However, the functions of OMNIS-MENU (e.g. automatic connection setup, etc.) do not work in OMNIS command mode. You can also use a K or P key to switch to OMNIS command mode, either from the *Application Menu* or directly from an application. By entering the command *@:menu* you can return to the *Application Menu*.



OMNIS-MENU also provides the user with **administrator functions**. These functions are identified by the asterisk (\*) in the first position of the menu name (e.g. the *\*UNLOCK* function).

The functionality of OMNIS-MENU can be extended by user-defined exit routines (see the “Administration and Programming” manual [1]). Extensions and modifications of this type are required because of the specific characteristics of certain applications.



---

## 6 The OMNIS-MENU user interface

The following points should be noted when using the masks of OMNIS-MENU:

### **Data input**

No distinction is made between uppercase and lowercase input.

You can move the cursor from one input field to another with the TAB keys.

You fill out the appropriate fields in each mask and send off the data using the SEND key.

The last two lines of each mask are reserved for messages from OMNIS-MENU and OMNIS.

### **Help function**

You activate the help function by pressing the F1 key. The help mask contains supplementary information relating to the contents of the current screen. Pressing the SEND key returns the display to the current menu.

## 6.1 Logging in to OMNIS-MENU

### Setting up the connection

The connection to OMNIS-MENU is set up as follows:

$$0 \text{ application-name}[\left. \begin{array}{l} \text{processor-number[/region-number]} \\ \text{processor-name} \end{array} \right\} \left[ \begin{array}{l} \text{,opch=idr]} \\ \text{[,pw=...]} \end{array} \right]$$

### Meaning of the operands

**application-name**

is the name of OMNIS-MENU and is defined by the OMNIS administrator (DCAM application name).

**processor-number**

used together with *region-number* to define the network address of the computer on which OMNIS-MENU is started.

**region-number**

used together with *processor-number* to define the network address of the computer on which OMNIS-MENU is started.

**processor-name**

Name of the processor on which OMNIS-MENU is started.

**opch=idr**

This operand is required if the terminal has an ID card reader, and *opch* was not specified during PDN generation.

**pw=...**

is the OMNIS-MENU connection password and is defined by the OMNIS-MENU administrator.

### Opening an OMNIS-MENU session

After the connection has been set up, the *Login* mask is displayed. In order to open your OMNIS-MENU session, you will need to fill in the input fields and send off the mask with the SEND key.

Since the layout of the login mask can be modified by the administrator (see the “Administration and Programming” manual [1]), the following illustration must be seen as only one example of a login mask.

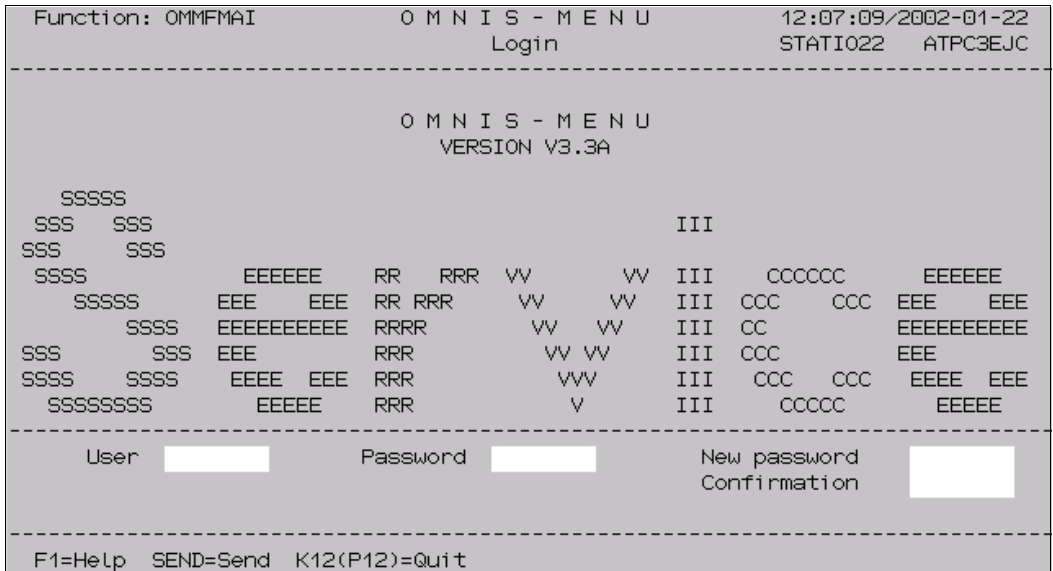


Figure 28: Login

The user name and the associated password must be entered in this mask. In addition, any user can optionally replace his or her old password with a new one. This also applies to expired passwords.

With the global parameter *CONNECTION-ATTEMPTS*, the OMNIS-MENU administrator can specify how many unsuccessful login attempts OMNIS-MENU is to permit from a terminal. If this value is exceeded, the related terminal is locked.

The administrator can unlock the terminal again with the administrator function *\*UNLOCK*.

## Meaning of the fields

### User

The user name is entered here. This is a name assigned to each user by the administrator (length not exceeding 8 characters).

### Password

The password is entered here. Entry takes place in blanking mode (length not exceeding 8 characters). The initial assignment is made by the administrator.

A user can be assigned no password with the specification *\*NONE*. No new password can subsequently be assigned to this user. Access to this user is permitted when either no password or the password *\*NONE* is specified.

### New password

The field can be used to define a new password (length not exceeding 8 characters). Entry takes place in blanking mode. The new password will be in effect the next time you login to OMNIS-MENU. The password must meet the conditions defined by the administrator (see the “Administration and Programming” manual [1], password complexity in DECLARE-PARAMETER).



For security reasons, the last ten passwords used cannot be used again.

### Confirmation

In order to detect possible keying errors, the new password must be entered a second time in the *Confirmation* field. Only when both entries are identical will the new password be accepted by OMNIS-MENU.

If an ID card is generated for the user, then after the authorization information has been entered the screen is cleared and the ID card requested separately with the message `PUSH CARD INTO CARD-READER`.

If the specified entries are correct, the *Application Menu* is displayed.

## 6.2 OMNIS-MENU Application Menu

After you have successfully logged on, the *Application Menu* is displayed.

```

Function: OMMFSEL          O M N I S - M E N U          12:08:14/2002-01-22
User      : KROTTEND      Application Menu          STATI022  ATPC3EJC
-----
                                                                01/03

      *** Entwicklung          *** Test-Partner
<ME>  1/133  OMSME          K9  P9  <UT11> OMUTM11 DIREKT
<MW>  1/133  OMSMW          K10 P10 <MX11> OMUTM11 UEBER MUX
<OE>  1/133  OMSE (DSSIM) K8  P7  <UT12> OMUTM12 DIREKT
<OW>  1/133  OMSW          K7  P8  <MX12> OMUTM12 UEBER MUX
<OQ>  1/133  OQ           <UT01> OMUTM01 DIREKT
                                     <MX01> OMUTM01 UEBER MUX
                                     <VM3>  29/214 OMSE VM3
                                     <017>  D017ZE00 SBZ  V11.2
                                     <V112> BLOCK V11.2 V214H028

ID      [ ] Password [ ] Passing (Y/N) N
Command [ ]

-----
F1=Help K12(P12)=Quit K11(P11)=OMNIS Command Mode

```

Figure 29: Application Menu

When the *Application Menu* is displayed, the cursor is located in the command line. The following functions are available on this menu:

- application selection
- entry of special data for signing on to UTM applications that support the multiplex protocol
- calling a help menu with the F1 key
- calling a help function with the OMNIS command *\*HELP*
- switching to the OMNIS command mode (only if the user is authorized to do so)
- setting up a connection to an application
- clearing down one or more connections
- logging out from OMNIS-MENU
- scrolling in the application list.

### Selecting an application

To select an application enter the PAC of the desired application in the command line and press the SEND key. The PACs are located ahead of the name of the application assigned by the administrator. Any application that has been assigned a K or F key (P key) by the administrator may also be started by pressing the corresponding K/F or P key. The key assignments are located in the columns *K/F* and *P*. You can also select applications in your user menu that are not currently displayed on the screen.

### Defining parameters in dialog

After selecting an application in the form *Application Menu* the mask *Predialog* (OMMFPRE) is issued if at least one of the attributes *PRONAME*, *PTNNAME* or *TYPE* is not defined yet (specification of *\*PREDIALOG* in OMNISKD or OMNISMD). You can define these attributes in the mask *Predialog* then.

```

Function: OMMFPRE          O M N I S - M E N U          12:20:59/2015-03-05
User   : REHWALDH          Predialog          STATFAC0  MCH4473N
-----
COMMENT: TYPE=*PRE,PRONAME=*PREDIALOG,PTNNAME=*PREDIALOG

Open connection  PAC    = PRE1
Host name        PRONAME = 
Partner name     PTNNAME = 
Partner typ      TYPE   =       (TIAM / UTM / DCAM)
-----
F1=Help DUE=Send K13(P19)=Quit
    
```

Already defined attributes are integrated into the mask, but they cannot be modified. The *TYPE* attribute may only be one of the types TIAM, UTM and DCAM. To assist the user the attribute COMMENT is integrated into the mask.

The default values of OMNIS also apply here:

- in absence of *PRONAME* the OMNIS-own host is assumed
- in absence of *TYPE* the type TIAM is assumed
- in absence of *PTNNAME* and *TYPE=TIAM* the application \$DIALOG is assumed. *TYPE=TIAM* and *PTNNAME=\$DIALOG* are allowed only in this combination.



## Signing on to a UTM application

The *ID*, *Password* and *Passing* fields relate to the procedure for signing on to a UTM application. These three fields are evaluated only for multiplex connections. If you do not know which applications these are, consult your OMNIS-MENU administrator. If an application is opened with a K or F key (P key), the *ID*, *Password* and *Passing (Y/N)* fields are not interpreted.

### Meaning of the fields

#### ID

If a user ID is specified here, this will be transferred to UTM instead of the name entered under *User* in the *login* mask. It is thus possible for the user to be generated with different user IDs for different UTM applications. (Length not exceeding 8 characters)

#### Password

If a password is specified here, this will be transferred to UTM instead of the name entered under *Password* in the *login* mask. OMNIS-MENU and UTM may thus use different passwords.

The password must not be entered in hexadecimal code.

#### Passing (Y/N)

Entering *Y* in the *Passing* field in the *Application menu* allows you to assign a new password for the UTM application (for multiplex connections only). Before signing on to the UTM application, the system requests confirmation by outputting the *Change password* mask.

This mask contains the fields *ID*, *Password*, *New password* and *Confirm new password*. The values are taken from the *Application menu* or *Login* masks in accordance with the *SIGNON* parameter.

The *Change password* mask allows you to:

- Send the fields which have already been filled in without modifying them
- Modify the fields and send them
- Cancel the login operation

If *Passing=Y*, the password from the login mask is taken, irrespective of the *SIGNON* parameter.

If *Passing=N* (default), the *Change password* mask is not output and the old password remains valid in UTM.

### Switching to OMNIS command mode

If you have the authorization, you can switch directly to OMNIS command mode by entering *\*O[MNIS]* in the command line or pressing the appropriate K or P key. Message OMM0016 (OMNIS-MENU deactivated) is displayed.

The user can then use all the functions of OMNIS. The command *@:menu* is used to return to OMNIS-MENU.

Authorization to switch to OMNIS command mode is granted by the administrator via OMNISKD or OMNISMD.

### Calling a help function with the OMNIS command \*HELP

By entering *\*HELP* in the command line, you can switch to the help function window. If you enter *\*HELP <message number>*, the message text of this message is immediately shown when the help function window opens. The meaning of the message texts and the action to be taken are also displayed. A diagram of the window and a description of the help function are provided on [page 147](#).

### Starting applications and clearing down one or more connections

In the *Application menu* the *No.* column contains marking fields in front of the PACs.

By marking fields with *c* you can cause the relevant application to be cleared down. This makes it possible to clear down two or more connections to any type of applications simultaneously.

Marking with *x* causes a connection to the application concerned to be started or reactivated.



It is not possible to clear down and open applications at the same time. It is not possible to open several applications at the same time. If even one connection is marked with *c*, any entry in the command line is ignored.

### Scrolling in the application list

You can use *+, -, +n, -n, ++, --* and *SEND* to scroll in the application list a page at a time. Here *n* specifies the number of screens you want to scroll forward or back.

The maximum number of screen pages in a menu is 45. 20 entries are possible per screen page, thus permitting a maximum of 900 menu entries.

You can use page-by-page scrolling in conjunction with an appropriately structured application list, for example, to always see a complete subject area on the screen.

## 6.3 Application switching

OMNIS-MENU supports both explicit and implicit application switching. Implicit application switching gives users the impression that they are still working with a single application.

### 6.3.1 Explicit application switching

There are three methods of explicit application switching:

- via the *Application Menu*
- with K or F keys (P keys)
- by entering *switch-character PAC* or just *PAC*:

You can return to the *Application Menu* from an application in two ways:

- via the relevant go-back key
- in line mode, using the command @:

#### Switching applications by means of the K or F keys (P keys)

The administrator can additionally assign K or F keys (P keys) in a user menu for each application. The P keys are loaded automatically with K/F keys (ESC sequences) after you have logged into OMNIS-MENU. The number of F keys available depends on the terminal type. The F1 key is reserved for the OMNIS-MENU help function.

By entering the PAC or pressing the appropriate K, F or P key, you can set up a connection to an application and load the P key set generated for this application. The K/F keys are not passed on to the application.

When you are in an application, you can switch directly to another active application using the appropriate K, F or P key; the current P key set assigned to the selected application is loaded automatically.



Application switching by means of K/F keys may be restricted because an application uses these keys for its own purposes.

In a “strict dialog” (indicator **DIA**), the K/F keys **cannot** be triggered via P keys. In this case, and if a P key assigned for switching to a different application is (inadvertently) reassigned, the only way you can switch to this application is by using the appropriate ESC sequences of the K or F keys.

The following ESC sequences should be entered for keys K4 ... K14:

K4	ESC V	K10	ESC >
K5	ESC W	K11	ESC =
K6	ESC M	K12	ESC <
K7	ESC N	K13	ESC ;
K8	ESC O	K14	ESC :
K9	ESC ?		

The following ESC sequences should be entered for keys F4 ... F24:

F4	ESC ^	F15	ESC y
F5	ESC _	F16	ESC z
F6	ESC p	F17	ESC {
F7	ESC q	F18	ESC
F8	ESC r	F19	ESC }
F9	ESC s	F20	ESC —
F10	ESC t	F21	ESC l
F11	ESC u	F22	ESC m
F12	ESC v	F23	ESC n
F13	ESC w	F24	ESC o
F14	ESC x		

The F4 ... F24 keys are not available on all display terminals.

### Switching applications with switch character PAC, or PAC:

You need to enter *switch character PAC* to switch applications directly if you are engaged in formatted dialog with an application and no K/F/P key is generated for the application you wish to switch to. This also works in line mode, but in this case you only need enter *PAC*:

The switch character is globally defined by the administrator.



You cannot send a message to an application solely in the form *switch-character PAC*. When you enter *switch-character PAC*, OMNIS-MENU interprets this as an application switch. Nor is it possible to send a long message.

### 6.3.2 Implicit application switching

Implicit application switching gives users the impression that they are still working with a single application. The user enters a single TAC (transaction code), and OMNIS-MENU performs all further steps required to switch applications without this being apparent to the user.

Implicit application switching can also take place between applications of different types, e.g. between DCAM and UTM applications. The applications can also reside on different hosts.

Implicit application is typically useful if a large UTM application is split into a number of applications to improve performance but should still appear to be a single application for the user.

#### Using the “implicit application switching” function

You can activate and deactivate the “implicit application switching” function. However, this only applies if a TAC allocation table exists. Otherwise OMNIS-MENU issues the message OMS0655.

The K or P key for enabling and disabling “implicit application switching” can be defined in the OMNIS configuration file using the OMNISKD control statement DECLARE-PARAMETER. Details of the effectiveness of “implicit application switching” can also be specified.

You can activate and deactivate the function in the *Application Menu* mask. In the *Application Menu* mask the current status is shown on the Help line.

When the “implicit application switching” function is activated, you enter the required TAC (transaction code) in a dialog with an application in the input field provided for this purpose, and OMNIS-MENU then performs the appropriate application switch for you.

If an implicit application switch entails setting up another connection, OMNIS-MENU sets up the connection automatically. OMNIS-MENU messages inform you if this is not successful, and OMNIS-MENU returns to the *Application Menu* mask.

## TAC allocation table

With the aid of the TAC allocation table, OMNIS-MENU decides which application is activated when implicit application switching takes place. The administrator creates the TAC allocation table, and it is globally valid.

A TAC (transaction code) uniquely identifies an application. A TAC is up to 8 characters long.

The TAC allocation table is a SAM file. The administrator decides on the order of the entries and also on the priority for dealing with them (see also Checks by OMNIS-MENU, TAC check).

To declare an entry in the TAC allocation table you use the OMNISKD *DECLARE-TAC* statement.

OMNIS-MENU is not fundamentally able to distinguish a TAC entry field from other data entry fields:

The TAC entry field must always be the first field in the mask. The TAC entry field can be up to 80 characters long (but only in one line).

OMNIS-MENU checks the contents of the TAC entry field up to the first blank and then compares the contents with the entry in the allocation table. On finding the first application with the corresponding menu name, OMNIS-MENU responds in accordance with the behavior defined for implicit application switching (“switch only” or “switch and send the TAC”). Caution: As it is basically also possible to switch to TIAM applications, the administrator must take care to keep to the order of the applications in the menus.

If no connection to the new application exists, OMNIS-MENU establishes one. Before the message that has been input is sent, OMNIS-MENU sends a start sequence that is contained in the text file. Caution: Only one command may be defined in the text file for applications that are implicitly accessible.

If the TAC is not found in the TAC allocation table, OMNIS-MENU sends the message to the current application.

With applications that support formats, the TAC entry fields must be at least 8 characters long.

When programs for mask support, such as FHS, are used, it is possible that not all entry fields will be (automatically) transferred (“automatic input”). Consequently it may be the case that OMNIS-MENU cannot transfer the TAC entry field to UTM.

*OMNIS commands under OMNIS-MENU*

TACs can also be used under OMNIS-MENU to pass on OMNIS commands to OMNIS. Since this also involves an implicit application switch, the TAC must always be the first field of the mask.

The TAC, which can be up to 4 characters in length, represents a valid OMNIS command. The OMNIS command can be specified in its minimum shortened form or in full, but may not be longer than 4 characters. The input in the command line of OMNIS-MENU contains the transaction code first, i.e. a valid OMNIS command, and then the operands of the command (e.g. HELP OMS0001), separated by blanks.

The corresponding entry in the TAC allocation table is declared with the OMNISKD statement *DECLARE-TAC*. The statement identifies the OMNIS application, with *PTNNAME* identifying the name of the application and *PRONAME* the processor name of OMNIS. The *SWITCH* operand must always be set to *SEND*.

*Examples of DECLARE-TAC statements for OMNIS commands:*

```
DECLARE-TAC TAC=HELP, PRONAME=G133H207, PTNNAME=NULLIS, SWITCH=SEND
DECLARE-TAC TAC=OPTI, PRONAME=G133H207, PTNNAME=NULLIS, SWITCH=SEND
DECLARE-TAC TAC=INF, PRONAME=G133H207, PTNNAME=NULLIS, SWITCH=SEND
```

**Initialization, configuration update**

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

An OMNIS-MENU configuration update during operation (administrator function *\*UPDATE*) also affects the global TAC allocation table.

As with the OMNIS-MENU generation data, the changes in the global TAC allocation table take effect immediately.

**Checks by OMNIS-MENU**

In the **initialization phase** OMNIS-MENU checks whether the globally defined key K13 (for activating/deactivating the “implicit application switching” function) is unique with respect to the reset key and the OMNIS command mode key. If the key is not defined uniquely, the “implicit application switching” function cannot be used. OMNIS-MENU then issues message OMM0654 (key ambiguous).

OMNIS-MENU checks the following for each record in the TAC allocation table:

- Is the TAC or the network address undefined?
- Is the characteristic part of the group TAC no more than 4 characters long?
- Is the TAC unique with respect to the menu name in the TAC allocation table?
- Is the behavior in the event of implicit application switching entered correctly (blank or 'S')?

In the event of an error OMNIS issues one of the messages OMM0650 to OMM0653 to SYSOUT or SYSLST and the “implicit application switching” function cannot be used.

In a **dialog** with an application OMNIS-MENU examines every input message in the following order:

1. Input with DUE1:
  - a) Partner switch with <pac>, via OMNIS?
  - b) No partner switch via OMNIS: occurrence of switch character followed by PAC?
  - c) “Implicit application switching” function, TAC or group TAC in message?

The TAC check for implicit application switching proceeds as follows:  
First OMNIS-MENU checks whether the specified TAC matches a defined TAC.  
Then the group TACs that are 4, 3, 2 and 1 characters long are checked, in exactly that order.  
This test mechanism allows exceptions from TAC groups to be defined.  
If, for example, the TAC allocation table contains the TAC XY and the group TAC X\*, OMNIS-MENU switches to the TAC XY (and not X\*) when XY is entered).
2. Input with the HCOPY key?
3. Input with the reset key?
4. Input with the OMNIS command mode key?
5. Input with an application's K/F key?
6. Activation/deactivation of implicit application switching with K13 (lowest priority)?  
OMNIS-MENU issues one of the messages OMM0038 or OMM0039 (activated/deactivated).



## 6.4 Logging out of OMNIS-MENU

If you are in either the *Application Menu* or *login* mask, you can log out from OMNIS-MENU by pressing the logout key. The key assigned to this function is shown in the lower left-hand corner of the relevant mask, after *FI=Help*. Another way of logging out from OMNIS-MENU is by entering *\*Q[UIT]* in the command line.

Users generated as having an ID card can also quit OMNIS-MENU either by removing the ID card or by pressing the K14 key.

Using the global parameter *DELETE-PKEYS*, the administrator can specify whether the settings of the P keys should be retained or deleted after logout from OMNIS-MENU.

### Calling the logout mask

Depending on the generation, you can have the following mask displayed when you press the logout key in the *Application Menu*.

```

Function: OMMFEND          O M N I S - M E N U          12:09:11/2002-01-22
User      : KRÖTTEND          Logout          STATI022  ATPC3EJC
-----
You have asked to quit OMNIS-MENU. The following options are available
to you:

K1 - Logs you out from OMNIS-MENU and disconnects you from all
    applications.

K2 - Logs you out from OMNIS-MENU but existing connections to
    applications are retained. You can subsequently log in again
    to OMNIS-MENU and continue working with your old sessions.

K3 - Takes you back to the Application Menu.

-----
Continue with K1, K2 oder K3

```

Figure 30: Logout

The required function is activated by pressing the K key assigned to that function.

Since the logout mask is self-explanatory, there is no corresponding help mask.

*Using the K2 key*

The K2 option can be set in the OMNIS *Logout menu* to retain connections on a user-specific basis. The OMNISKD control statement DECLARE-USER can be used to define whether a particular user can be provided with the K2 option in the logout mask (OMMFEND).

```
DECLARE-USER . . . ,LOGOUT-HOLD-AVAILABILITY=YES | NO
```

The connection can be retained on a user-specific basis only if the global parameter *ONE-USER-ONE-CONNECTION=YES* is set. It is therefore better to work with *ONE-USER-ONE-CONNECTION=YES*.

However, if global parameter *ONE-USER-ONE-CONNECTION=NO* is set, the user names are not unambiguous and consequently the connection cannot be retained on a user-specific basis. In this case OMNIS-MENU issues message OMM0034 (K2 key). You can abort only with K1. All connections are then closed.

## 6.5 Viewing error information

If you enter *\*HELP* in the command line of the *Application Menu*, the following mask appears on the screen:

```

Function: OMMFSHE           O M N I S - M E N U           12:14:04/2002-01-22
User      : KROTTEND        Help function           STATI022   ATPC3EJC
-----
OMM0011 CONNECTION SETUP INITIALIZED
OMM0012 CONNECTION LOST TO APPLICATION ($DIALOG/G133H207)
OMM0027 INVALID COMMAND
OMM0014 F OR K KEY NOT DEFINED IN MENU
OMM0011 CONNECTION SETUP INITIALIZED
OMM0013 CONNECTION SETUP TO APPLICATION ($DIALOG/DB16ZEB4) NOT EXECUTED; REASON
= 0C480000
OMM0033 END OF MENU LIST REACHED
OMM0032 START OF MENU LIST REACHED
OMM0029 OMNIS-MENU CONFIGURATION HAS BEEN CHANGED SUCCESSFULLY
OMM0011 CONNECTION SETUP INITIALIZED
OMM0012 CONNECTION LOST TO APPLICATION ($DIALOG/G133H207)
OMM0027 INVALID COMMAND
-----
Command:(++,--,+,-,*HELP,*HELP Msgnr)      K12(P12)=Quit

```

Figure 31: Help function

If you enter *\*HELP <message number>* , the message text of that message is immediately shown when the help function window opens. The meaning of the message text and the actions to be taken are also displayed immediately.

By means of +,-,++,-- and SEND, you can also page through various screens of messages in the window. In this way, you can review all the messages output by OMNIS-MENU. The memory capacity of the messages viewable by paging cannot exceed 4096 bytes. When this memory limit is reached, the oldest message is deleted from memory.

## 6.6 Diagnostic information in the event of an error

Messages from OMNIS and OMNIS-MENU are displayed in the last two lines of the menus.

Error analysis is possible with the aid of the following information:

- error number
- precise information on the terminal inputs leading to the error
- copy of the tables of OMNIS-MENU
- OMNIS logging file (unedited)
- where appropriate, a dump
- SYSLST and SYSOUT log
- trace files

---

## 7 Special considerations of OMNIS-MENU

This chapter deals with special points that must be observed with respect to application connection setup, application switchover, K, F and P key management, ID card information and chip cards, as well as special features involved in administrator messages.

### 7.1 User names and passwords in OMNIS-MENU

#### Connection setup

The dialog between OMNIS-MENU and a TIAM application when a connection is set up can be represented as follows:

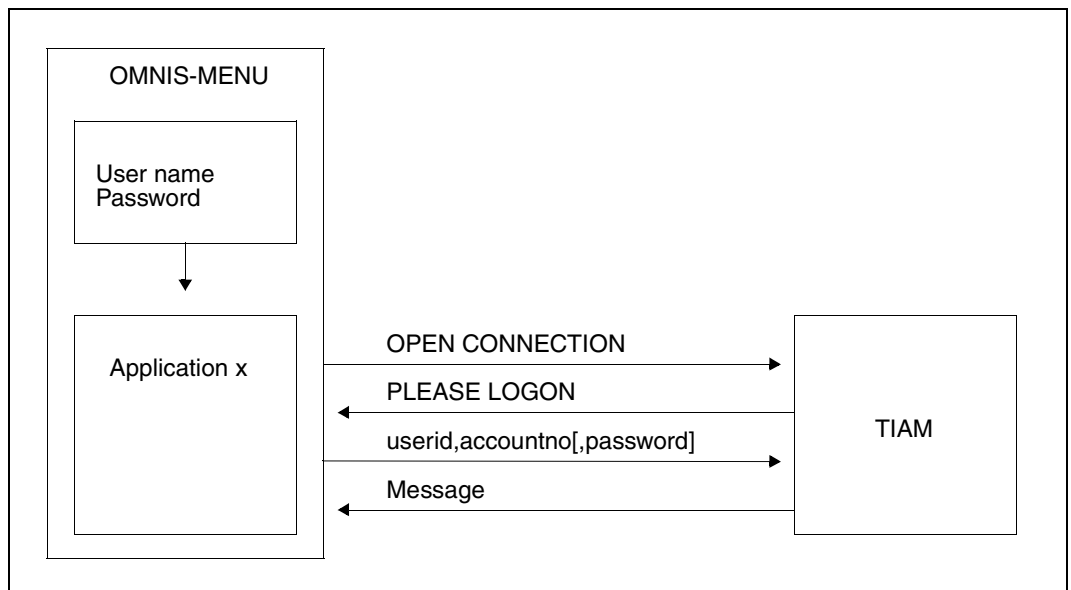


Figure 32: Connection setup OMNIS-MENU/TIAM

For automatic *LOGON*, the *LOGON* command is taken from the OMNIS text file. For reasons of data security, the password should not be located in the text file and must be entered by the user when it is requested (input is blanked).

Further commands from the text file can call a procedure or an application in order that the format for the application appears immediately after *OPEN CONNECTION*.

A similar dialog takes place when a connection is set up to a UTM application:

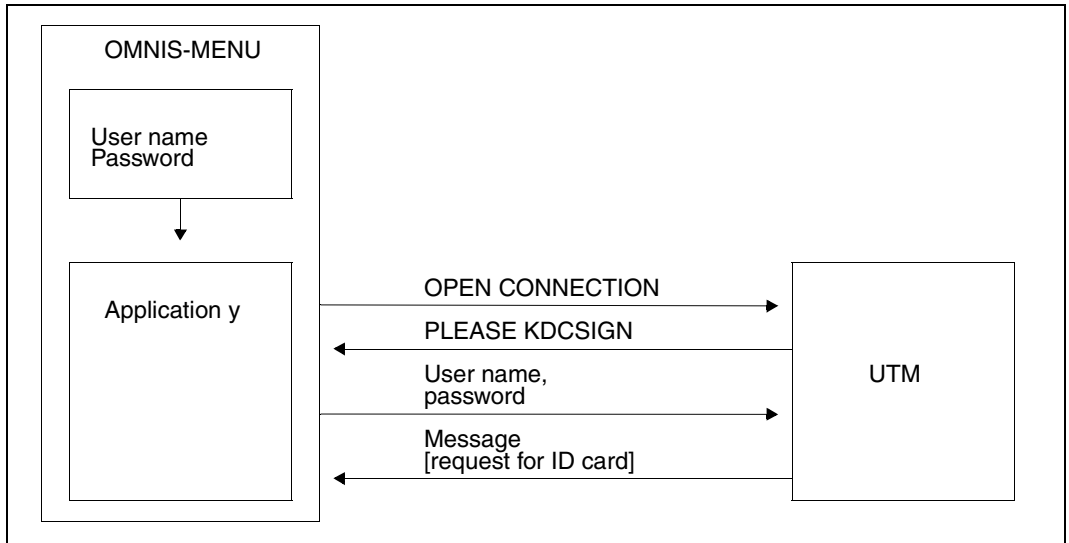


Figure 33: Connection setup OMNIS-MENU/UTM

For *KDCSIGN* the user name can be taken from the OMNIS text file (see also “Text file” and “User-specific start sequences” in the “Administration and Programming” manual [1]). For reasons of data security, the password should not be located in the text file and must be entered by the user when it is requested (input is blanked).

In the special case of multiplex connections, connection setup is possible with *OPEN SESSION*. At *session* setup, the user name, password and, where appropriate, ID card information can be passed to UTM.

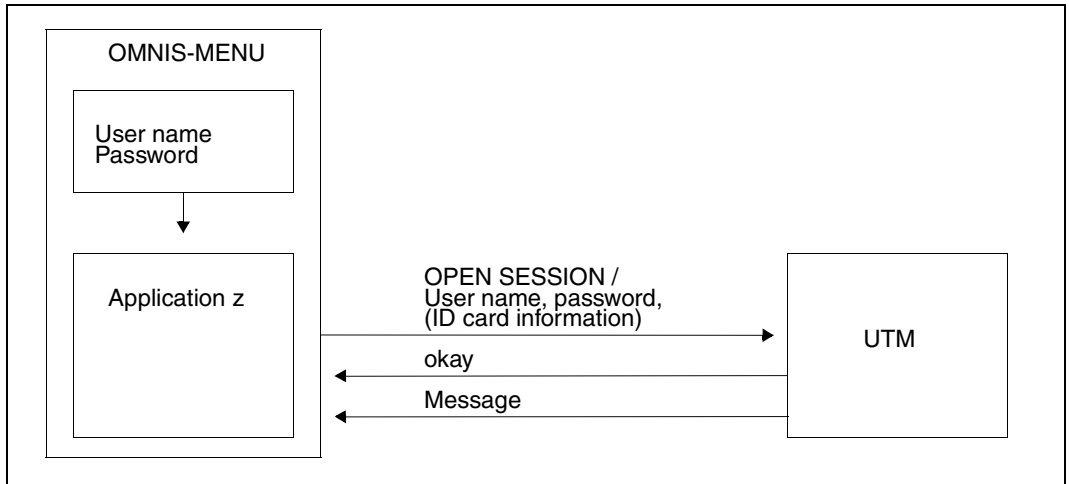


Figure 34: Connection setup OMNIS-MENU/UTM as of V3.1 (multiplex connection)

If a user name (password) unknown to UTM is specified here, OMNIS-MENU issues error message `OMM0015`. The data can be corrected in the *Application Menu* (see [page 135](#)).

When a connection is set up to a UTM application via a multiplex connection, the specifications in the *Login* mask and *Application Menu* are evaluated in accordance with the generation of the application with OMNISKD or OMNISMD.

The following table indicates which information for different entries in the *Application Menu* is passed to the UTM application with *OPEN SESSION* for the *SIGNON* data preset with OMNISKD or OMNISMD:

SIGNON data / OMNISKD or OMNISMD	Input data OMNIS-MENU (Application Menu)			
	No entry	User	Password	User and password
B(oth) (user name and password)	User name (A) Password (A) (ID info)	User name (B) Password (A) (ID info)	User name (A) Password (B) (ID info)	User name (B) Password (B) (ID info)
U(ser name)	User name (A) (ID info)	User name (B) (ID info)	User name (A) Password (B) (ID info)	User name (B) Password (B) (ID info)
N(one)		User name (B) (ID info)	Error message	User name (B) Password (B) (ID info)
S(uppresse)				

User name (A) = Entry in the *Login* mask (OMMFMAI)

User name (B) = Entry in the *Application Menu* (OMMFSEL)

Password (A) = Entry in the *Login* mask (OMMFMAI)

Password (B) = Entry in the *Application Menu* (OMMFSEL)

ID info = ID card information

Error message = OMNIS-MENU rejects the input(s) with an error message.

The following table shows how UTM responds to the data that has been entered in the *Application Menu* and passed to UTM in accordance with the OMNISKD or OMNISMD generation (see table above), assuming that the *SIGNON* data is correct.



UTM KDCDEF	Data passed OMNIS-MENU -> UTM			
	No entry	User name	User name and password	User name, password and Passing = Y
User name Password	KDSIGN or connection	BIND-REJECTED	BIND-ACCEPTED	BIND-ACCEPTED ***
User name	KDSIGN or connection	BIND-ACCEPTED	BIND-REJECTED	BIND-REJECTED
No entries	BIND-ACCEPTED	BIND-ACCEPTED	BIND-ACCEPTED	BIND-ACCEPTED

BIND-ACCEPTED = The sign-on data is accepted by UTM.

BIND-REJECTED = The sign-on data is not accepted by UTM.

\*\*\* = The new password specified in the login mask is taken over by UTM.

In the event of an error (incorrect *sign-on* data sent), UTM sends a negative *BIND-RESPONSE* to OMNIS-MENU and the user is requested to reenter the data.

### Changing passwords with multiplex connections

It is possible for the user to change his or her password when logging on to OMNIS-MENU (see [page 132](#)). This could result in discrepancies between the passwords used in OMNIS-MENU and in UTM.

The user can enter his or her old password in the *Application Menu* and, by entering *Y* in the *Passing* field, have the new password taken over for use in UTM.

Such changes of password must be carried out for all UTM applications in the described form so that no further entries are required when a connection is next set up.

## 7.2 Management of K, F and P keys

In OMNIS-MENU, each application can have a K key or F key assigned to it by the administrator. The keys thus assigned can also be loaded into P keys.

At generation time the administrator can specify whether the settings of the P keys should be retained or deleted when logging in to or logging out from OMNIS-MENU.

### P keys

A P key set can be defined and generated for each application (see the “Administration and Programming” manual [1]). This set of P keys generated with the utility routine PLUS is loaded automatically by OMNIS-MENU when a switchover is made to this application. If a P key set is assigned to an application in OMNIS-MENU and *SAVE=PKEY* or *SAVE=ALL* is set, the saved P keys are restored when you switch applications, not the generated P key set.

It is also possible for the user to assign P keys while working with an application. In doing so he or she may overwrite assignments made by the administrator for the application itself or for switching to other applications.

If the P keys of an application are loaded in such a way that they cannot be overwritten (function control character / programming lock P keys), OMNIS-MENU will not be able to reload the P keys.

If the P key for OMNIS command mode is also assigned to an application, the Application Menu function will predominate and the P key is not displayed in the field *OMNIS command mode* of the *Application Menu*.

### K keys

When assigning K keys, make sure that the logout key is uniquely assigned. If the corresponding K key is inadvertently assigned to an application, the reset function will predominate. The application can then be called up only by entering the PAC or *switch character PAC*.

If the K key for OMNIS command mode is also assigned to an application, the application selection will predominate and the K key is not displayed in the *OMNIS command mode* field of the *Application Menu*.

The K key for OMNIS command mode must not be confused with the KPAC in OMNIS. When using OMNIS-MENU you should avoid the use of KPAC.

If the K key assigned to an application is pressed during the dialog with this application, the K key is passed on to the application. K keys are intercepted by OMNIS-MENU only in the case of an actual change of partner.

Since K14 is reserved for the ID card, this key cannot be assigned (see [page 158](#)).

### **F keys**

The number of F keys available for assignment depends on the type of terminal.

The F1 key may not be assigned, as it is reserved for the OMNIS-MENU help function.

If the F key assigned to an application is pressed during the dialog with this application, the F key is passed on to the application. F keys are intercepted by OMNIS-MENU only in the case of an actual change of application.

## 7.3 Application switching behavior

When you switch applications, OMNIS-MENU supports the functions described below.

### Automatic disconnection

When an application is defined, the *automatic disconnection* function can be used to specify that the connection to the application should be cleared down automatically when the application is exited.

### Save screen contents and P keys

The *save screen contents* function of OMNIS-MENU is automatically activated with the *SAVE* operand when a connection is set up to the application and is then executed by OMNIS.

When the application is reselected, the saved screen will be displayed at the terminal. Note that in line mode the screen is cleared after the next input.

If an error occurs during restoration of the screen contents of an application, a restart message is sent to the application.

The *save P keys* function is also activated automatically by OMNIS-MENU with the *SAVE* operand when a connection is set up to the application and executed by OMNIS.

If a P key set is assigned to an application in OMNIS-MENU and *SAVE=PKEY* or *SAVE=ALL* is set, the saved P keys are restored when you switch applications, not the generated P key set.

### Message flow control

The *message flow control* function is activated with the *EXCLUSIVE-PARTNER* operand when you log into OMNIS-MENU and is executed by OMNIS.

### Restart message

The restart message can be specially defined for each individual application. However, OMNIS-MENU will only use this administrator-defined restart message if the application is in phys or formatted mode.

If no special restart message has been defined, or if the application is in line mode, a “standard message” is sent to the application.

No restart message can be sent to UTM MUX partners, since a *REPEAT* is always sent in this case.

**Standard restart message for applications in formatted mode**

The UTM command *KDCLAST* or, in the case of a multiplex connection (*ROUTE=MUX*) the protocol element *REPEAT*, is sent to a UTM application. The UTM application then repeats its last output, which OMNIS-MENU forwards to the terminal.

No defined screen restart provisions exist for DCAM and TIAM applications. Where the partner type is not UTM, the same message is sent to such partners as when deactivating and reactivating the terminal and entering K3. If FHS is used, the return code “Screen destroyed” may be reported by FHS and it is the responsibility of the application to process this return code correctly.

When switching between applications in formatted mode, it is important to ensure that the applications have a screen restart capability, i.e. that the screen can be reconstructed with the last format displayed before the switch.

For applications that do not have a screen restart capability, you should use the *save screen contents* function.

**Standard restart message for applications in line mode**

In the case of DCAM and UCON applications in line mode, a slash is displayed at the terminal as a data-entry prompt. TIAM applications are sent the code for SEND.

## 7.4 Handling ID card information

OMNIS-MENU behaves differently depending on whether or not ID card information was specified when the user was generated:

### **ID card information was specified when the user was generated**

OMNIS-MENU reads in the ID card information when the user logs in and uses it for its (own) authorization check. If an application requests the ID card information, the relevant data is sent directly from OMNIS to the UTM application. The ID card must not be removed.

In the special case of a multiplex connection to a UTM application, OMNIS-MENU automatically passes the ID card information via *OPEN-SESSION* to the UTM application.

### **No ID card information was specified when the user was generated**

If an application requests the ID card, the information is entered via the ID card reader. If ID card information is again requested by an application after switching applications, the data is sent directly from OMNIS to the UTM application and it is not necessary to remove the ID card. Removal of the ID card in dialog with an application is forwarded by OMNIS-MENU (K14), causing the connection to the application to be cleared down.

## 7.5 Administrator messages

During a dialog with an application, messages from OMNIS (see the *MESSAGE* command on [page 234](#)) are displayed on screen line 25 or in the two message lines of the OMNIS-MENU masks (except the *Login* mask).

## 7.6 Use of XHCS

The character sets must be defined on the computer on which OMNIS-MENU is running and also on the partner computer.

If the XHCS functionality is also to be supported for parallel EXIT routines, these must be converted to VTSUCB.

---

## 8 Kerberos support via OMNIS-MENU

In modern, complex working environments, users often need access to multiple applications which may also be located on different servers. Consequently, they often have to use different user IDs and passwords. Different applications may also impose different rules these user IDs and passwords must comply with. In addition, it is often necessary to change different passwords at different intervals.

All this means more administration work. This affects not only users but also user administrators who have to reset forgotten passwords and re-enable user IDs that have been locked because the password has expired.

This increased administrative work can be avoided through the use of a Single Sign On system (SSO system). A SSO system is a system which permits automatic and convenient logon to network resources in heterogeneous networks. After a one-off identification and authentication an SSO system automates all subsequent logons by the user in the network.

Single Sign On with Kerberos is supported in BS2000 and SECOS (see the "SECOS" manual [10]).

### **Kerberos concept**

Kerberos is a standardized network authentication protocol which was developed at the Massachusetts Institute of Technology (MIT). It is a security system based on cryptographic encryption methods. For authentication with Kerberos, no passwords are sent over the network in plain text. This prevents passwords from being intercepted in the network.

The current version of Kerberos (V5) is standardized in RFC1510 (Request for Comments). The RFC standards themselves are defined by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). Comprehensive information on the RFCs is available on the home page of the IETF: <http://www.ietf.org>.

Kerberos works with symmetrical encryption. In other words: All keys are present at two locations, at the site of the key owner (principal) and at the KDC (Key Distribution Center). A key is derived from a principal's password.

### **Kerberos principal**

The Kerberos principal has a unique name, which can consist of any number of components. The components are separated from each other by the component separator '/'. The last component is the realm, which is separated from the other components by the realm separator '@'.

The name of an applications's principal comprises three components: application, instance and realm.

Typical format: Application/Instance@REALM where

Application is the 'host' for the application \$DIALOG or the name of the application

Instance is the DNS name of the server on which the application runs

REALM is the name of the Kerberos domain, written in upper case letters

Example of a typical Kerberos principal in BS2000:

```
host/bs2.de.ts.fujitsu.com@TS.FUJITSU.COM
```

### **Administering the keys in the key table**

In BS2000 the name of the principal must be added to the key table with the SECOS command /ADD-KEYTAB-ENTRY.

The secret keys on the BS2000 host are administered in the key table. An entry in the key table consists of the name of the BS2000 system as entered in the Key Distribution Center (KDC), and multiple keys, which are derived from the specified password and the system name using a cryptographical procedure. The password itself is not stored.



## 8.1 Precondition

### Precondition for the use of Kerberos

An existing KDC is a precondition, e.g. the Domain Controller (PDC) of Windows 2000 or the Windows 2003 Server, which supports this functionality.

The terminal emulation and the client operating system have to be equipped for the use of Kerberos.

The BS2000 server has to detect that the connection is prepared for Kerberos. For this purpose the client (e.g. the terminal emulation) has to log on as display station DSS9763 (Device type X'4F') at connection set-up.

### Terminal-Emulation MT9750

MT9750 supports BS2000 Kerberos authentication protocol and thus SSO.

### BS2000, SECOS and SECOS-KRB

In SECOS there is a rudimentary support of Kerberos authentication for \$DIALOG (TIAM). The SECOS-component SECOS-KRB comprises the interface for the management of Kerberos authentication in BS2000.

## 8.2 Access verification done by OMNIS-MENU

Access to OMNIS-MENU can be protected by a password and/or an ID reader. The user has to authenticate with a password or ID reader upon login or with a password and an ID reader.

Apart from this access protection, access protection is now possible by means of Kerberos. *Restriction:* access by ID reader and Kerberos cannot be combined.

If for a user a principal has been declared, the checking of the access protection with Kerberos is the first check that will be made.

If, apart from Kerberos, also a password for the user has been defined, also the password will be checked after the Kerberos check.

## 8.3 Checking of the Kerberos principal

OMNIS-MENU performs the checking of the principal upon the login of a user. For this purpose

- the principal has to be declared in the OMNIS-MENU by means of OMNISKD (OMNISMD).
- the Kerberos ticket has to be requested from the client
- the ticket has to be checked.

## 8.4 Process of an authentication

Authentication is performed in the following steps when an OMNIS-MENU connection is started:

1. The user of a terminal emulation opens an OMNIS-MENU session.
2. After connection set up OMNIS-MENU sends the mask "Login" to the emulation.
3. The user completes the entry fields "user", "password" (if applicable) and further fields and sends the mask to OMNIS-MENU.
4. OMNIS-MENU recognizes the access protection for the user by means of the Kerberos principal. The following activities are then executed invisibly for the user:
  - a) OMNIS-MENU sends a Kerberos ticket request (generated by SECOS) to the terminal emulation.
  - b) The terminal emulation will receive a ticket from the KDC and sends it to OMNIS-MENU.
  - c) OMNIS-MENU invokes SECOS, where the ticket is validated by deciphering.
  - d) OMNIS-MENU then checks whether the ticket owner identified as Kerberos principal has been entered in the DECLARE-USER statement of the respective user. Depending on the result of this check, access is permitted or denied.
5. If access is permitted, the user gets the menu defined in the application selection. Upon denying, the Login mask is output with an error message and the return code.

OMNIS(-MENU) does not store the ticket.

OMNIS-MENU and \$DIALOG perform their Kerberos check independently of each other. Even if OMNIS-MENU has defined the Kerberos authentication and has already requested a ticket, the \$DIALOG partners request their own tickets. OMNIS takes no notice of these communications.

## 8.5 Declaration of the principal in OMNIS-MENU

### Key administration

When using \$DIALOG it is generally not required to administer a key for each Kerberos user. This applies for OMNIS-MENU as well.

For this reason an OMNIS-MENU user does not need an entry in the key table of BS2000, as the entries in the key table of BS2000 are related to the applications and not to their users.

### OMNISKD

The user management of OMNIS-MENU contains an operand in the OMNISKD statement DECLARE-USER which enables an OMNIS-MENU user to be assigned the relevant principal.

In this case a user can only be reached via a single principal. If it should be possible to reach OMNIS-MENU via various principals, multiple users must be declared. However, they can use the same menu.

The principals are stored in the user table of OMNIS-MENU (in encrypted form).

### OMNISMD

In the "Define User" mask the generation program OMNISMD contains an entry field with which the relevant principal can be assigned to an OMNIS-MENU user.

### OMNISFC

OMNISFC takes principals into account when conversion takes place.

## 8.6 Connection clear down, connection loss

A connection clear down and/or loss of connection to a terminal may occur in various situations. The actions of OMNIS-MENU following a connection clear down and or a connection loss can be determined by the user and/or administrator.

If the user is in the masks "Application Menu" or "Login", he can log out of OMNIS-MENU by pressing the logout key.

The DISCONNECTION-MASK operand in the DECLARE-PARAMETER statement enables the administrator to define whether the logout mask of OMNIS-MENU is displayed. After logging out from OMNIS-MENU, this provides the user with the opportunity of performing follow-up actions using the K keys:

- K1 terminates OMNIS-MENU and clears down all partner connections which are still open
- K2 terminates OMNIS-MENU, but holds all partner connections which are still open so that they can be resumed automatically when reconnection takes place. This function is available only if the other conditions described below are fulfilled.
- K3 returns to OMNIS-MENU (in the case of erroneous logout)

The LOGOUT-HOLD-AVAILABILITY operand in the DECLARE-USER statement enables the administrator to define whether the K2 key for the user-specific holding of the partner connection is available to a particular user in the logout mask.

With the operand ONE-USER-ONE-CONNECTION of the DECLARE-PARAMETER command, the administrator can determine whether a user may connect only once at the same time to OMNIS-MENU.

The user-specific holding of the partner connection is possible upon connection clear down or connection loss only if the global parameter ONE-USER-ONE-CONNECTION=YES has been set. Otherwise OMNIS-MENU informs with the message OMM0034.

All connections will be closed down during the process.

The user-specific holding of Kerberos-authenticated partner connections is only permitted for Kerberos-authenticated users. OMNIS-MENU and the partner of the connection determine the identity of the user unambiguously and independently of each other by means of Kerberos authentication. After a renewed login, OMNIS-MENU checks the user's identity, thus ensuring that only that user can take over the Kerberos-authenticated partner connections and that the authentication of the partner retains its validity.

When *SECURITY LEVEL*≠*LOW*, Kerberos-authenticated partner connections do not switch to the HOLD status, but are always cleared down when the connection to the terminal is lost.

## 8.7 Connection take-over

### User-specific connection take-over

An OMNIS-MENU application (OMNIS partner) is always assigned to exactly one user (OMNIS terminal). OMNIS-MENU users cannot take over applications of another user. OMNIS-MENU users can only perform a user-specific holding and a user-specific take-over (OCCUPY) if the user of the applications has logged on once again.

### Connection take-over of Kerberos-authenticated user

The OMNIS administrator can change settings by means of the OMNIS-command OCCUPY, i.e. the administrator is in a position to take over partners from the terminal of another user. When using the command OCCUPY of partners, OMNIS does not verify whether their terminal is Kerberos-authenticated. The terminals and/or partners can be taken over by the OMNIS-administrator and the APASS (Administrator Password).

### Connection take-over of Kerberos-authenticated TIAM-partner

Kerberos-authenticated partners (currently only TIAM) cannot be taken over by OMNIS administrator.

Kerberos-authenticated users can take over their Kerberos-authenticated partner connections when the OMNIS start parameter *SECURITY LEVEL=LOW*. Otherwise connections to partners authenticated by Kerberos (formerly by a chip card) are cleared down upon loss of connection towards the data station.



---

## 9 OMNIS commands

OMNIS commands are used to control the functionality of OMNIS. They are entered at a terminal, but instead of being passed on to a partner they are processed by OMNIS itself.

This section describes the OMNIS commands, starting with an overview of the available commands. This is followed by an explanation of the command scopes, which provides information about the various operand settings. The syntax of the command descriptions is then explained, followed by the actual descriptions of the OMNIS commands.

### 9.1 Overview of the OMNIS commands

The following commands are implemented in OMNIS (A = administrator command):

CANCEL	(A)	Terminate connection
CHANGELOG	(A)	Switch logging file
CLSCON		Close connection to partner
CMD	(A)	Execute BS2000 command
CONFUPDATE	(A)	Change the configuration while OMNIS is running
DUMP	(A)	Output dump
END	(A)	Terminate OMNIS
EXIT		Administer exit routines
FILE		Process command file
GROUP		Define partner group
HALT		Sign off terminal from OMNIS
HCOPY		Enter hardcopy unit in printer table
HELP		Assistance with commands and messages
INF		Output information
MDEF		Define message table
MESSAGE		Send message to a terminal
OCCUPY		Take over existing connection
OPNCON		Open connection to a partner

OPTION		Define terminal-specific attributes
PARAM		Define partner-specific attributes
PRIORITY	(A)	Define scope of commands
RESTART		Restart OMNIS-OMNIS connection/printer connection
SET	(A)	Set operating parameters
TRACE	(A)	Switch on/off trace
TURN		Page in dialog logging file

## 9.2 Command scope

The commands *SET*, *OPTION*, *PARAM* and *OPNCON* and the OMNISKD statements *DECLARE-TERMINAL* and *DECLARE-PARTNER* (see the “Administration and Programming“ manual [1]) have, for the most part, identical operands. They differ primarily in their scope.

### Global scope

The *SET* command has global scope, i.e. operands defined with this command affect all OMNIS users.

### Terminal-specific scope

The scope of *DECLARE-TERMINAL* and *OPTION* applies to one terminal only, i.e. operands defined with these commands affect one user or all partners of one user.

The difference between *DECLARE-TERMINAL* and *OPTION* lies only in the time at which the operands are defined (upon declaration or after connection setup). The last value specified applies.

### Partner-specific scope

The scope of *DECLARE-PARTNER*, *OPNCON* and *PARAM* applies to one partner only, i.e. operands defined with these commands affect only a single partner.

The difference between *DECLARE-PARTNER*, *OPNCON* and *PARAM* lies only in the time at which the operands are defined (upon declaration, when a connection is being established, or after connection setup). The last value specified applies.

### User-specific scope

When OMNIS-MENU is used, user names are also permitted with the commands *OCCUPY*, *INF*, *CANCEL* and *MESSAGE*.



The following figure shows the various command scopes:

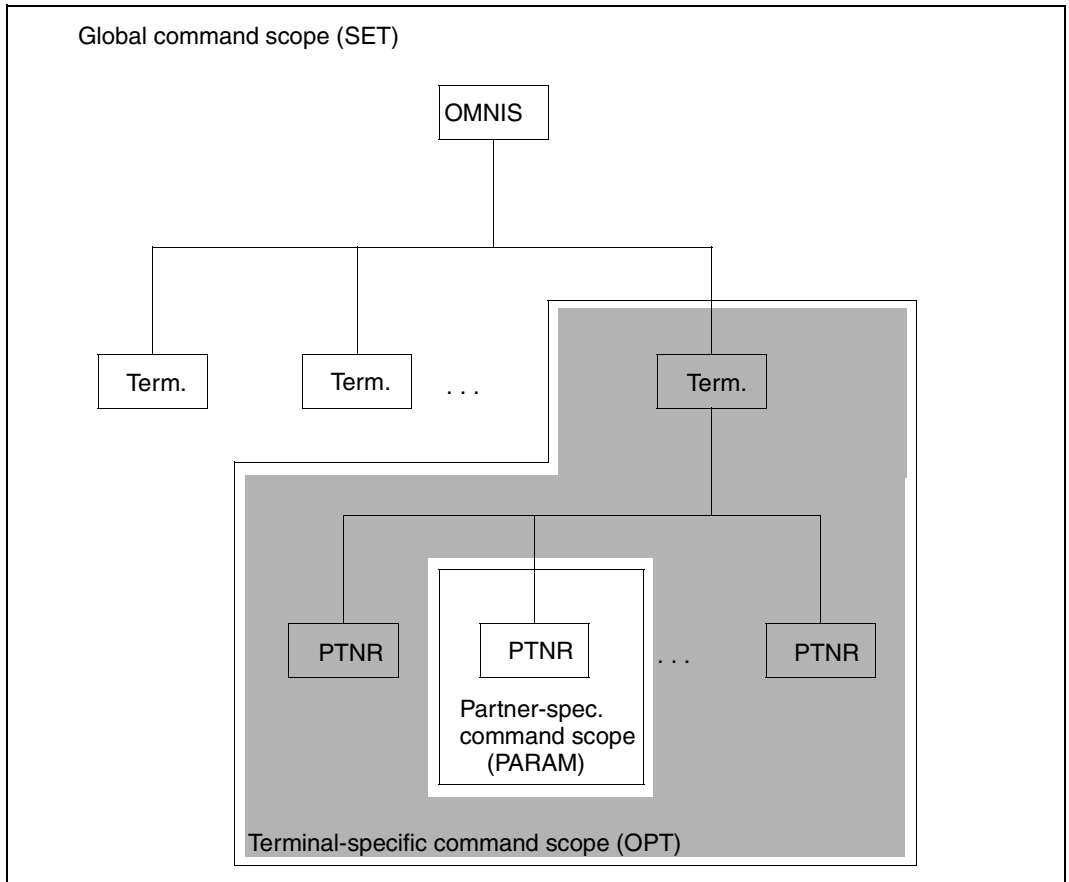


Figure 35: Command scopes

### Command scope hierarchy

The command scope hierarchy determines the actual operand settings at a given time.

As the default option, partner-specific commands have the highest priority, followed by terminal-specific commands and, lastly, globally valid commands.

For example, the two commands *OPTION LOGGING=YES* and *PARAM pac1, LOGGING=NO* have the effect that all the messages for a terminal, except those from the partner with address code *pac1*, are logged (regardless of what is specified in the *SET* command).

If the special value *STD* (i.e. “undefined”) is specified for an operand, a lower-priority command then becomes valid.

For example, *PARAM pac1, LOGGING=STD* cancels the exceptional status of partner *pac1* defined in the previous example; the value defined in the *OPTION* command applies again (note the difference between this and *PARAM pac, LOGGING=NO*).

The *SET* command defines default values for operands which apply if *STD* is specified in all scopes for these operands.

For example, the following command combination sets the default for the *LOGGING* operand in the command *SET* to *LOGGING=NO*.

```
PARAM/OPNCON/DECLARE-PARTNER...LOGGING=STD,
OPTION/DECLARE-TERMINAL...LOGGING=STD,
SET...LOGGING=STD
```

### Modifying the scope

The administrator is permitted to modify the priority of the various command scopes with respect to the command operands *EXIT*, *HOLD=AUTO* and *LOGGING*.

If *PRIORITY LOGGING-PRIORITY=SET-PAR-OPT* is specified, the priority *SET* before *PARAM* before *OPTION* applies. The value for *LOGGING* selected with *SET* for all terminals and partners thus has the highest priority.

The command *PRIORITY LOGGING-PRIORITY=PAR-OPT-SET* can be used to switch back to the default priority *PARAM* before *OPTION* before *SET*.


If *PRIORITY HOLD-PRIORITY=SET-OPT* is specified, the *HOLD=AUTO* operand defined with the *SET* command applies. If *PRIORITY HOLD-PRIORITY=OPT-SET* is specified, the *HOLD=AUTO* value specified for the terminal applies.

If *PRIORITY EXIT-PRIORITY=SET-PAR-OPT* is specified, the setting *SET* before *PARAM* before *OPTION* applies. With *PRIORITY EXIT-PRIORITY=PAR-OPT-SET*, the setting *PARAM* before *OPTION* before *SET* applies.

With *PRIORITY...SET-PAR-OPT*, the administrator can thus suppress various functions selected by the users: logging (of partner inputs and outputs), the use of exit routines and the automatic takeover of connections to active partners after a loss of connection (*HOLD=AUTO*).

## 9.3 Syntax description

### Explanation of the metasyntax of OMNIS commands

Notation	Explanation	Example
UPPERCASE LETTERS	Uppercase letters indicate constants which must be entered exactly as shown.	DISPLAY=YES Enter the following: DISPLAY=YES
lowercase letters	Lowercase letters indicate variables which the user must replace with actual values.	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> .   The commands <i>OPTION</i> , <i>PARAM</i> , <i>PRIORITY</i> , <i>SET</i> and <i>TRACE</i> are subject to specific default value rules. Please refer to the corresponding notes under these commands.	DISPLAY={ <u>NO</u> }  Specify for example: DISPLAY=NO or DISPLAY=YES. If DISPLAY is not specified, DISPLAY=NO applies.
[ ]	Brackets enclose entries which may be omitted.	O[PNCON]  Specify for example: OPNCON, OPN or O.

## General command format

The functions of OMNIS are controlled by commands. Commands to OMNIS can be entered

- explicitly using the OMNIS address code `@` (to the local OMNIS), or an OMNIS address code in the form `oac` (to an OMNIS connected to the local OMNIS via an OMNIS-OMNIS connection)
- implicitly by using a K key (to the local OMNIS).

An explicit command input has the same format as a message to a partner with address code `@` or `oac`, and is subject to the general rules for partner address codes; therefore, for sequences of commands, the partner address code need only be entered with the first command.

An explicit command generally uses the following syntax:

```
[@:]cmd [operand-list]
[oac:]cmd [operand-list]
```

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

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

Comments are user-defined character strings enclosed in double quotes, e.g. "This is a comment".

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

If an OMNIS command cannot be analyzed, the string is output on the terminal with the position which cannot be processed highlighted.

### Examples of valid commands

```
@:MDEF X,MSG=AAAAAAA,D=N
@:  MDEF    X  , MSG = AAAAAAA,    D =N
@:MDEF X, MSG=AAAAAAA, D=N      "comment"
@:  MDEF"comment"X"comment"    ,MSG="comment"AAAAAAA,D=N
```

### Examples of invalid commands

```
@:MDEF X,MSG=AAAA AAA,D=N
@: MD"comment"EF X,MSG=AAAAAAA,D=N
```

**Syntax of the main 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 a variable concept means that variables can be used in some commands. This applies to the following commands:

<b>Command</b>	<b>Parameter</b>
MDEF	REPLY
OPNCON	PRONAME PTNNAME
OPTION	HCOPY(DEVICE,FORM)
SET	HCOPY(DEVICE,FORM)

You will find details on the variable substitution concept in the "Administration and Programming" manual [\[1\]](#)

## 9.4 Descriptions of the individual commands

### CANCEL - Terminate connection

The *CANCEL* command terminates specific connections, including “foreign” connections (those of another OMNIS terminal).




The administrator can use the *CANCEL* command without restriction.

You can use *CANCEL* to terminate foreign connections if the terminal has the requisite authorization. A uniquely identified user can terminate all his or her own connections with *CANCEL*.

	Operation	Operands
Format 1	CAN[CEL]	$\left\{ \begin{array}{l} \text{PID} = \text{num} \\ \text{PAC} = \text{pac} \left\{ \begin{array}{l} [, \text{TID} = \left\{ \begin{array}{l} \text{OWN} \\ \text{num} \end{array} \right\} ] \\ [, \text{T[ERMINAL]} = ([\text{ptn}][, \text{pro}])] \\ [, \text{U[SER]} = \text{user}] \end{array} \right\} \end{array} \right\}$ $[, \text{AP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} ]$ [,KILL]
Format 2	CAN[CEL]	$\left\{ \begin{array}{l} \text{TID} = \text{num} \\ \text{T[ERMINAL]} = ([\text{ptn}][, \text{pro}]][, \text{U[SER]} = \text{user}] \end{array} \right\}$ $[, \text{AP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} ]$ [,KILL]

	Operation	Operands
Format 3	CAN[CEL]	$\left. \begin{array}{l} \text{HID=num} \\ \text{HC[OPY]=hac} \end{array} \right\}$ $[,\text{AP}[\text{ASS}]=\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}]$ $[,\text{F}[\text{AILURE}]=\left. \begin{array}{l} *[\text{S}[\text{AVE}] \\ *[\text{I}[\text{GNORE}] \\ \text{hac} \end{array} \right\}]$ $[,\text{KILL}]$
Format 4	CAN[CEL]	$\left. \begin{array}{l} (\text{MUX} = (\text{ptnname}, \text{praname}) \\ \text{MID} = \text{mid} \end{array} \right\}$ $[,\text{AP}[\text{ASS}]=\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}]$ $[,\text{KILL}]$
Format 5	CAN[CEL]	$\left. \begin{array}{l} (\text{OAC} = \text{oac} \\ \text{OID} = \text{num} \end{array} \right\}$ $[,\text{AP}[\text{ASS}]=\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}]$ $[,\text{KILL}]$



PID	is the PID of the partner to which the connection is to be terminated. The “owner“ of the partner is notified with the message OMS0087.
PAC	is the PAC of the partner of the selected terminal, to which the connection is to be cleared down.
TID	(Format 1) acts in combination with the PAC argument to define a specific partner to which the connection is to be cleared down.
TID	(Format 2) is the TID of the terminal to which the connection is to be terminated. The terminal's operator is informed by the message OMS0086. If the terminal is defined with <i>HOLD=YES/AUTO</i> , the connections to the partners of the terminal are not terminated, and active partners remain active.
OWN	You select connections of your own terminal.
num	You select connections of the terminal having the TID <i>num</i> .
TERMINAL	is the network address of the terminal whose connections are to be selected.
ptnr	is the partner name of the terminal.
pro	is the processor name of the terminal.
USER	is the user whose connections are to be selected.
	If multiple connections meet the search criteria you specify, OMNIS rejects the <i>CANCEL</i> command and issues an <i>INF TERMINAL</i> command to enable you to repeat the <i>CANCEL</i> command with the unique TID.
user	is the name of the user.
APASS	specifies the valid administration password. This operand is mandatory if the terminal is not defined as the administrator (by specifying <i>ADM=YES</i> in the <i>OPTION</i> or <i>DECLARE-TERMINAL</i> command.) Since, if <i>SECURITY LEVEL=HIGH</i> is set, the <i>APASS</i> operand is rejected with the message OMS0190, the <i>CANCEL</i> command can be entered only at the BS2000 console in this case.
passwd	is the administration password.
?	The administration password is to be requested by OMNIS in blanking mode after input of the command.
KILL	You can specify <i>KILL</i> in order to clear down inactive connections as well (e.g. connections in the <i>OPN</i> state).

HID	<p>is the HID of the printer to which the connection is to be terminated. Termination of the connection has the same effect as a connection failure (i.e. the alternate printer handling defined in the declaration or <i>HCOPY</i> command is performed) unless different handling is defined with <i>FAILURE</i>.</p> <p>After input of the command, the messages OMS0088 (HC connection cancelled) and OMS0103 (no alternate device) are displayed if no alternate printer is defined, or OMS0102 (alternate unit assigned) is displayed if an alternate printer is defined.</p>
HCOPY	<p>Specifies the printer to which the connection is to be terminated.</p>
hac	<p>is the hardcopy address code of the printer to which the connection is to be terminated. Termination of the connection has the same effect as a connection failure unless different handling is defined with <i>FAILURE</i>.</p> <p>After input of the command, the messages OMS0088 (HC connection cancelled) and OMS0103 (no alternate device) are displayed if no alternate printer is defined, or OMS0102 (alternate unit assigned) is displayed if an alternate printer is defined.</p>
FAILURE	<p>redefines the alternate printer handling defined in the printer declaration. The <i>FAILURE</i> setting is valid until the next <i>RESTART</i> command.</p> <p>After <i>RESTART</i>, the alternate printer defined with <i>FAILURE</i> is activated again. The messages OMS0095 (restart initiated), OMS0097 (restart successful) and OMS0099 (hardcopy available again) are output:</p>
*SAVE	<p>Messages destined for the printer are to be written to the logging file and output on the printer when it is available again (<i>RESTART</i> command).</p>
*IGNORE	<p>Messages destined for the printer are neither to be output nor to be written into the logging file, which means that they will be lost.</p>
hac	<p>Messages destined for the printer are to be output on the alternate printer <i>hac</i>.</p>

When the *CANCEL MUX/MID* command is executed, all sessions affected by the associated connection are lost:

MUX	<p>You specify the multiplex connection to which you are applying the <i>CANCEL</i> command.</p>
ptnr-name	<p>is the name of the UTM application to which the transport connection is to be terminated.</p>
praname	<p>is the processor of the UTM application to which the transport connection is to be terminated.</p>

**MID**                    With *mid* you specify the identification number of the transport connection which is to be cleared down.

Once an OMNIS-OMNIS connection has been cleared down with *CANCEL*, all users of the associated indirect connection are sent a message informing them that they can no longer work with the partner. The command does, however, only clear down the OMNIS-OMNIS connection itself, so you can still use *OCCUPY* to take over connections of associated terminals. You can also still use *INF*.

**OAC**                    *oac* is the OMNIS address code of the OMNIS-OMNIS connection that is to be cleared down.

**OID**                    *num* is the identification number of the OMNIS-OMNIS connection that is to be cleared down.

### *Examples*

@: /CAN TID=20

@: OMS0086 TERMINAL CONNECTION CANCELLED BY OMNIS ADMINISTRATOR

@: /CAN PID=35

@: OMS0087 CONNECTION TO PARTNER "A (PTNNAME/PRONAME)" CANCELLED BY OMNIS ADMINISTRATOR

@: /CAN HID=17,FAILURE=\*SAVE

@: /CAN PID=A,AP=?

@: OMS0137 PLEASE ENTER APASS:

@: ?                    (blanked)

@: OMS0087 CONNECTION TO PARTNER 'A (\$DIALOG/G133H207)' CANCELLED BY OMNIS ADMINISTRATOR

## CHANGELOG - Manage logging file

The *CHANGELOG* command offers the following functions to manage the logging file during ongoing operation:

- Synchronous switching to a new logging file (ACTION=CHANGE)  
You can specify whether the old logging file should be deleted.
- Synchronous creation of a copy of the current logging file (ACTION=COPY)  
A copy of the current logging file is created with a freely-selectable name.

The synchronous functions are executed when a *CHANGELOG* command which was entered explicitly is processed.

- Asynchronous, time-driven switching to a new logging file (REPEAT=)  
A single or regularly recurring time at which the logging file is switched can be specified.
- Asynchronous, size-driven switching to a new logging file (SIZELIMIT=)  
A maximum size can be specified for the logging file. The switchover is executed automatically when the logging file reaches the specified size.

The asynchronous functions are initiated automatically by OMNIS when one of the defined events occurs.



In ongoing operation, the *CHANGELOG* command is reserved for use by the administrator. However, it can also be specified in the startup file with at least one of the operands SIZELIMIT or REPEAT.

Operation	Operanden
CH[ANGELOG]	<p>[AC[TION]=<math>\left. \begin{array}{l} \text{CH[ANGE]} \\ \text{CO[PY]} \end{array} \right\} ]</math></p> <p>[AP[ASS]=<math>\left. \begin{array}{l} \text{'password'} \\ ? \end{array} \right\} ]</math></p> <p>[D[ELETE]=<math>\left. \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ]</math></p> <p>[N[AME]=<math>\left. \begin{array}{l} \text{STD} \\ \text{&lt;filename 1..22&gt;} \end{array} \right\} ]</math></p> <p>[S[IZELIMIT]=<math>\left. \begin{array}{l} \text{&lt;int 96..262144&gt;} \\ \text{R[ESET]} \\ \text{U[NCHANGED]} \end{array} \right\} ]</math></p> <p>[R[EPEAT]=(*)]</p> <p>(*) <math>\left. \begin{array}{l} \left. \begin{array}{l} \text{[N[O]} \\ \text{D[ATE]} \end{array} \right. = \left. \begin{array}{l} \text{mm:dd} \\ \text{TODAY} \end{array} \right\} \text{E]=} \left. \begin{array}{l} \text{hh:mm} \\ \text{STD} \end{array} \right\} \\ \\ \left. \begin{array}{l} \text{[P[ERIOD]} \\ \text{H[OURS]} \end{array} \right. = \left. \begin{array}{l} \text{&lt;int 1..24&gt;} \\ \text{STD} \end{array} \right\} \text{STD} \left. \begin{array}{l} \text{hh:mm} \\ \text{STD} \end{array} \right\} \\ \\ \left. \begin{array}{l} \text{[D[AILY]} \\ \text{[W[EELK]} \\ \text{[W[DAY]} \end{array} \right. \left. \begin{array}{l} \text{(T[IME]=} \\ \text{&lt;int 1..7&gt;} \\ \text{STD} \end{array} \right\} \left. \begin{array}{l} \text{hh:mm} \\ \text{STD} \end{array} \right\} \\ \\ \left. \begin{array}{l} \text{[M[ONTH]} \\ \text{M[DAY]} \end{array} \right. = \left. \begin{array}{l} \text{&lt;int 1..31&gt;} \\ \text{--} \end{array} \right\} \left. \begin{array}{l} \text{hh:mm} \\ \text{STD} \end{array} \right\} \end{array} \right\}</math></p>

ACTION	specifies the action that shall be performed.
CHANGE	the old logging file is closed and is thus available for offline analysis. A new logging file is opened.
COPY	a copy of the current logging file is created, the old logging file is still in use.
APASS	specifies the valid administration password. This operand is mandatory if the terminal is not defined as the administrator (by specifying <i>ADM=YES</i> in the <i>OPTION</i> or <i>DECLARE-TERMINAL</i> command.) If <i>SECURITY LEVEL=HIGH</i> is set, the <i>APASS</i> operand is rejected with the message OMS0190.  The <i>CHANGELOG</i> command can, therefore, be entered only at the BS2000 console in this case.
passwd	is the administration password.
?	The administration password is to be requested by OMNIS in blanking mode after input of the command.
DELETE	specifies whether or not the old logging file shall be deleted. The parameter must only be stated, if <i>ACTION=CHANGE</i> has been specified.
NO	the old logging file will not be deleted.
YES	the old logging file will be deleted.
NAME	specifies the prefix of the name of the new logging file. The parameter must only be stated, if <i>ACTION=COPY</i> has been specified.
STD	the name of the copy will be derived from the name of the current logging file. A timestamp, taking the format of <yyyy-mm-dd>.<hhmm>, is appended to the name of the current logging file. In this case the specified name of the logging file (link name SOMSLOG, see the "Administration and Programming" manual [1]) must not be longer than 18 characters, since a counter, taking the format of <.nnn>, is appended to this name additionally. If the name used so far is too long, an error message will be output and the <i>NAME=&lt;filename 1..22&gt;</i> operand must be used.

<filename 1..22>	<p>if a copy is created, a timestamp taking the format &lt;yyyy-mm-dd&gt;.&lt;hhmm&gt; is appended to the specified name, indicating the file and the time of creation.</p> <p>For this reason the name specified here can have a maximum of 22 characters.</p> <p>The correctness of the file type is assumed and will not be verified. In case of an error no copy will be created.</p>
SIZELIMIT	<p>specifies the maximum size of the logging file in PAM pages. The parameter must only be stated, if ACTION=CHANGE has been specified.</p>
<int 96..262144>	<p>if this file size is reached, a change in the logging file will automatically be performed.</p>
RESET	<p>the previously valid setting for asynchronous, size-driven execution of the function is reset, i.e. the size of the logging file is not monitored.</p>
UNCHANGED	<p>the previously valid setting for asynchronous, size-driven execution of the function is retained. If no <i>CHANGELOG</i> command with the SIZELIMIT operand has yet been entered, no size-controlled switching of the logging file takes place.</p>
REPEAT	<p>specifies the time or period in which the logging file is to be changed (ACTION=CHANGE) or in which a copy (ACTION=COPY) is to be created.</p>
NO	<p>the function is executed only once when the specified time is reached.</p>
TIME	<p>specifies the time in the format &lt;hh:mm&gt; (&lt;00:00&gt;..&lt;23:59&gt;) or STD (00:00 h).</p>
DATE	<p>specifies the date in the format &lt;mm:dd&gt; (&lt;1..12&gt;:&lt;1..31&gt;) or TODAY (the current date).</p>

PERIOD	the function is executed periodically. TIME determines the start time, HOUR the length of the time interval. The function is executed from the next possible interval time, see <a href="#">“Examples” on page 188</a> .
TIME	specifies the time in the format <hh:mm> (<00:00>..<>23:59>) or STD (00:00 h).
HOURS	specifies the time interval in hours (1..24) or STD (1 hour).
DAILY	the function is executed daily at the specified time.
TIME	specifies the time in the format <hh:mm> (<00:00>..<>23:59>) or STD (00:00 h).
WEEKLY	the function is executed weekly at the specified time.
TIME	specifies the time in the format <hh:mm> (<00:00>..<>23:59>) or (00:00 h).
WDAY	specifies the weekday (Monday = 1, Tuesday = 2, ..., Sunday = 7) or STD (Monday =1).
MONTHLY	the function is executed monthly at the specified time.
TIME	specifies the time in the format <hh:mm> (<00:00>..<>23:59>) or (00:00 h).
MDAY	specifies the day in the month or STD (1, the first day of the month).
RESET	the previously valid setting for asynchronous, time-driven execution of the function is reset, i.e. no time monitoring takes place for the logging file.
UNCHANGED	the previously valid setting for asynchronous, time-driven execution of the function is retained. If no <i>CHANGELOG</i> command with the REPEAT operand has yet been entered, no time-controlled switching of the logging file takes place.



How to form the name for the new logging file is described in the “Administration and Programming” manual [1].

OMNIS can no longer access data contained in the old logging file. This data is lost for current dialog operations, but is still available to the administrator during analysis of the old logging file.

If a memory space reservation is not explicitly requested with the `SPACE` operand when a new logging file is opened, the values of the last successfully opened logging file are used. If, for whatever reason, a new logging file cannot be opened, the `CHANGELOG` command is not executed.

The time-controlled asynchronous change of the logging file only functions when the `RTIMER` has been activated (see command `SET`). If the `RTIMER` is deactivated and subsequently activated again, asynchronous changes of the logging file, which would have been executed during the time the `RTIMER` has been deactivated, will be lost. They will not be executed at a later time.

#### Notes

- If `NONE` of the new operands is specified, the behavior is the same as in OMNIS up to V8.3A, i.e. specification of `CHANGELOG [APASS=]` leads to synchronous switching of the logging file. The old logging file is retained. The default name is used for the new logging file.  
Existing settings for asynchronous switching of the logging file are retained.
- If the operands `SIZELIMIT` or `REPEAT` are specified, the `CHANGELOG` command is executed synchronously.  
Existing settings for asynchronous switching of the logging file are retained.
- The values of the `SIZELIMIT` and `REPEAT` operands determine the time when the `CHANGELOG` function is executed asynchronously.  
If only one of the two operands is specified, this value is changed accordingly; the value not specified remains unchanged.
- If `SIZELIMIT` has been specified, the change of the logging file is performed, if the logging file has reached the specified file size (in PAM pages).
- If `REPEAT` has been specified, the `CHANGELOG` function is executed if the specified time or the end of the specified time period has been reached.
- If one of the conditions `SIZELIMIT` or `REPEAT` is fulfilled, the `CHANGELOG` function is executed.
- If for `MDAY` the values 29, 30, 31 are specified, the last day of a month is assumed automatically for months not including this date (e.g. June 31).

- If for DATE the values 29, 30, 31 are specified, an error message will be output for months not including this date (e.g. June 31) and *CHANGELOG* will be executed using REPEAT=RESET.
- If REPEAT=NO the date is specified in the format <mm-dd> without specifying a year. Thus the next 12 months can be covered. If the date of the previous day is specified, the respective date in the next year is set.  
However, this logic does not apply for the current date: If the current date has been specified as date (explicitly or by means of the default value), the specified day of the current year will be assumed in any case. If the specified time has already lapsed, the respective time is NOT assumed as to mean a time in almost one year, but rather an error message is output and *CHANGELOG* will be executed using REPEAT=RESET.
- For REPEAT=NO one of the two operands DATE or TIME has to be specified. If both for DATE the default value TODAY and for TIME the default value <00:00> is used, this time is already over, the respective OMNIS message will be output and *CHANGELOG* will be executed using REPEAT=RESET.
- REPEAT=PERIOD defined periodic times for executing the specified function.  
TIME=09:00, HOURS=4 defines, for example, the times 01:00; 05:00; 09:00; 13:00; 17:00 and 21:00. The function is executed at the next defined time after the time at which the command was entered and then at the other defined times, see [“Examples” on page 188](#).
- The value specifications of a new *CHANGELOG* command overwrites the settings valid so far, i.e. a new SIZELIMIT-specification overwrites the existing SIZELIMIT-specification.
- If SIZELIMIT or time (periods) (REPEAT) has not been specified, the previous settings will be maintained.  
If none of the two parameters have been specified, a synchronous change of the logging file is performed.  
If one of the two parameters has been specified, these values will be changed accordingly, and the value that has not been specified will remain unchanged.
- The parameter value UNCHANGED can be specified for SIZELIMIT and REPEAT, so that these settings remain unchanged.  
This specification only makes sense, if only the setting for DELETE, NAME or ACTION for the asynchronous change of the logging file shall be changed (i.e. a SIZELIMIT or a REPEAT has to be specified), however the sizes and the time setting should remain unchanged.
- If RESET for SIZELIMIT or REPEAT has been specified, the previous settings for an asynchronous change of the logging file for SIZELIMIT or REPEAT become ineffective.

- A file-size-controlled change of the logging file always has to be executed with a change of the logging file. In this case, it does not make sense to just create a copy, since the file would keep on growing and thus the SIZELIMIT would be exceeded again.  
If ACTION=COPY and SIZELIMIT=nnn have been explicitly specified, an OMNIS message will be output.  
If ACTION=COPY has been explicitly specified and SIZELIMIT has been set by an earlier *CHANGELOG*-command (with ACTION=CHANGE), an OMNIS message will be output.  
If SIZELIMIT has been explicitly specified and ACTION=COPY has been set by an earlier *CHANGELOG*-command (with REPEAT parameter) an OMNIS message will be output.
- If DELETE=YES has been specified, the old logging file will be deleted. This specification is permitted only upon ACTION=CHANGE.  
If ACTION=COPY and DELETE=YES has been explicitly specified, an OMNIS message will be output.  
If ACTION=COPY has been explicitly specified and DELETE=YES from an earlier *CHANGELOG* command setting, an OMNIS message will be output.  
If DELETE=YES has been explicitly specified and ACTION=COPY from an earlier *CHANGELOG* command setting, an OMNIS message will be output.
- The specifications for DELETE and NAME in an asynchronously executed *CHANGELOG* command are recorded and will apply for all the following executions of an asynchronous change of the logging file.
- The indications for DELETE and NAME of a synchronous change of the logging file only apply for the synchronous change of the logging file and do not change any settings for an asynchronous change of the logging file.
- The *CHANGELOG* command can also be specified in the startup file. In this case at least one of the operands SIZELIMIT or REPEAT must be specified, otherwise the command will be rejected.

*Examples*

1. One-off change of the logging file:

```
@:/CMD SHOW-FILE-ATTRIBUTES FILE-NAME=OMNIS.LOGGING.
@:      96 :A:$OMNIS.OMNIS.LOGGING.001
@::A:   PUBLIC: .....
```

```
@:/CHANGELOG APASS='HUGO'
```

```
@:/CMD SHOW-FILE-ATTRIBUTES FILE-NAME=OMNIS.LOGGING.
@:      96 :A:$OMNIS.OMNIS.LOGGING.001
@:      96 :A:$OMNIS.OMNIS.LOGGING.002
@::A:   PUBLIC: .....
```

2. Weekly change of the logging file (on Mondays at 00:00):

```
@:/CHANGELOG ACTION=CHANGE,REPEAT=WEEKLY(TIME=STD,WDAY=1)
```

3. Size-driven change of the logging file (when 1000 PAM pages are reached):

```
@:/CHANGELOG ACTION=CHANGE,SIZELIMIT=1000
```

4. Periodic creation of a copy of the logging file (every 4 hours):

```
@:/CHANGELOG ACTION=COPY,REPEAT=PERIOD(TIME=09:00,HOURS=4)
```

This command defines the following times for creating the copy:

01:00; 05:00; 09:00; 13:00; 17:00; 21:00.

If the command is entered at 11:00, the logging file is changed for the first time at 13:00, then at 17:00, and so on.

## CLSCON - Close connection to partner

The *CLSCON* command is used to clear down a connection to one or more partners.

Operation	Operands
C[ <i>LSCON</i> ]	<div style="display: flex; align-items: center;"> <div style="font-size: 3em; margin-right: 10px;">}</div> <div style="display: flex; flex-direction: column; gap: 5px;"> <span>pac</span> <span>gac</span> <span>*C</span> <span>*D</span> <span>*T</span> <span>*U</span> <span>*</span> </div> </div> [,KILL]

- pac** is the address code of the partner to which the connection is to be cleared down.
- gac** is the address code of a partner group. All connections to the members of the group are cleared down.
- \*C,\*D, \*T,\*U,\*** are the address codes of the standard partner groups. All connections to UCON partners (\*C), DCAM partners (\*D), TIAM partners (\*T), UTM partners (\*U) or all partners (\*) are to be cleared.
- KILL** By specifying *KILL* you can also clear down inactive connections (such as connections in the *OPN* state).

For the partner, this results in loss of the connection (see [page 53](#)).

Undeclared partners are no longer known to OMNIS after the connection has been closed (they no longer appear in the output of the *INF* command). The address codes assigned to such partners may be re-used.

Declared partners are set to 'inactive', but remain in the system. The connection can be reestablished with the *OPNCON* command.

In the case of direct connections, the entire *CLSCON* command is executed immediately and the message OMS0054 is output.

For indirect connections, the *CLSCON* command is executed in two parts:  
 The master passes the command on to the slave (message OMS0054).  
 Command processing is not complete, however, until the slave has sent an acknowledgment in the form of message OMS0055 (connection cleared down).

The partner address code is not deleted from the partner groups to which it belongs.



Messages relating to this partner in the logging file can no longer be accessed with the *TURN* command.

*Examples*

```
@: /CLSCON A1C
```

```
@: OMS0055 CONNECTION TO PARTNER name/processor CLEARED DOWN (PAC=A1C)
```

```
@: /
```

```
@: /C X
```

```
@: OMS0054 CLSCON COMMAND ACCEPTED (PAC=X)
```

```
@: /
```

```
...
```

```
@: OMS0055 CONNECTION TO PARTNER name/processor CLEARED DOWN (PAC=X)
```

```
@: /
```

## CMD - Execute BS2000 command

The *CMD* command is used to execute a BS2000 command with MCLP capability.



The *CMD* command is accepted only if test mode is active (*OPTION TESTMODE=YES*) and thus only from the OMNIS administrator if *SECURITY LEVEL≠HIGH*.

Operation	Operands
CMD	bs2000-command

*bs2000-command* is the BS2000 command (without a slash) which is to be executed.

The command is executed with the aid of the BS2000 macro *CMD*; see the “Executive Macros“ manual [4]. The results of executing the BS2000 command are output on the terminal in the form of OMNIS messages. This function is intended primarily for diagnostic purposes. If OMNIS is running under TSOS, the *CMD* command is rejected.



The use of the BS2000 commands *START-PROGRAM*, *LOAD-PROGRAM*, *ABEND* and *LOGOFF* should be avoided, since these commands will terminate OMNIS. OMNIS does not check whether any of these commands is entered.

### Examples

```
@: /OPTION TESTMODE=YES
```

```
@: /CMD SHOW-SYSTEM-STATUS
```

```
@: T1 T2 T3 T4PR T4PU T4FD T4TP T5AC T5KP T6 T7 T8
@: 0 0 2 1 0 0 0 0 0 0 0 2
```

```
@: /CMD SHOW-USER-STATUS
```

```
@: NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
@: OMNIS 3292 2 EXTP 9 250 2.6375 499 1
@: CHAOS 3713 3 DIA 0 250 36.4717 2000 W
@: MINERAL 3837 3 TPU 0 250 3.9071 2000 1
@: NAME TSN TYPE PRI SIZE COPIES CPU-USED RTSN
@: CHAOS 3097 4 LP3 250 17 0 3091
@: /
```

## CONFUPDATE - Update the configuration

The *CONFUPDATE* command causes OMNIS to read in a configuration file newly created with OMNISKD, and thus permits the OMNIS configuration to be modified while OMNIS is running.

The *CONFUPDATE* command acts on all the objects defined in the OMNIS configuration file (notably on exit groups, printers, multiplex connections and OMNIS-OMNIS connections).



The *CONFUPDATE* command is reserved for use by the administrator.

Operation	Operands
CO[NFUPDATE]	AP[ASS]= $\left. \begin{array}{l} \text{'passwd' } \\ ? \end{array} \right\}$

**APASS** specifies the valid administration password. If *SECURITY LEVEL=HIGH* is set, the *APASS* operand is rejected with the message OMS0190. The *CONFUPDATE* command can therefore be entered only at the BS2000 console if *SECURITY LEVEL=HIGH* is set.

**passwd** is the administration password.

**?** The administration password is to be prompted for by OMNIS in blanking mode after input of the command.

When executing *CONFUPDATE*, OMNIS does not check the consistency of the information in the configuration file. This means that faulty modifications (such as invalid entries) will be detected only when the entries are actually used (as the result, for example, of an *OPNCON* command).

In order to avoid having all currently connected users locked out until the configuration file has been read in, OMNIS rejects requests for access to files which were migrated from HSMS (Hierarchical Storage Management System).

If an error occurs, an error message is displayed on the terminal at which the *CONFUPDATE* command was issued.



**DECLARE-GROUP, DECLARE-PARTNER and DECLARE-TERMINAL interpretation**

The configuration changes become effective only when a terminal (re)connects itself to OMNIS (sign-off with *HOLD=NO*). Furthermore, connections with *CONNECT=START* are reestablished only if the terminal is not connected to OMNIS and is not in the HOLD state.

**DECLARE-EXIT interpretation**

Exit groups are uniquely identified by their *eac*. Following execution of *CONFUPDATE*, the exit declaration of the new configuration file applies. Exceptions:

- Any undeclared exit group (i.e. an exit group created by an *EXIT* command) which is present in OMNIS is retained even if it not declared in the new configuration file.
- Any exit group assigned by means of *SET* or *OPTION* to the terminal at which the *CONFUPDATE* command was entered is left unchanged.

If an exit needs to be loaded (typically owing to a recompilation), the *CONFUPDATE* must be followed the *EXIT* command with *LOAD=YES*.

**DECLARE-HARDCOPY interpretation**

Printers must be uniquely identified by their *hac* and their network address (*ptnname/proname*). If ambiguous arguments are used, the printer is left unmodified. OMNIS then displays an error message on the terminal at which the *CONFUPDATE* command was issued.

Any undeclared printer (i.e. a printer added by a *HARDCOPY* command) which is present in OMNIS is retained unmodified even if it not declared in the new configuration file. In all other cases the hardcopy declaration of the new configuration file applies following execution of *CONFUPDATE*.

If a printer which is to be removed is currently active, OMNIS initially places it in the *CLSCON-PENDING* state. That means that OMNIS will reject any new assignments but will not remove the printer until the queued output has been completed. Note however that OMNIS will only accept single messages no more than 4000 characters long from DCAM. It will not continue printing output which is longer than 4000 characters.

**DECLARE-MUX interpretation**

*MUX* entries are uniquely identified by their network address (*ptnname/proname*).

If a connection which is to be cleared is currently active, OMNIS initially places it in the *CLSCON-PENDING* state. That means that OMNIS will reject any new connection requests but will not clear the connection until there are no more sessions running on it.

**DECLARE-OMNIS interpretation**

OMNIS-OMNIS connections must be uniquely identified by their *oac* and their network address (*ptnname/proname*). If ambiguous arguments are used, the entry is left unmodified. OMNIS then displays an error message on the terminal at which the *CONFUPDATE* command was issued.

If an OMNIS-OMNIS connection which is to be cleared is currently active, OMNIS initially places it in the *CLSCON-PENDING* state. That means that OMNIS will reject any new connection requests but will not clear the connection until there are no more partners indirectly connected via the connection.

**DECLARE-PKEY evaluation**

The P key modules are reloaded.

If the P key modules of the current configuration are currently no longer required by any active application, they will be unloaded.

If an active application can still access them, they remain loaded. They can, however, be updated if a further update is carried out, in which case all configurations used to date are checked to see whether the P key modules can be unloaded.

## DUMP - Output dump

The *DUMP* command generates a memory dump without terminating OMNIS. Only the *DUMP* command without additional operands can be entered, i.e. a memory dump of OMNIS is created immediately after the command is input. By specifying operands, you can define a conditional memory dump, which is generated when a specific OMNIS message arrives.



The *DUMP* command is accepted only if test mode is active (*OPTION TESTMODE=YES*) and thus only from the OMNIS administrator if *SECURITY LEVEL≠HIGH*.

Operation	Operands
DUMP	[DUMP-MESSAGE = <omnis-message-number>] [,INSERT=(string,insert-number)]

DUMP-MESSAGE = <omnis-message-number>

By specifying the *omnis-message-number* (e.g. OMS0126), you can define a conditional memory dump, i.e. the memory dump is not generated until a specific message arrives. Only messages from message number OMS0001 through OMS0499 are permitted. The first time the specified message occurs, a memory dump is created. Then this setting is deleted, i.e. even if the specified message occurs several times, only one memory dump is created. The message can be specified more precisely with the *INSERT* operand.

INSERT=(string,insert-number)

By specifying *INSERT*, you can define more precisely a conditional memory dump which is generated in response to a specific message. You can specify that only a certain insert in the message text is to initiate the memory dump. The contents of the insert (e.g. address code) are defined with *string*. With *insert-number*, you can define the position of the insert in the message text (e.g. third insert = 3).

The memory dump is generated with the macro *CDUMP*; see also the “Executive Macros” manual [4]. The dump is written to a file under the user ID under which OMNIS is running.

This function is intended primarily for generating diagnostic documentation in the event of OMNIS errors without having to terminate OMNIS.

You should always take a dump in the case of errors which occur only sporadically. Similarly, you should take a dump if errors occur in conjunction with exit routines.

Enter the *DUMP* command at a different terminal from that on which the error occurred. In the event of problems with OMNIS-OMNIS connections, always take a dump of each OMNIS.

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

Parts of the OMNIS memory (including the *OPASS* passwords) are protected and are not output to the dump file.



OMNIS is blocked during dump output, i.e. OMNIS accepts no further messages.

*Example*

```
@: /OPTION TESTMODE=YES
```

```
@: /DUMP DUMP-MESSAGE=OMS0005, INSERT=(PAM, 1)
```

```
@: /
```

## END - Terminate OMNIS

The *END* command terminates OMNIS.



The *END* command is reserved for use by the administrator.

Operation	Operands
END	AP[ASS]= $\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$

### APASS

specifies the valid administration password. This operand must be specified regardless of whether or not the terminal is defined as the administrator (with *OPTION ADM=YES, DECLARE-TERMINAL ADM=YES*). This ensures that an *END* input intended for a partner which was inadvertently entered with the partner address code @: does not terminate OMNIS. If *SECURITY LEVEL=HIGH* is set, the *APASS* operand is rejected with the message OMS0190.

The *END* command can, therefore, be entered only from the BS2000 console when *SECURITY LEVEL=HIGH*.

passwd

is the administration password.

?

After the command has been input, the administration password is to be prompted for by OMNIS in blanking mode.

The *END* command terminates all existing connections (to OMNIS terminals and to OMNIS partners) and closes all files.



Before the *END* command is issued, you should ensure that all OMNIS users have concluded their dialog with OMNIS (*INF SYSTEM, TYPE=SUMMARY* command). If there are still some users connected to OMNIS, a message should be broadcast (command *MESSAGE ALL,'text'*) requesting that they terminate their dialog with OMNIS.

### Example

```
@: /END AP='HUGO'
@: OMS0013 OMNIS NORMALLY TERMINATED
DISCONNECTED
```

## EXIT - Administer exit routines

The *EXIT* command is used to manage or administer the exit routines, i.e. to generate, load, delete and modify such routines.



Depending on the setting defined with the *PRIORITY* command, the *EXIT* command may be reserved for use by the administrator.

Operation	Operands
E[ <i>XIT</i> ]	eac  $[,A[CTION]=\left\{ \begin{array}{l} M[ODIFY] \\ C[REATE] \\ D[ELETE] \end{array} \right\}]$  $AP[ASS]=\left\{ \begin{array}{l} 'passwd' \\ ? \end{array} \right\}$  $[,M[ODULES]=\left\{ \begin{array}{l} module \\ (module[,...]) \end{array} \right\}]$  $[,L[OAD]=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}]$

- eac specifies the address code of the exit group.
- ACTION defines the function to be performed.
  - CREATE A new exit group is to be defined; at least one module name must be specified.
  - DELETE The exit group is to be deleted.
  - MODIFY An existing exit group is to be redefined; at least one module name must be specified. This function is equivalent to consecutive input of the two commands  
 EXIT eac, ACTION=DELETE  
 EXIT eac, ACTION=CREATE,MOD=module
- APASS specifies the valid administration password. This operand is required only if the *EXIT* command is available only to the administrator, i.e. when *PRIORITY EXIT-AUTORIZATION=ADM*
- passwd is the administration password.

?	After the command has been input, the administration password is to be prompted for by OMNIS in blanking mode.
MODULES	specifies the exit routines which are to belong to the group.
LOAD	specifies whether or not modules are to be loaded.
NO	Only those modules are loaded which have not yet been loaded. Modules already loaded are merely assigned to the group.
YES	All modules are loaded. Modules already loaded are first deleted.

With one *EXIT* statement, up to 14 exit routines (modules) can be assigned to one EAC. The total set of all exit routines assigned to one EAC forms an exit group. Any module may appear only once in a given exit group.

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

Using one of the commands *SET*, *OPTION* or *PARAM*, an exit group can be assigned either globally, or to one specific terminal or partner.

If the *EXIT* command contains the EAC of the current group (currently assigned with *SET* or *OPTION*), the command is not executed. Execution of the command is considered successful only if it has been possible to load all exit routines. If even just one exit routine could not be loaded, the exit group involved is deleted (even if *ACTION=MODIFY*).

In order to avoid having all currently connected users locked out until the modules from the EXIT module library have been read in, OMNIS rejects requests for access to EXIT module libraries which were migrated from HSMS (Hierarchical Storage Management System).

### Examples

```
@: /EXIT EX2, ACTION=CREATE, MODULES=MOD3
```

```
@: OMS0107 EXIT GROUP 'EX2' DECLARED
```

```
@: /EXIT EX3, ACTION=CREATE, MODULES=MOD3
```

```
@: OMS0107 EXIT GROUP 'EX3' DECLARED
```

```
@: /INF EXIT
```

```
@: EAC EID MODULES
```

```
@: EX3 15 MOD3
```

```
@: EX2 11 MOD3
```

```
@: /EXIT EX3, ACTION=DELETE
```

```
@:/INF EXIT
@: EAC EID MODULES
@: EX2 11 MOD3

@:/EXIT EX2,ACTION=MODIFY,MODULES=(MOD5,MOD3,MOD1),LOAD=YES
@: OMS0107 EXIT GROUP 'EX2' DECLARED

@:/INF EXIT
@: EAC EID MODULES
@: EX2 16 MOD5 MOD3 MOD1

@:/PAR P11,EXIT=EX2
@:/PAR P12,EXIT=EX2
@:/OPT EXIT=EX2

@:/INF EXIT,EXIT=EX2
@: EXIT= EX2
@: TID: 14(A)
@: PID: 24(A) 23(A)
@:/
```



## FILE - Process command file

The *FILE* command starts the processing of a command file.

Operation	Operands
F[ILE]	$I[INPUT]=\left\{ \begin{array}{l} \text{filename} \\ \text{library-name(member)} \end{array} \right\}$ $[,D[ISPLAY]=\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ]$ $[,AP[ASS]=\left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} ]$

- INPUT** specifies the name of the command file. The command file is read via SYSDTA and thus may be either a SAM or ISAM file or a type *S* library member (see the manuals “Commands“ [3] and “LMS“ [8]). In order to prevent all users connected to OMNIS from being locked out until the command file has been read in, OMNIS rejects any attempt to access command files which have been migrated from HSMS (Hierarchical Storage Management System).
- DISPLAY** specifies whether the input read from the command file is to be output to the terminal.
- YES** The input read from the command file is to be output to the terminal.
- i**
- DISPLAY=YES* may be specified only by the administrator.
- NO** The input read from the command file is not to be output to the terminal.
- APASS** specifies the administration password (meaningful only if *DISPLAY=YES* is specified).
- passwd** is the valid administration password.
- ?** After the command has been input, the administration password is to be requested by OMNIS in blanking mode.

A user may enter the *FILE* command only if *SECURITY LEVEL=LOW* is set. The administrator may also enter the *FILE* command if *SECURITY LEVEL=MEDIUM* is set, providing OMNIS is not running under TSOS. If *SECURITY LEVEL=HIGH* is set, the *FILE* command is rejected.

The command file to be processed may contain any OMNIS commands or messages to partners, and these will be executed in sequence or forwarded to the partners just as if they had been entered from a terminal.

The rules for specification of the partner address code are the same as those for input at a terminal: if no PAC is specified, the message is sent to the partner to whom the last message was sent.

If the *FILE* command is issued from an exit routine, the partner address code must be specified in the file.

The processing of a *FILE* command is interrupted by a further *FILE* command (in this file) or by the *MESSAGE* command with *INPUT=filename* (i.e. nesting is not possible).

Further details on the processing of command files can be found in the “Administration and Programming” manual [1].



OMNIS does not accept any messages from terminals or partners while processing a command file. Long command files can interfere with OMNIS operation for all users.

In the case of messages to UTM and TIAM partners, correct execution cannot be guaranteed, since the protocol to be observed for this connection does not permit asynchronous inputs.

*Examples*

*A* is a TIAM partner and *U* a UCON partner.

The file *command-file* looks like this:

```
@:SET ACK=YES,LOGGING=YES
@:MDEF MTB1,ACTION=CREATE
A:/SHOW-USER-STATUS
U:/STA MSG
@:MDEF MTB1,MSG=DMS0715,R='.NO'
@:MDEF MTB1,MSG=EXC0222,D=N

@:/FILE INPUT=command-file
@: OMS0040 MESSAGE TABLE MTB1 CREATED

A:NAME      TSN TYPE  PRI      CPU-USED CPU-MAX ACCOUNT#
OMNISTST 0001 3 DIA  0 250      5.0234   2000 1
PARFUM   4711 3 DIA  0 250     10.2456   2000 1
A:% S171 NO SPOOLOUT PRESENT
A:/
U:% 0 00.142355 % EXC0031 NO MESSAGE OUTSTANDING
U:% 0 00.142355 % NBR0709 '/STATUS'-COMMAND PROCESSING COMPLETED.

@:/F I=command-file,D=Y,AP=X'478A3256'
@:SET ACK=YES,LOGGING=YES
@:MDEF MTB1,ACTION=CREATE
@: OMS0040 MESSAGE TABLE MTB1 CREATED
A:/SHOW-USER-STATUS
U:/STA MSG
@:MDEF MTB1,MSG=DMS0715,R='.NO'
@:MDEF MTB1,MSG=EXC0222,D=N
@:/
```

## GROUP - Define partner group

The *GROUP* command is used to define a partner address code which allows common access to any number of partners.

Operation	Operands
G[ROUP]	gac  [,ADD= $\left\{ \begin{array}{l} \text{pac} \\ (\text{pac } [, \dots]) \end{array} \right\}$ ]  [,DEL= $\left\{ \begin{array}{l} \text{pac} \\ (\text{pac } [, \dots]) \end{array} \right\}$ ]  

**gac** is the group address code of the partner group to be defined. This group address code may not already be defined as a partner address code. Each group address code used on a terminal must be unique.

**ADD** specifies a partner address code or a list of partner address codes (not group address codes) which is/are to be assigned to the group *gac*.

**DEL** specifies a partner address code or a list of partner address codes (not group address codes) which is/are to be deleted from the group *gac*.

A group is set up as soon as a group address code is first assigned to it, and is deleted once the last partner address code has been deleted from it.

The total set of all members (partner address codes) assigned to one group address code forms a group.

With one *GROUP* command, up to 25 members can be assigned to one group address code. However, the number of group members can be extended by entering further *GROUP* commands with the same group address code.

The group address code can be entered in message inputs and in some commands (e.g. *OPNCON*, *PARAM*) instead of the partner address code.

The groups \*, \*C, \*D, \*T and \*U are predefined and are managed automatically by OMNIS.

These predefined groups contain:

- \* all partners of a terminal
- \*C all UCON partners
- \*D all DCAM partners
- \*T all TIAM partners
- \*U all UTM partners of the terminal



The defined groups and the group address codes assigned to them can be queried with the *INF GROUP* command.

### *Examples*

```
@: /INF G
```

```
@: NO GROUP TO DISPLAY.
```

```
@: /GROUP A,ADD=(X,Y,Z)
```

```
@: /INF G
```

```
@: GAC PAC-LIST
```

```
@: A X Y Z
```

```
@: /G A,ADD=(U,V),DEL=X
```

```
@: /INF G
```

```
@: GAC PAC-LIST
```

```
@: A Y Z U V
```

```
@: /G A,DEL=(U,V,Y,Z)
```

```
@: /INF G
```

```
@: NO GROUP TO DISPLAY.
```

```
@: /
```

## HALT - Sign off terminal from OMNIS

A terminal terminates its connection to OMNIS by means of the *HALT* command.

Operation	Operands
HA[LT]	$[HO[LD]=\left. \begin{array}{l} S[TD] \\ Y[ES] \\ A[UTO] \\ N[O] \end{array} \right\}]$ $[,OP[ASS]=\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}]$

- HOLD** specifies whether the connections to the partners are to be retained or closed.
- NO** The connections are to be closed.
  - YES** The connections are to be retained. Specifying *HOLD=YES* for a terminal (with *OPTION*, *HALT* or *DECLARE-TERMINAL*) is effective only for partners for whom *HOLD=NO* has not been set (with *OPNCON*, *PARAM* or *DECLARE-PARTNER*). *HOLD=YES* is effective only for partens for whom OMNIS can assign an *OPASS*.  
  
Connections to Kerberos-authenticated partners (previously chip card) are always closed when the connection to the terminal is lost.  
  
If *SECURITY LEVEL≠LOW*, partners do not switch to the *HOLD* status.
  - AUTO** The partners are taken over again when a connection is reestablished after being lost. Specifying *HOLD=AUTO* for a terminal (with *OPTION*, *HALT* or *DECLARE-TERMINAL*) is effective for all partners, regardless of their *HOLD* settings.  
  
Connections to Kerberos-authenticated partners (previously chip card) are always closed when the connection to the terminal is lost.  
  
In the case of a **non-unique user** and *SECURITY LEVEL≠LOW*, the specification *HOLD* is ignored; *HOLD=NO* always applies. Whether or not a user is unique is checked by OMNIS-MENU.
  - STD** The value specified with one of the commands *SET*, *OPTION* or *DECLARE-TERMINAL* applies.

OPASS	specifies an <i>OCCUPY</i> password for the terminal (meaningful only if <i>HOLD=YES</i> ). If the <i>OPASS</i> operand is not specified, the password specified with one of the commands <i>OPTION</i> or <i>DECLARE-TERMINAL</i> applies.
passwd	is the <i>OCCUPY</i> password.
?	The <i>OCCUPY</i> password is to be requested by OMNIS in blanking mode.

All connections to partners which the terminal has set up are either closed (implicit *CLSCON* command for each partner) or remain open (in this case the dialog can be resumed after renewed sign-on of the terminal to OMNIS or from another terminal).

After the *HALT* command, a terminal is no longer connected to OMNIS and must conduct a new network dialog (PLEASE ENTER NET COMMAND, PLEASE REQUEST PARTNER).

*HOLD=YES/AUTO* is only relevant if there are still connections set up to partners, and you intend to continue working with them after the *HALT* and a new terminal sign-on.

If no active connections to partners exist, *HOLD=YES/AUTO* has no effect; an undeclared terminal is removed from the list of terminals, a declared one set to inactive.

*HOLD=AUTO* is valid from the time the *HALT* command is entered until a new setting is specified with the *HALT* or *OPTION* command, even after a new sign-on to OMNIS. The same applies to *HOLD=YES*. Here, the terminal must first be occupied after a new sign-on. *HOLD* settings made with the *HALT* command have a higher priority than settings made with *OPTION* and *DECLARE-TERMINAL*.

With *HALT HOLD=AUTO* and *SET HOLD=NO*, the setting *HOLD=NO* applies after the command *PRIORITY HOLD-PRIORITY=SET-OPT*.



The terminals which have signed off with *HOLD=YES/AUTO* can be queried with the *INF TERMINAL, ACTIV=HOLD* command.

### Examples

```
@: /HALT
```

```
@: OMS0020 OMNIS SESSION NORMALLY TERMINATED
DISCONNECTED
(Terminal=(MF311085/V214111); TID=0060)
```

```
@: /HALT HOLD=YES,OP=X'53DF3F49'
```

```
@: OMS0020 OMNIS SESSION NORMALLY TERMINATED
DISCONNECTED
(Terminal=(MF311085/V214111); TID=0060)
```

## HCOPY - Enter hardcopy unit in printer table

The command *HCOPY* is used to enter a hardcopy unit or RSO printer in the printer table or to defined a terminal on which the dialog conducted on another printer can be output in parallel. It can also be used to specify the behavior of OMNIS if the printer fails.

Operation	Operands
HC[OPY]	hac ,PR[ONAME]=proname ,PT[NNAME]=ptnname [,CON[NECT]= $\left. \begin{array}{l} \text{S[TART]} \\ \text{U[SE]} \end{array} \right\} ]$ [,FA[ILURE]= $\left. \begin{array}{l} \text{*I[GNORE]} \\ \text{*S[AVE]} \\ \text{hac} \end{array} \right\} ]$

- hac** specifies a hardcopy address code which uniquely identifies a printer (or a terminal for *LISTENING*; see [page 120](#)). The hardcopy address codes of each printer must be unique within the OMNIS system.
- PRONAME** specifies, together with *PTNNAME*, the symbolic network address of the printer (or the terminal for *LISTENING*; see [page 120](#)).
- PTNNAME** specifies, together with *PRONAME*, the symbolic network address of the printer (or the terminal for *LISTENING*; see [page 120](#)).
- CONNECT** specifies when the connection to the printer is to be set up.

  - START** The connection to the printer is to be set up as soon as the printer is entered in the table.
  - 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** specifies what is to happen when a printer fails.

  - \*IGNORE** Messages destined for the printer are to be ignored (and discarded).
  - \*SAVE** Messages destined for the printer are to be written to the paging file and output when the printer is again available (*RESTART* command).
  - hac** Messages destined for the printer are to be output on the alternate device with the hardcopy address code *hac*.



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 operational.

A few messages (up to 10) may be lost when a printer fails. Those are the messages which OMNIS had already sent to the printer before the failure was detected.

The connection to the printer can be cleared down by the administrator by means of the *CANCEL* command.

The connection to a printer is cleared down if it is no longer assigned to a partner or to a terminal.

A printer which has failed can be reactivated by means of the *RESTART* command.

*Example*

```
@: /HCOPY HC01,PTN=DR511144,PRO=V214V111
```

```
@: /HCOPY HC02,PTN=DR511155,PRO=V214V111,FAILURE=HC01
```

```
@: /OPTION HCOPY=HC02
```

```
@: /I HC,AC=ALL
```

```
@: HAC HID PTNNAME /PRONAME STATUS INOP
```

```
@: HC02 0012 DR511155/V214V111 ACTIVE HC01
```

```
@: HC01 0011 DR511144/V214V111 INACT NONE
```

```
@: /
```

## HELP - Assistance with commands and messages

The *HELP* command causes OMNIS to output information about OMNIS commands, OMNIS and/or BS2000 messages and DCAM return codes at the terminal.

Operation	Operands
H[ELP]	
H[ELP]	bs2000-message-number
H[ELP]	command-name
H[ELP]	omnis-message-number
H[ELP]	dcam-return-code

### Format 1: "HELP" without operands

outputs a list of all OMNIS commands with a brief description of each.

### Format 2: "HELP bs2000-message-number"

outputs detailed information about the specified BS2000 message. If *SECURITY-LEVEL=HIGH* is set, this command is rejected for reasons of data protection; if *SECURITY-LEVEL=MEDIUM* is set, the terminal must have the administrator privilege.

### Format 3: "HELP command-name"

outputs a precise description of the syntax for the specified command, together with a brief description of its effect.

### Format 4: "HELP omnis-message-number"

outputs detailed information on the reasons for the specified message and lists possible actions or responses. The message number must be specified in the form *OMSnnnn*.

### Format 5: "HELP dcam-return-code"

displays the meaning of a DCAM return code. A DCAM return code is output in the message *OMS0005 macro-ERROR code AT ADDRESS address* (if the first letter of *macro* is a "Y"). The *dcam-return-code* must be specified in the form *YDDcccc*, where *cccc* represents the first four characters of *code*. For compatibility reasons, the form *YDDcccc* is also accepted.

**Description of the metalanguage symbols in HELP texts:**

Not all of the metalanguage symbols used in the manual can be displayed on the screen; the following characters are therefore used instead:

Notation in the manual	Notation on the screen	Meaning
lowercase letters	:lowercase letters:	Variables which are to be replaced by concrete values.
[ ]	< >	Operands which can be omitted.
{ } e.g. $\left. \begin{array}{l} \text{Y[ES]} \\ \text{NO} \end{array} \right\}$	! e.g. YES ! NO	Alternatives, one or none of which can be selected, e.g. YES/NO
<u>underlined</u>	no underlining on screen, first value in the list	Default value if no value is specified.
[ ]	not displayed on the screen	Character sequences which may be omitted within a command/operand or variable.

*Examples*

@:/HELP

@: OMS0900 OMNIS HELP: OVERVIEW OF COMMANDS

@: -

@: - BREAK	- Terminate connection	HELP	- Assistance
@: - CALL	- Expect input	INF	- Output information
@: - CANCEL	- Close connection	MDEF	- Define message table
@: - CHANGELOG	- Switch logging file	MESSAGE	- Send message to a terminal
@: - CLSCON	- Close connection	OCCUPY	- Take over connection
@: - CMD	- Execute BS2000 cmd.	OPNCON	- Open connection to partner
@: - CONFUPDATE	- Configuration update	OPTION	- Define term.-spec. attrs.
@: - DUMP	- Output dump	PARAM	- Define ptrn.-spec. attrs.
@: - END	- Terminate OMNIS	PRIORITY	- Define scope of commands
@: - EXIT	- Admin. exit-routine	RESTART	- Restart connection
@: - FILE	- Process command file	SET	- Set operating parameters
@: - GROUP	- Define partner group	TRACE	- Define trace function
@: - HALT	- Sign off terminal	TURN	- Page-turn in logging file
@: - HCOPY	- Enter hardcopy unit		

@: - A more detailed description of a command can be obtained by

@: - entering '@:HELP :command:'.

@: RESPONSE

@:/

@:/HELP HELP

@: OMS0907 OMNIS HELP: 'HELP' COMMAND

@: - The HELP command displays more information about commands, messages,  
 @: - return codes (the first 4 characters of :code: in the  
 @: - message 'OMS0005 :macro: ERROR :code: AT :address:') and BS2000 messages.  
 @: - HELP 'command overview'  
 @: - HELP :bs2000-message: 'BS2000-Message'  
 @: - HELP :command-name: 'command syntax'  
 @: - HELP :message-number: 'explanation of message'  
 @: - HELP OMY:dcam-return-code: 'explanation of DCAM return code'  
 @: - The following metasyntax is used: names in uppercase letters must  
 @: - remain as they stand; names in lowercase letters must be replaced by  
 @: - current values; operands in angle brackets are optional; variables are  
 @: - additionally enclosed in colons; comments are enclosed in quotes.  
 @: - BS2000 messages must not be entered, when SECURITY-LEVEL=HIGH is set.  
 @: RESPONSE  
 @:/

**@:/HELP MDEF**

@: OMS0908 OMNIS HELP: 'MDEF' COMMAND  
@: - The MDEF command is used for message table administration.  
@: - MDEF :table-name:  
@: - < ,ACTION= CREATE ! DELETE ! MODIFY >  
@: - < ,BELL= YES ! NO >  
@: - < ,DISPLAY= YES ! NO >  
@: - < ,ID= :string: ! (:string:::position:) >  
@: - < ,INSERT= (:string:::insert-number:) >  
@: - < ,MSG= :message-number: ! (:message-number:::position:)>  
@: - < ,REPLY= :string: ! (:string:,...) >  
@: - CREATE creates message tables.  
@: - DELETE deletes message tables.  
@: - MODIFY is used to generate/update an entry.  
@: - MSG and ID identify the message and its position.  
@: - REPLY defines one or more automatic replies.  
@: - DISPLAY=NO suppresses message output.  
@: - BELL=YES triggers an audible alarm.  
@: RESPONSE  
@:/

**@:/HELP YDD0C4C**

@: DCAM RETURN CODE 0C4C: PARTNER NOT ACCESSIBLE (THE APPLICATION IS  
@: NOT OPENED).  
@:/

**@:/HELP OMS0001**

@: OMS0001 USER IS CONNECTED TO OMNIS (TERMINAL=((&00)/(&01)); TID=(&02))  
@: - No further information.  
@: - (&00): station-name  
@: - (&01): processor-name  
@: - (&02): terminal-identifier  
@: RESPONSE  
@:/

## INF - Output information

The *INF* command allows you to call up current information: about connections such as partners, terminals and groups; about global resources such as hardcopy, MUX connections, OMNIS-OMNIS connections, exit routines and message tables, as well as the settings of OMNIS.

*INF* displays the information according to your authorization (administrator, user identified by name, user not identified by name) and the operands specified.



Information about all partners, terminals, partner groups is available only to the administrator (specification of *APASS*, *DECLARE-TERMINAL ADM=YES* or *OPTION ADM=YES*). By switching on the test mode (*OPTION TESTMODE=YES*), the administrator can obtain additional information about partners, terminals, multiplex connections and OMNIS-OMNIS connections. However, test mode can be activated only if *SECURITY LEVEL≠HIGH*.

A user identified by name can call up any information which is registered under his or her name.

No.	Operation	Operands
1	I[NF]	A[LL]
2	I[NF]	P[ARTNER] $\left\{ \begin{array}{l} [PID = \text{number}] \\ [PAC = \text{pac}] \left\{ \begin{array}{l} [, TID = ] \left\{ \begin{array}{l} OWN \\ num \end{array} \right\} \\ [, T[ERMINAL] = ([ptn][,pro])] \\ [, U[SER] = \text{user}] \end{array} \right\} \end{array} \right\}$ $[, AC[TIV] = \left\{ \begin{array}{l} Y[ES] \\ A[LL] \\ H[OLD] \\ N[O] \end{array} \right\} ]$ $[, AP[ASS] = \left\{ \begin{array}{l} 'passwd' \\ ? \end{array} \right\} ]$

No.	Operation	Operands
3	I[NF]	<p>T[ERMINAL]</p> <p>{ [,TID = ] { OWN num } }</p> <p>{ [,T[ERMINAL] = ([ptn],[pro])][,U[SER] = user] }</p> <p>[,AC[TIV]= { Y[ES] A[LL] H[OLD] N[O] } ]</p> <p>[,AP[ASS]= { 'passwd' ? } ]</p>
4	I[NF]	<p>HC[OPY]</p> <p>[,HID=number]</p> <p>[,AC[TIV]= { ALL Y[ES] N[O] } ]</p>
5	I[NF]	<p>G[ROUP]</p> <p>[,TID= { O[WN] num } ]</p> <p>[,AP[ASS]= { 'passwd' ? } ]</p>

No.	Operation	Operands
6	I[NF]	MT[AB]  [,MT[AB]= $\left. \begin{array}{l} \text{name} \\ \text{ALL} \end{array} \right\}$ ]
		[,MSG= $\left. \begin{array}{l} \text{message-number} \\ \text{ALL} \end{array} \right\}$ ]
7	I[NF]	MU[X]  [,AC[TIV]= $\left. \begin{array}{l} \text{Y[ES]} \\ \text{A[LL]} \\ \text{N[O]} \end{array} \right\}$ ]  [,MID=number]
8	I[NF]	E[XIT] [,EX[IT]=eac]
9	I[NF]	O[MNIS]  [,AC[TIV]= $\left. \begin{array}{l} \text{Y[ES]} \\ \text{N[O]} \\ \text{A[LL]} \end{array} \right\}$ ]  [,OID=number]
10	I[NF]	S[YSTEM]  [,T[YPE]= $\left. \begin{array}{l} \text{S[ETTINGS]} \\ \text{P[ARAMETERS]} \\ \text{D[IAGNOSTICS]} \\ \text{SU[MMARY]} \end{array} \right\}$ ]
11	I[NF]	TR[ACE]



**Format 1: "INF ALL"**

Shows how many partners/terminals are connected to OMNIS. This command is equivalent to the command *INF SYSTEM TYPE=SUMMARY*.

**Format 2a: "INF PARTNER" without PID or PAC**

Displays a list of partners of a selected terminal or of a user. Only the most important information (*PAC, KPAC, PID, PTNNAME, PRONAME, TYPE, STATUS, ROUTE*) is displayed for each partner.

TID	selects the terminal for whose partners information is to be displayed.
OWN	Information about the partners of one's own terminal is to be displayed.
num	Information about the partners of the terminal with TID <i>number</i> is to be displayed.
TERMINAL	is the network address of the terminal whose partners are to be selected.
ptnr	is the partner name of the terminal.
pro	is the processor name of the terminal.
USER	is the user whose partners are to be selected.
user	is the name of the user.
ACTIV	selects the partners to be displayed.
YES	Only the active partners are to be displayed (i.e. those to which a connection exists).
NO	Only the inactive partners are to be displayed (i.e. those that are declared, but to which no connection exists).
ALL	All partners are to be displayed (whether active or not).
HOLD	A list of all partners with HOLD state is to be displayed.
APASS	specifies the administration password. If <i>SECURITY LEVEL=HIGH</i> , the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	is the administration password.
?	After the command has been input, the administration password is to be requested by OMNIS in blanking mode.

**Format 2b: "INF PARTNER" with PID or PAC**

Displays all information about one single partner. The format differs depending on whether or not test mode has been activated. If *ROUTE=MUX* is defined, additional information is output. This information is output only if test mode has been activated and is intended for diagnostic purposes.

PAC	specifies the partner address code of the partner for which the information is to be displayed.
PID	specifies the partner about which information is to be displayed.

**Format 3a: "INF TERMINAL" without TID**

Displays information about selected terminals. Only the most important information (*TID*, *DSTNAME*, *PRONAME*, *TYP*, *STATUS*, *ROUTE*) is displayed for each terminal. The selection criteria are linked with a logical "AND".

TERMINAL	is the network address of the selected terminal.
ptnr	is the partner name of the terminal.
pro	is the processor name of the terminal.
USER	is the selected user.
user	is the name of the user.
ACTIV	selects the terminals about which information is to be displayed.
YES	Only the active terminals are to be listed (i.e. those to which a connection exists).
NO	Only the inactive terminals are to be listed (i.e. those that are declared, but to which no connection exists and which have no partners).
HOLD	All terminals to which no connection exists any more, but which still have active partners, are to be listed.
ALL	All terminals are to be listed.
APASS	specifies the administration password. If <i>SECURITY LEVEL=HIGH</i> , the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	is the administration password.
?	After the command has been input, the administration password is to be requested by OMNIS in blanking mode.

**Format 3b: "INF TERMINAL" with TID**

Displays all information about a single terminal. The format depends on whether or not test mode has been activated.

TID	selects the terminal for which information is to be displayed.
OWN	Information about the partners of one's own terminal is to be displayed.
num	Information about the partners of the terminal with TID <i>number</i> is to be displayed.

**Format 4a: "INF HCOPY" without HID**

Displays a list of directly connected printers.

ACTIV	selects the printers about which information is to be displayed.
YES	A list of all active printers, (i.e. those to which a connection exists) is to be displayed.
NO	A list of all inactive printers (i.e. those that are declared, but to which no connection exists).
ALL	All printers are to be listed.

**Format 4b: "INF HCOPY" with HID**

Displays a list of all partners or terminals to which the specified printer is assigned.

HID	selects the desired printer.
-----	------------------------------

**Format 5: "INF GROUP"**

Lists the partner groups of a terminal.

TID	selects the terminal whose partner groups are to be displayed.
OWN	The partner groups of the user's own terminal are to be displayed.
num	The partner groups of the terminal with TID <i>number</i> are to be displayed.
APASS	specifies the administration password. If <i>SECURITY LEVEL=HIGH</i> , the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	is the administration password.
?	After the command has been input, the administration password is to be requested by OMNIS in blanking mode.

**Format 6: “INF MTAB“**

Displays information about message tables.

MTAB	specifies the message tables about which information is to be displayed.
name	Information is to be displayed about the message table <i>name</i> .
ALL	Information is to be displayed about all message tables.
MSG	selects the message table entries to be displayed.
message-number	Message table entries with the message number <i>message-number</i> are to be displayed.
ALL	All message table entries are to be displayed. INF MTAB with no further operands lists all message tables.

**Format 7a: “INF MUX“ without MID**

Displays a list of the multiplex connections. The information displayed for each connection is the MID (the identification number of the multiplex connection), the declaration operands and the number of active sessions.

ACTIV	selects the multiplex connection about which information is to be displayed.
YES	All active multiplex connections are to be displayed.
NO	All inactive multiplex connections are to be displayed.
ALL	All multiplex connections are to be displayed.

**Format 7b: “INF MUX“ with MID**

Displays information about a certain multiplex connection. This information is output only in test mode and is intended for diagnostic purposes.

MID	is the identification number of the multiplex connection.
-----	---

**Format 8a: “INF EXIT“ without EAC**

Lists all exit groups.

**Format 8b: "INF EXIT" with EAC**

Lists all partners or terminals assigned to the specified exit group.

EXIT specifies the address code of the exit group *eac*.

**Format 9a: "INF OMNIS" without OID**

Lists all OMNIS-OMNIS connections. Only the most important data (*OAC*, *OID*, *PTNNAME*, *PRONAME*, *STATUS*) is displayed for each connection.

ACTIV selects the OMNIS-OMNIS connections to be displayed.

YES All active OMNIS-OMNIS connections are to be displayed.

NO All inactive OMNIS-OMNIS connections are to be displayed.

ALL All OMNIS-OMNIS connections are to be displayed.

**Format 9b: "INF OMNIS" with OID**

Displays all information about a single OMNIS-OMNIS connection. Additional information about an OMNIS-OMNIS connection is displayed if test mode is active.

OID specifies the OID of the OMNIS-OMNIS connection.

**Format 10: "INF SYSTEM"**

Displays global information.

TYPE selects the information to be displayed.

SETTINGS The parameters set by the administrator using the *SET* command are to be displayed.

PARAMETERS The parameters defined on starting OMNIS are to be displayed.

DIAGNOSTICS The following diagnostic information should be output by OMNIS:

- Management information of the logging file
- Information about the creation of a memory dump in the case of a particular OMNIS message number
- Information about the enabled or disabled trace (same information as with INF TRACE)

SUMMARY The number of partners/terminals connected to OMNIS is to be displayed.

**Format 11: “INF TRACE“**

Displays global information about the currently activated and deactivated traces. If at least one trace is active, there is also a current trace file, the name of which is also output.

*Examples*

```
@: /INF S,TYPE=SETTINGS
```

```
@: DSTMAX =          1000, PTNMAX =          2000, PACMAX =          2000,
@: APASS =           YES, HOLD =           *NO, HCT-FORM=          STD,
@: HCOPY =           *NO, LOGGING =          YES, ACK =           YES,
@: MTAB =           *NO, EXIT =           *NO, OPNCON =          FREE,
@: BREAK-KEY =       *NO, CALL-KEY =         K3, CALL-INF =        YES,
@: PAC =             *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG =*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =             *NO,
@: EXIT-PRIO = PAR-OPT-SET, REPLY =          ALL, EXIT-AUTH =      ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT=           OFF MDEF-AUTH =      ADM,
@: HOLD-PRIO = OPT-SET      , INPUT-SAVE=     *NO OPNSTART =      YES,
@: EXCL-PARTNER =     NO, SAVE =            *NO. RTIMER =         ON,
@: INPUT-SAVE-NUMBER = *NO.
@: /
```

```
@: /INF S,TYPE=PARAMETERS
```

```
@: APPNAME/NEA =      OMNIS
@: APPNAME/ISO =      OMNISISO
@: PREFIX =           OMS
@: PRONAME =          G133H207
@: VIRT. PRONAME =    G133H207
@: LOGGING-FILE =     OMNIS.LOGGING.001
@: STARTUP-FILE =     OMNIS.STARTUP
@: CONFIG-FILE =      OMNIS.CONFIGURATION
@: MODUL-LIBRARY =    OMNIS.MODLIB
@: BULLETIN-FILE =    OMNIS.BULLETIN
@: TEXT-FILE =        OMNIS.TEXT
@: PAGE-POOL =        OMNIS.PAGE
@: IOAREA-LENGTH =    04096
@: TWORK-LENGTH =     00256
@: PWORK-LENGTH =     00256
@: TEXTKEY-LENGTH =   00004
@: SECURITY LEVEL =    MEDIUM
@: DCAM INTVERS =     08
@: VTSU-B VERSION =   13.0A
@: VTSUCB-VERSION =   06
@: /
```

```
@: /INF S,TYPE=DIAGNOSTICS
```

```
@: LOGGING FILE      = EK.LOG.001
@: ACTUAL SIZE       = 00000104
```

```

@: SIZELIMIT      = 00020000
@: REPEAT        = DAILY (TIME = 00:00          )
@: ACTION        = CHANGE
@: LOGGING-COPY  =
@: DELETE       = NO
@: DUMP MESSAGE= OMS0005,(INSERT = PAM          , 01)
@: TRACE-DATEI  =      SYS.DIA.OMNIS.83XE.1
@: CONNECTION-TRACE =      YES
@: EXIT-TRACE   =      NO
@: TRANSPORT-TRACE =      SELECT
@:      SELECT TERM = STAT0818/ATPC3EJC
@:      SELECT HCOPY =
@:      SELECT MID =
@:      SELECT OMNIS =
@:/

@:/INF S,TYPE=SUMMARY
@: NUMBER OF TERMINALS:      2
@: NUMBER OF PARTNER:      2
@:/

@:/INF PARTNER,AC=ALL
@: PAC      PID PTNNAME /PRONAME  TYP  STATUS ROUTE  KPAC USERID TID=      27
@: P10      33 $DIALOG /G133H207 TIAM ACT   DIR    NO  TSOS
@: P6       32 $CONSOLE/G133H207 UCON ACT   DIR    NO  TSOS
@: P7       31 $DIALOG /G133H207 TIAM INACT  IND    NO  TSOS
@: P8       30 $DIALOG /G133H207 TIAM INACT  DIR    NO  TSOS
@: P9       29 $DIALOG /G133H207 TIAM INACT  DIR    NO  TSOS
@: M        28 OMTM11 /G133H207 UTM  INACT  MUX    NO  TSOS
@:/

@:/INF PARTNER,PID=33      "Test mode is inactive"
@: ZVT-ENTRY FOR PID=      33, TID=      27:
@: PAC =      P10, PTNNAME = $DIALOG, PRONAME = G133H207,
@: PID =      33, TYPE =      TIAM, APPNAME = $$$$$$$$,
@: STATUS = INACT, ROUTE =      DIR, OPASS =      NO,
@: MTAB =      STD, EXIT =      STD, HOLD =      STD,
@: KPAC =      NO, CHANGE =      YES, HCOPY =      STD,
@: CLASS =      OUT, COLOUR =      GREEN, PROTOCOL =      OMNIS,
@: LOGGING =      STD, LPASS =      NO, DECLARED =      YES,
@: AUTO-LOGOFF=STD, LINE25 =      STD, START-SEQ =      NO,
@: LMSG =      NO, LCASE =      NO, SAVE=      STD,
@: PAC-PREFIX= STD, BERID=      NO, CONECT=      OPNCON,
@: SNMP =      NO, USERID =      TSOS
@:/

```



OOMNIS also logs the user ID in the 25th line on the screen

```

@:/INF PARTNER,PAC=P10      "Test mode is inactive"
@: ZVT-ENTRY FOR PAC=      P10, TID=      27:
@: PAC =      P10, PTNNAME = $DIALOG, PRONAME = G133H207,
@: PID =      0049, TYPE =      TIAM, APPNAME = $$$$$$$$,
@: STATUS = INACT, ROUTE =      DIR, OPASS =      NO,
@: MTAB =      STD, EXIT =      STD, HOLD =      STD,
@: KPAC =      NO, CHANGE =      YES, HCOPY =      STD,
@: CLASS =      OUT, COLOUR =      GREEN, PROTOCOL = OMNIS,
@: LOGGING = STD, LPASS =      NO, DECLARED =      YES,
@: AUTO-LOGOFF=STD, LINE25 =      STD, START-SEQ =      NO,
@: LMSG =      NO, LCASE =      NO, SAVE=      STD,
@: PAC-PREFIX= STD, BERID=      NO, CONECT=      OPNCON,
@: SNMP =      NO, USERID = TSOS
@:/

@:/INF PARTNER,PID=22      "Test mode is active"
@: ZVT-ENTRY FOR PID=0022:
@: TABID =      PTN, TYPE =      41, LINKD =      00070E6C,
@: LINKP =      000B102C, LINKWS =      007FFFFFF, ZVTID =      0022,
@: TAC =      P1 , TACL =      0002, TACANZ =      0000,
@: RPBADR =      007FFFFFF, PUFFADR =      007FFFFFF, WORKADR =      007FFFFFF,
@: AID =      00000000, CID =      00000000,
@: PTNNAME = $DIALOG , PRONAME = V214H101, CREF =      00000000,
@: L25TAC =      $$$$, SSHTYPE =      00, SSHEDIT =      04002000,
@: MTAB =      STD , EXIT =      STD, OPASS =      NO,
@: LOGANF =      00000000, LOGEND =      00000000, LOGPTR =      00000000,
@: LOGPTR$ =      0000, TRNPTR$ =      0000, TRNPTR =      00000000,
@: TRNTAC =      $$$$, TERMTYPE =      2D, ACKANZ =      00,
@: FLAG0-3 =      00000000, FLAG4-7 =      00000400, FLAG8-11 =00000060
@: FLAG12-15 =00101000, FLAG16-19=00000000, FLAG20-23=00000000,
@: VTCFCT =      001, VTCFCVT =      0001, VTCRET =      00000000,
@: VTCLEN =      032, VTCINM =      L, VTCOUT, =      L,
@: VTCMODE =      L, VTCHC =      N, VTCBEL, =      N,
@: VTCNLG =      N, VTCRBYT =      40, VTCRIN, =      N,
@: VTCLOC =      N, VTCOHD =      N, VTCCTR =      ,
@: VTCIHD =      N, VTCLOW =      N, VTCSPIN =      N,
@: VTCFC =      N, VTCHOM =      N, VTCCNA =      ,
@:/

@:/INF TERMINAL,ACTIV=ALL
@: TID  TERMNAME/PRONAME  TYPE STATUS ROUTE KPAC USER
@:   60 MF311085/V214V111 TERM ACT   DIR   NO
@:   54 IVR11005/V214V111 TERM ACT   DIR  K12
@:   23 MF311149/V214V111 TERM DECL  DIR   NO
@:   19 MF311037/V214V111 TERM DECL  DIR   NO
@:/

```



```

@:/INF TERMINAL,TID=OWN "Test mode is inactive"
@: ZVT-ENTRY FOR TID= 27:
@: PAC = STD, PTNNAME = IVR11005, PRONAME = V214V101,
@: TID = 27, TYPE = TERM, ADM = YES,
@: STATUS = ACT, ROUTE = DIR, OPASS = NONE,
@: MTAB = STD, EXIT = STD, HOLD = STD,
@: KPAC = NO, CHANGE = YES, HCOPI = STD,
@: ACK = STD, LISTENING = STD, COLOUR = GREEN,
@: LOGGING = YES, BERID = NONE, DECLARED = YES,
@: CALL-KEY = STD, BREAK-KEY = STD, CALL-INF = STD,
@: DIS-MOD = SYSTEM, CONNECT = LOGON, OPNCON = STD,
@: PACANZ = 0000, INPUT-LOGGING=SEND, OUTPUT-LOGGING=SEND,
@: AUTO-LOGOFF= *NO, LINE25 = STD, USER= ,
@: EXCL-PARTNER= STD, SAVE = STD, REPLY = STD,
@: INPUT-SAVE= F4, USER-PROT= OMNIS, TESTMODE= NO,
@: SNMP = NO, TRANS-PROT= NEA, HCY-FORM= STD,
@:/INPUT-SAVE-NR=STD.
@:/

@:/INF TERMINAL,TID=23 "Test mode is active"
@: ZVT-ENTRY FOR TID= 23:
@: TABID = DST, TYPE = 81, LINKKD = 007FFFFFF,
@: LINKP = 000B10EC, LINKWS = 007FFFFFF, ZVTID = 23,
@: TAC = @ , TACL = 0001, TACANZ = 0000,
@: RPBADR = 007FFFFFF, PUFFADR = 0004B2D0, WORKADR = 007FFFFFF,
@: AID = 00000000, CID = 004E0200,
@: PTNNAME = MF311085, PRONAME = V214V111, CREF = 00000000,
@: L25TAC = $$$$, SSHTYPE = 00, SSHEDIT = 04002000,
@: MTAB = STD, EXIT = STD, OPASS = NO,
@: LOGANF = 000103F9, LOGEND = 00010A92, LOGPTR = 00000000,
@: LOGPTR$ = 0000, TRNPTR$ = 0000, TRNPTR = 00000000,
@: TRNTAC = $$$$, TERMTYPE = 2D, ACKANZ = 00,
@: FLAG0-3 = 00011000, FLAG4-7 = 000004C0, FLAG8-11 =00801360,
@: FLAG12-15 =00100000, FLAG16-19=00000000, FLAG20-23=00000000,
@: VTCFCT = 001, VTCFCVT = 0001, VTCRET = 00000000,
@: VTCLEN = 032, VTCINM = L, VTCOUT, = L,
@: VTCMODE = L, VTCHC = N, VTCBEL, = N,
@: VTCNLG = N, VTCRBYT = 40, VTCRIN, = N,
@: VTCLOC = N, VTCOHD = N, VTCCTR = ,
@: VTCIHD = N, VTCLOW = N, VTCSPIN = N,
@: VTCFC = N, VTCHOM = N, VTCCNA =
@:/

@:/INF MTAB
@: LIST OF ALL MTABS:
@: TAB4 TAB3 TAB2 TAB1
@:/

```

@:/INF MTAB,MTAB=TAB1

@: MTAB 'TAB1' CONTAINS FOLLOWING MESSAGE NUMBERS:

@: EXC0230 EXC0231 EXC0232 EXC0233

@:/

@:/INF MTAB,MTAB=ALL

@: MTAB 'TAB4' CONTAINS NO MESSAGE NUMBER.

@: MTAB 'TAB3' CONTAINS NO MESSAGE NUMBER.

@: MTAB 'TAB2' CONTAINS FOLLOWING MESSAGE NUMBERS:

@: EXC0233 EXC0234 EXC0235 EXC0212

@: MTAB 'TAB1' CONTAINS FOLLOWING MESSAGE NUMBERS:

@: EXC0230 EXC0231 EXC0232 EXC0233

@:/

@:/INF MTAB,MTAB=TAB1,MSG=ALL

§: MSG M-POS B D ID,INSERT,REPLY(S)

§: EXC0230 021 N N ID-POS=029,ID=10

§: REPLY=ANTWORT1

§: EXC0231 021 N N

§: EXC0232 021 N N

§: EXC0233 021 N Y ID-POS=029,ID=13

@: REPLY= ANTWORT4

@:/

@:/INF MUX,ACTIV=ALL

@: MID PTNNAME /PRONAME STATUS CONNECT LPASS SESSIONS AVAIL

@: 10 MUX1 /V214H101 ACT START NO 00020 NO

@: 11 UTMAPP1 /V214H303 ACT OPNCON YES 00025 NO

@: 13 UTMAPP2 /V214H303 LOSCON START NO 00000 NO

@:/

@:/INF MUX,MID=15 "Test mode is active"

@: ZVT-ENTRY FOR MID=15

@: TABID = MUX, TYPE = 30, ZVTID = 15,

@: LINKWS = 007FFFFFFF, LINKMUX = 007FFFFFFF, RPBADR = 007FFFFFFF,

@: PUFFADR = 007FFFFFFF, SESSADR = 000D3000, SESSANZ = 00000,

@: AID = 00320028, CID = 00580100, LPASS = YES,

@: LOGANF = 00000000, LOGEND = 00000000, LOGPTR = 00000000,

@: PTNNAME = MUXWIENA, PRONAME = V214H303,

@: FLAG0-3 = 00000000, FLAG4-7 = 000004A0, FLAG8-11 = 00000060,

@: FLAG12-15 = 00000000, FLAG16-19= 00000000

@:/

```

@:/INF PARTNER,PID=22      "Test mode is active"
@: ZVT-ENTRY FOR PID=22:
@: TABID =          PTN, TYP =          73, LINKD =    00070E6C,
@: LINKP =    000B102C, LINKWS =    007FFFFFF, ZVTID =          22,
@: TAC =          P1 , TACL =          0002, TACANZ =          0000,
@: RPBADR =    007FFFFFF, PUFFADR =    007FFFFFF, WORKADR =    007FFFFFF,
@: AID =          00000000 , CID =          00000000,
@: PTNNAME =    OMUTM11 , PRONAME = G133H207, CREF =    00000000,
@: L25TAC =          $$$$, SSHTYPE =          00, SSHEDIT =    04002000,
@: MTAB =          STD , EXIT =          STD, OPASS =          NO,
@: LOGANF =    00000000, LOGEND =    00000000, LOGPTR =    00000000,
@: LOGPTR$ =    0000, TRNPTR$ =    0000, TRNPTR =    00000000,
@: TRNTAC =          $$$$, TERMTYPE =          2D, ACKANZ =          00,
@: FLAG0-3 =    00000000, FLAG4-7 =    00000400, FLAG8-11 =00000060,
@: FLAG12-15 =00101000, FLAG16-19=00000000, FLAG20-23=00000000,
@: VTCFCT =          001, VTCFCVT =          0001, VTCRET =    00000000,
@: VTCLEN =          032, VTCINM =          L, VTCOUT, =          L,
@: VTCMODE =          L, VTCHC =          N, VTCBEL, =          N,
@: VTCNLG =          N, VTCRBYT =          40, VTCRIN, =          N,
@: VTCLOC =          N, VTCOHD =          N, VTCCTR =          ,
@: VTCIHD =          N, VTCLOW =          N, VTCSPIN =          N,
@: VTCFC =          N, VTCHOM =          , VTCCNA =          ,
@: MULTIPLEX SPECIFIC DATA:
@: TABID =          HMUX, ZVTID =          0023, LINKPTN =    000
@: LINKMUX =    00090B2C, OMSREF =    00020002, UTMREF =    002
@: MUXVERS =    0001, SEQOMS =          23, SEQUTM =
@: OPCODE =    USER-MSG, FLAG1-2 =          8000, STATCHAR1 =    012
@: STATCHAR2 =    00501809, SESSSTAT =          00, TACVG =    TRA
@: ACTTAC =    ACTTAC1, FORMAT =    FORMX1
@:/

@:/INF EXIT
@: EAC EID MODULES
@: EX01 63 EX1 EX2 EX3
@:/

@:/INF EXIT,EXIT=EX01
@: EXIT= EX01
@: TID: 22(A)
@: PID: 52(I) 51(I) 50(I) 49(I) 48(I)
@:/

@:/INF G
@: GRP PAC-LIST
@: G1 T01 T02 T03 T04 T05 T06 T07 T08 T09 T10
@: T11 T12 T13 T14
@: G2 P1 CONS T07 T11 T16 T18
@:/

```

```
@:/INF 0,ACTIV=ALL
@: OAC ID PTNNAME /PRONAME STATUS
@: @OM1 10 OM50Y /V214H101 ACT
@: @SCH 10 OMSKUR /V214H101 INACT
@:/

@:/INF OMNIS,OID=10
@: OAC = @SCH, PTNNAME =OMSKUR , PRONAME =V214H101,
@: OID = 10, STATUS = INACT, CONNECT = LOGON,
@: TIME = 00000000, LPASS = NO, OPNCON = FREE
@:/

@:/INF HCOPY,ACTIV=ALL
@: HAC HID PTNNAME /PRONAME STATUS INOP
@: X3 13 MF311160/V214V111 INACT NONE
@: X2 12 MF311150/V214V111 ACTIV NONE
@: X1 11 MF311100/V214V111 INACT NONE
@:/

@:/INF HCOPY,HID=13
@: ASSIGNMENTS TO HARDCOPY (HID= 12,HAC=X2 ):
@: PID 15
@: TID 14
@:/

@:/INF TRACE
@: TRACE-DATEI = SYS.DIA.OMNIS.02TD.1
@: CONNECTION-TRACE = YES
@: EXIT-TRACE = YES
@: TRANSPORT-TRACE = YES
@: SELECT-TERM = /
@: SELECT-HCOPY =
@: SELECT-MID =
@: SELECT-OMNIS =
```

## MDEF - Define message table

The *MDEF* command is used to administer the message tables. It can be used to generate and delete message tables and to generate or modify message table entries.



Depending on the setting made with the *PRIORITY* command, the *MDEF* command may be reserved for use by the administrator.

Operation	Operands
MD[EF]	table-name  [,A[CTION]= $\left\{ \begin{array}{l} \text{M[ODIFY]} \\ \text{C[REATE]} \\ \text{D[ELETE]} \end{array} \right\}$ ]  AP[ASS]= $\left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$  [,B[ELL]= $\left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\}$ ]  [,MSG= $\left\{ \begin{array}{l} \text{message-number} \\ \text{(message-number,position)} \end{array} \right\}$ ]  $\left\{ \begin{array}{l} \left[ ,\text{ID}=\left\{ \begin{array}{l} \text{string} \\ \text{(string,position)} \end{array} \right\} \right] \\ \left[ ,\text{IN[SERT]}=\text{(string,insert-number)} \right] \end{array} \right\}$  [,R[EPLY]= $\left\{ \begin{array}{l} \text{string} \\ \text{(string[,...])} \end{array} \right\}$ ]  [,D[ISPLAY]= $\left\{ \begin{array}{l} \text{Y[ES]} \\ \text{N[O]} \end{array} \right\}$ ]  

table-name	specifies the name of the message table to be processed. The formal syntax of <i>table-name</i> is identical to that of <i>pac</i> . The names N[O], Y[ES], and S[TD] are not permitted for <i>table-name</i> .
ACTION	specifies whether the message table is to be created, deleted, or modified.
CREATE	The message table <i>table-name</i> is to be created. No further operands are allowed.
DELETE	The message table <i>table-name</i> is to be deleted. No further operands are allowed.
MODIFY	A message table entry is to be created or modified. The operand <i>MSG</i> must be specified for <i>ACTION=MODIFY</i> . <i>MSG</i> (and possibly <i>ID</i> or <i>INSERT</i> ) serve to identify the BS2000 message, and thus the entry. If a table entry already exists with the <i>MSG</i> (and <i>ID</i> or <i>INSERT</i> if present) specified in the command, the operands <i>REPLY</i> and <i>DISPLAY</i> are adopted in the entry. Otherwise, a new table entry is created.
APASS	specifies the valid administration password. This operand is required only if the <i>MDEF</i> command is available only to the administrator, i.e. when <i>PRIORITY MDEF-AUTHORIZATION=ADM</i>
passwd	is the administration password.
?	After the command has been input, the administration password is to be prompted for by OMNIS in blanking mode.
BELL	specifies whether or not the audible alarm is triggered at the terminal during message output.
YES	The audible alarm is triggered at the terminal during message output.
NO	The audible alarm is not triggered at the terminal during message output.
MSG	specifies the number of the message for which the table entry is to be created or modified. If <i>ACTION=MODIFY</i> , <i>MSG</i> is mandatory.
message-number	is the number of the message, consisting of 1-7 freely selectable characters (other than blank and comma).
position	specifies the starting position of the desired message (default = 21). If the message is specified more than once with different positions, the last value specified for <i>position</i> is used. By default, <i>position</i> is set to the position at which the message key of console outputs is expected for UCON partners. It is thus not necessary to specify <i>position</i> for messages from UCON.

ID	<p>The <i>ID</i> operand specifies a string (within the message) which is to be checked in addition to the message number. This enables a number of different actions to be initiated in response to one message, depending on the message content.</p> <p>For a message table entry, you may only use either the <i>ID</i> operand or the <i>INSERT</i> operand. Within the message table, you may mix <i>ID</i> and <i>INSERT</i> specifications.</p>
string	specifies the character string which is to be compared. <i>string</i> may be up to 30 characters in length.
position	<p>specifies the position within the message at which <i>string</i> is to be searched for. <i>position</i> must be an integer between 1 and 127 and is interpreted as the displacement relative to the start of the output line (the message prefix, e.g. date and time of day, is included). The first character of the output line has a displacement of 0.</p> <p>If <i>position</i> is not specified, 29 is taken as the default value (this is the relative position of the second character after the 7-byte message number and is normally the start of the actual message text).</p> <p>If <i>ID</i> is not specified, no additional check is made of the message text; only the message number is used as the selection criterion.</p>
INSERT	<p>The <i>INSERT</i> operand specifies an insert string within the message which is to be checked in addition to the message number. This enables a number of different actions to be initiated in response to one message, depending on the message content.</p> <p>For a message table entry, you may only use either the <i>ID</i> operand or the <i>INSERT</i> operand. Within the message table, you may mix <i>ID</i> and <i>INSERT</i> specifications.</p>
string	specifies the character string which is to be compared. <i>string</i> may be up to 30 characters in length.
insert-number	specifies which number insert <i>string</i> denotes. <i>insert-number</i> is an integer between 0 and 14.
REPLY	<p>specifies a list of replies which OMNIS is to send to the originator of the message described by <i>MSG</i> and <i>ID</i>.</p> <p>If the message originator specifies a message job ID equal to "000" (column 7-9 in the message output format), an answer will automatically be sent to his/her TSN. If the message originator specifies a message job ID not equal to "000", this is no longer possible. The originator is then identified via the dummy parameter &amp;MSGADDR. &amp;MSGADDR has the content of column 2-9 of the console message, which is "sender-message job ID" (see the "Introduction to System Administration" [6]).</p>

The total length of all replies must not be greater than 70 (note that a one-byte length field must be included for each reply). The individual replies must be specified in a format that UCON can understand:

- Messages beginning with a '.' (period) are considered to be “true” replies. “True” replies are possible only for messages requiring a response.
- Messages beginning with a slash (/) are interpreted as BS2000 console commands.
- Messages beginning with an authorization name or key, or a device mnemonic, are displayed only at the console(s) so addressed.

The ampersand (&) must not occur within the reply as it is reserved for the detection of variables *&TSN*, *&VMP* and *&MSGADDR*.

Each *&TSN* character sequence occurring in a string is replaced by the current TSN (i.e. the one in the message prefix), e.g. *REPLY='/CAN &TSN'* terminates the process which has caused the message.

Each *&VMP* character sequence occurring in a string is replaced by the prefix of the corresponding virtual console. It is therefore also possible for virtual consoles to selectively answer messages.

Each *&MSGADDR* character sequence occurring in a string is replaced by the current “originator-message job ID” field.

See also [section “Automatic system operating through message tables” on page 111](#).

DISPLAY	specifies whether the message identified by <i>MSG</i> and <i>ID</i> is to be output at the terminal.
YES	The message is to be output (default). If <i>REPLY</i> is also specified, the automatically generated replies are also output on the terminal.
NO	Regardless of the specification for <i>REPLY</i> , neither messages nor replies are output.

Particular attention should be paid to the order of the *MDEF* commands, as this has a considerable influence on both the run time and the effects of a message table; see also [page 111](#).

When one *MTAB* entry is replaced by another (by means of an *MDEF* command with an existing message number and *ID*), the order of the message table entries remains unchanged.

Large messages tables are best defined in the startup file or in a command file.



Message table entries cannot subsequently be deleted. They can, if necessary, be neutralized with *MDEF name,MSG=message-number [,ID=id],D=Y*.



Message tables or entries can be examined using the command *INF MTAB*.

In the case of multi-line DADM messages, the *ID* must point to the first line. If *DISPLAY=NO*, the subsequent lines are also suppressed.

For virtual consoles only the operand *ID*, but not the operand *INSERT* is supported for the check of the message string.

### Examples

```
@: /INF MTAB
```

```
@: NO MTAB TO DISPLAY.
```

```
@: /MDEF MSTD,ACTION=CREATE
```

```
@: OMS0040 MESSAGE TABLE 'MSTD' CREATED
```

```
@: /MDEF MPRO,ACTION=CREATE
```

```
@: OMS0040 MESSAGE TABLE 'MPRO' CREATED
```

```
@: /INF MTAB
```

```
@: LIST OF ALL MTABS:
```

```
@: MPRO MSTD
```

```
@: /MDEF MSTD,MSG=BLS0519,D=N,ID=(' EDT' ,38),R='/MES TSN=&TSN, '*** $EDT ***''
```

```
@: /MDEF MSTD,MSG=BLS0519,D=N,ID=(' LMS' ,38),R='/MES TSN=&TSN, '*** $LMS ***''
```

```
@: /MDEF MSTD,MSG=BLS0519,D=N
```

```
@: /MDEF MPRO,MSG=JMS0154,D=Y,B=Y,IN=(' IDEFIX' ,2)
```

```
@: /MDEF MPRO,MSG=JMS0154,D=Y,B=Y,IN=(' OMSE' ,0)
```

```
@: /MDEF MSTD,MSG=EXC0420,D=N
```

```
@: /MDEF MTB1,MSG=EXC0736,D=Y,B=Y
```

```
@: /INF MT,MT=ALL,MSG=ALL
```

```
@: CONTENTS OF MTAB 'MPRO':
```

```
@: MSG      M-POS  B  D  ID,INSERT,REPLY(S)
```

```
@: JMS0154  021    Y  Y  INS-NR=02,INSERT=IDFIX
```

```
@: JMS0154  021    Y  Y  INS-NR=00,INSERT=OMSE
```

```
@: EXC0420  021    N  N
```

```
@: EXC0736  021    Y  Y
```

```
@: CONTENTS OF MTAB 'MSTD':
```

```
@: MSG      M-POS  B  D  ID,INSERT,REPLY(S)
```

```
@: BLS0519  021    N  N  ID-POS=038,ID=EDT
                        REPLY=/MES TSN=&TSN, '*** $EDT ***'
```

```
@: BLS0519  021    N  N  ID-POS=038,ID=LMS
                        REPLY=/MES TSN=&TSN, '*** $LMS ***'
```

```
@: BLS0519  021    N  N
```

```
@: /
```

## MESSAGE - Send messages

The *MESSAGE* command can be used to send messages to specific terminals, according to their authorization (administrator, user identified by name, user not identified by name):

- All users can send messages to the OMNIS administrator.
- The OMNIS administrator can send messages to all terminals.
- All users can send messages to a specific selected terminal (TID, TERMINAL).
- A user identified by name can send messages to all terminals which are registered under his or her name.

Operation	Operands
M[ESSAGE]	$\left\{ \begin{array}{l} \text{TID} = \left\{ \begin{array}{l} \text{OWN} \\ \text{num} \end{array} \right\} \\ \text{[T[ERMINAL] = ([ptr][,pro])][,U[SER] = user]} \\ \text{ALL} \\ \text{ADM} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{'message'} \\ \text{[,I[INPUT]=} \left\{ \begin{array}{l} \text{filename} \\ \text{library-name(member)} \end{array} \right\} \end{array} \right\}$ $\text{[,AP[ASS]=} \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} \text{]}$

**TID** specifies the TID of the terminal to which *message* or the contents of the file *filename* or *library-name(member)* are to be sent.

**TERMINAL** specifies the network address of the terminal to which you wish to send a message.

**ptrn** is the partner name of the terminal.

**pro** is the processor name of the terminal.

**USER** specifies the user to whom you wish to send a message.

**user** is the name of the user.

ALL	specifies that <i>message</i> or the contents of the file <i>filename</i> or <i>library-name(member)</i> are to be sent to all terminals (broadcast).
	<div style="display: flex; align-items: flex-start;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px; text-align: center; width: 20px; height: 20px; line-height: 20px;">i</div> <div> <p>The <i>ALL</i> operand is reserved for use by the OMNIS administrator.</p> <p>The message is also sent to all terminals connected to another OMNIS and connected via indirect connections to the local OMNIS.</p> </div> </div>
ADM	specifies that <i>message</i> or the contents of the file <i>filename</i> or <i>library-name(member)</i> are to be sent to all terminals declared as administrators (with <i>OPTION ADM=YES</i> or <i>DECLARE-TERMINAL ADM=YES</i> ).
message	is the text of the message to be sent. The message length may not exceed 80 characters. The text must be enclosed within apostrophes. Any apostrophes in the message text must be duplicated, e.g.: <i>MESSAGE ALL, 'Caution "TEST" started', AP='xxx'</i>
	<div style="display: flex; align-items: flex-start;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px; text-align: center; width: 20px; height: 20px; line-height: 20px;">i</div> <div> <p><i>message</i> may be specified only <b>after</b> the parameter for the destination (TID, TERMINAL, USER, ALL, ADM).</p> </div> </div>
INPUT	<p>specifies a SAM or ISAM file (or a type-S LMS library member) whose contents are to be sent. In both cases, the file contents are read in via SYSDTA. <i>INPUT</i> is accepted only if the terminal is already known as an administrator (with <i>OPTION ADM=YES</i> or <i>DECLARE-TERMINAL ADM=YES</i>) or if the administration password is specified in the <i>MESSAGE</i> command. A user may specify <i>INPUT</i> only if <i>SECURITY LEVEL=LOW</i>. The administrator may also specify <i>INPUT</i> if <i>SECURITY LEVEL=MEDIUM</i>, provided OMNIS is not running under TSOS.</p> <p>In order to prevent all users connected to OMNIS from being locked out until the message file has been read in, OMNIS rejects any attempt to access a message file which has been migrated from HSMS (Hierarchical Storage Management System).</p>
APASS	specifies the valid administration password. If <i>SECURITYLEVEL=HIGH</i> , the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	is the administration password.
?	After the command has been input, the administration password is to be prompted for by OMNIS in blanking mode.

The terminal(s) to which *message* or the contents of the file *filename* (or library member *library-name(member)*) are sent is (are) informed of the initiator of the *MESSAGE* command (*message* OMS0015 or OMS0016)

By specifying the TID in the *message* OMS0015 or OMS0016, a terminal can respond specifically to a message.



During processing of a *MESSAGE* command, OMNIS accepts no other messages from terminals or partners. *MES ALL* may temporarily block OMNIS operations for all users and, if there is a large number of terminals, lead to memory bottlenecks in the data communication system.

### Examples

```
@: /MES 15, 'HELLO, HOW ARE YOU ?'
@: /
```

```
@: /MES (PART1,PRO1), 'HELLO, HOW' 'S LIFE ?'
@: /
```

```
@: /MESSAGE ALL, 'OMNIS IS CLOSING DOWN IN 10 MINUTES!!!'
@: OMS0016 BROADCAST FROM TERMINAL 61:
@: OMNIS IS CLOSING DOWN IN 10 MINUTES!!!
@: /
```

```
@: /M ALL, I=OMNIS.BULLETIN, AP=X'66FAC7F9'
@: OMS0016 BROADCAST FROM TERMINAL 61:
@: *****
@: * OMNIS is closing down in 10 minutes ! *
@: *****
@: /
```



## OCCUPY - Take over existing connection

The *OCCUPY* command can be used to take over partners of a terminal from another terminal. Either specific partners of a terminal or all partners of a terminal can be taken over, or the partners of a unique user who is identified by name.

The connections to Kerberos-authenticated partners (previously chip card) can only be taken over from the same Kerberos-authenticated users when *SECURITY LEVEL=LOW*, regardless of the administration authorization of the terminal wishing to take over the connections. The principal is not checked

The *OCCUPY* command is executed if *SECURITY LEVEL≠LOW* only for unique users who are identified by name, regardless of the administration authorization of the terminal wishing to take over the partners.

No.	Operation	Operands
1	OCC[UPY]	$\left. \begin{array}{l} \text{PID} = \text{number} \\ \text{PAC} = \text{pac} \left\{ \begin{array}{l} \text{,TID} = \left\{ \begin{array}{l} \text{OWN} \\ \text{num} \end{array} \right\} \\ \text{[,T[ERMINAL]} = ([\text{ptn}][\text{,pro}]][\text{,U[SER]} = \text{user}] \end{array} \right\} \end{array} \right\}$ $\text{[,AP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$ $\text{[,NEW[PAC]} = \text{pac}$ $\text{[,OP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$
2	OCC[UPY]	$\left. \begin{array}{l} \text{TID} = \text{number} \\ \text{[,T[ERMINAL]} = ([\text{ptn}][\text{,pro}]][\text{,U[SER]} = \text{user}] \end{array} \right\}$ $\text{[,AP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$ $\text{[,OP[ASS]} = \left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}$

PID	specifies the PID of the partner which is to be taken over.
PAC	specifies the partner address code of the partner which is to be taken over by TID or TERMINAL.
TID	selects the terminal whose partners are to be taken over.
	 When a terminal is taken over, its groups and all its other attributes are taken over as well. Special case: The terminal whose partners you are taking over uses OMNIS-MENU, while your terminal was previously not using OMNIS-MENU. In this case after you enter <i>OCCUPY TID</i> the OMNIS-MENU login mask <i>OMFMAI</i> will be output on your original terminal.
OWN	Partners of your own terminal are to be taken over.
num	Partners of the terminal with the TID <i>number</i> (number) are to be taken over.
TERMINAL	is the network address of the terminal whose partners are to be selected.
ptnr	is the partner name of the terminal.
pro	is the processor name of the terminal.
USER	is the user whose partners are to be selected.
user	is the name of the user.
	 If multiple connections meet the search criteria you specify, OMNIS rejects the command and issues an <i>INF TERMINAL</i> command to enable you to repeat the command with the unique TID.
APASS	specifies the valid administration password. If <i>SECURITYLEVEL=HIGH</i> , the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	specifies the administration password directly.
?	After the command has been input, the administration password is to be requested by OMNIS in blanking mode.
NEWPAC	specifies the partner address code which will be used in future to address the partner. If <i>NEWPAC</i> is omitted, the partner address code of the partner is not changed. <i>NEWPAC</i> is mandatory if the terminal has already assigned the partner address code of the partner to be taken over elsewhere. <i>NEWPAC</i> is only interpreted if the PID is specified. If <i>OCCUPY</i> is used for a partner which is already assigned to the user's own terminal, <i>NEWPAC</i> can be used to change this partner's partner address code.

**OPASS** specifies the *OCCUPY* password that has been assigned with one of the commands *OPNCON*, *PARAM*, *DECLARE-PARTNER*, *HALT*, *OPTION* or *DECLARE-TERMINAL* to the partner or to the terminal which you wish to take over. If no administration authorization exists, *OPASS* must be specified, in which case only partners and terminals to which an *OPASS* has been assigned can be taken over.



The OMNIS administrator can take over partners (terminals) without knowing *OPASS* or can take over partners (terminals) for which no *OPASS* has been defined.

**passwd** is the *OCCUPY* password in the command.

**?** the *OCCUPY* password is requested in blanking mode.

When *OCCUPY* is used, care must be taken that the type of terminal from which the connection was set up and the type of terminal issuing the *OCCUPY* command are identical. If this is not the case, problems with message editing (message header, format control characters) may occur.

If a partner which uses the 8-bit code (=XHCS) is occupied from a terminal which uses the 7-bit code, errors will result.

If the terminal has already reached the maximum permitted number of PACs (see *SET* command), the *OCCUPY* command is rejected.

### Examples

```
@: /INF P
```

```
@: NO PARTNER TO DISPLAY.
```

```
@: /INF T,AP='XXX',AC=ALL
```

```
@: TID  TERMNAME /PRONAME  TYPE  STATUS  ROUTE  KPAC  USER
```

```
@: 22  MF111203 /V214V111  TERM  ACT    DIR    STD
```

```
@: 14  MN303028 /V214H303  TERM  ACT    DIR    STD
```

```
@: 14  MN303028 /V214H303  TERM  DECL  DIR    STD
```

```
@: /OCCUPY TID=22,OP='YYY'
```

```
@: OMS0072 OCCUPY COMMAND COMPLETED (PAC=AAA,PID=23)
```

```
@: /INF P
```

```
@: PAC  PID  PTNNAME /PRONAME  TYPE  STATUS  ROUTE  KPAC  TID= 12
```

```
@: AAA  23  $DIALOG /V214H303  TIAM  ACT    DIR    NO
```

```
@: /
```

## OPNCON - Open connection to a partner

The *OPNCON* command sets up a connection to a partner.

Operation	Operands
O[PNCON]	$\left. \begin{array}{l} \text{pac} \\ \text{gac} \\ *C \\ *D \\ *T \\ *U \\ * \end{array} \right\}$ <p>[,AP[PNAME]=appname]</p> $[,B[ERID]=\left\{ \begin{array}{l} \text{authorization[,password]} \\ ? \end{array} \right\}]$ $[,CL[ASS]=\left\{ \begin{array}{l} \text{O[UTPUT]} \\ \text{S[AVE]} \\ \text{D[ELETE]} \end{array} \right\}]$ $[,CODE=\left\{ \begin{array}{l} \text{S[YS CODE]} \\ \text{B[INARY]} \end{array} \right\}]$ $[,COL[OR]=\left\{ \begin{array}{l} \text{N[O]} \\ \text{B[LUE]} \\ \text{C[YAN]} \\ \text{G[REEN]} \\ \text{M[AGENTA]} \\ \text{R[ED]} \\ \text{W[HITE]} \\ \text{Y[ELLOW]} \end{array} \right\}]$ $[,CP[ASS]=\left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\}]$



Operation	Operands
	$[,EX[IT]=\left\{ \begin{array}{l} \underline{S[TD]} \\ \underline{N[O]} \\ \text{eac} \end{array} \right\} ]$
	$[,HC[OPY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ \underline{N[O]} \\ \underline{Y[ES]} \\ \text{hac} \end{array} \right\} ]$
	$[,HO[LD]=\left\{ \begin{array}{l} \underline{S[TD]} \\ \underline{N[O]} \\ \underline{Y[ES]} \end{array} \right\} ]$
	$[,KP[AC]=\left\{ \begin{array}{l} \text{num} \\ \underline{N[O]} \end{array} \right\} ]$
	$[,LC[ASE]=\left\{ \begin{array}{l} \underline{Y[ES]} \\ \underline{N[O]} \end{array} \right\} ]$
	$[,LIN[E25]=\left\{ \begin{array}{l} \underline{S[TD]} \\ \underline{N[O]} \\ \underline{Y[ES]} \\ \underline{O[MNIS]} \end{array} \right\} ]$
	$[,LO[GGING]=\left\{ \begin{array}{l} \underline{S[TD]} \\ \underline{N[O]} \\ \underline{Y[ES]} \end{array} \right\} ]$
	$[,LP[ASS]=\left\{ \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} ]$

Operation	Operands
	$[,MT[AB]=\left\{\begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{name} \end{array}\right\}]$ $[,OP[ASS]=\left\{\begin{array}{l} \text{'passwd'} \\ NONE \\ ? \end{array}\right\}]$ $[,PAC=\left\{\begin{array}{l} \underline{S[TD]} \\ L[INE] \\ P[REFIX] \\ N[O] \end{array}\right\}]$ $[,PR[ONAME]=\left\{\begin{array}{l} \text{prname} \\ *ANY \end{array}\right\}]$ $[,PROT[OCOL] = \left\{\begin{array}{l} \underline{O[MNIS]} \\ D[SSIM] \end{array}\right\}]$ $[,PT[NNAME]=\text{ptnname}]$ $[,RO[UTE]=\left\{\begin{array}{l} \underline{D[IRECT]} \\ I[NDIRECT] \\ M[UX] \end{array}\right\}]$ $[,SA[VE] = \left\{\begin{array}{l} \underline{SC[REEN]} \\ P[KEY] \\ A[LL] \\ N[O] \\ \underline{ST[D]} \end{array}\right\}]$ $[,SN[MP]=\left\{\begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array}\right\}]$

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

- pac Partner address code of the declared partner to which the connection is to be opened, or the partner address code to be assigned to the undeclared partner. This PAC may not already have been assigned to another partner or another partner group. Partner address codes must be unique within a terminal.
- gac Connections are to be established to all partners in the group with the address code *gac* as long as they have not yet been set up.
- \*C,\*D,\*T,\*U,\* Connections are to be established to all declared partners of the type UCON (\*C), DCAM (\*D), TIAM (\*T) or UTM (\*U), or to all declared partners (\*), as long as they do not yet exist.
- APPNAME defines the name of the “simulated terminal“ with which OMNIS sets up the connection to the partner. OMNIS is “known“ to the partner by this name.  
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.
- BERID specifies the authorization and password generated in the SKP. The operand *BERID* 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)
  - passwd Password generated in the SKP (1-8 characters, to be specified in the form *c'password'*).
  - ? Specifies that OMNIS is to request the password in blanking mode.

---

CLASS	defines the class of message to which the partner is to be assigned.
OUTPUT	The partner is to be allocated the class OUTPUT, i.e. all messages from the partner are displayed immediately at the terminal.
DELETE	The partner is to be allocated the class DELETE, i.e. all messages from the partner are deleted, and not displayed at the terminal.
SAVE	The partner is to be allocated the class SAVE, i.e. messages from the partner are not displayed immediately, but written to the logging file, where they can be read by paging.
CODE	is evaluated only if TYPE=DCAM; it specifies how the partner sends data:
SYSCODE	Data is sent in EBCDIC.
BINARY	Data is sent binary form.
COLOR	specifies the color in which the partner's messages are to be output:
NO	OMNIS is not to 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.
CPASS	(alternative: <i>LPASS</i> ) defines a connection password to be used by OMNIS on setting up a connection to the partner. Its use depends on the type of partner; a connection password is not needed for TIAM, for instance. If both <i>CPASS</i> and <i>LPASS</i> are specified in one command, the password specified last is used.
passwd	is the password (1-4 characters) for connection setup.
?	specifies that OMNIS is to request this password in blanking mode.
EXIT	defines whether or not an exit routine is used.
NO	The partner's message traffic is not to proceed via an exit routine.
eac	The partner's message traffic is to be passed through the routines of the exit group named <i>eac</i> .

STD	Messages are passed to an exit routine on a terminal-specific or OMNIS-global basis ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
HCOPY	defines message output on the printer assigned to the terminal.
NO	The partner's messages are to be displayed at the terminal only.
YES	The partner's messages are to be displayed and printed simultaneously.
hac	The partner's messages are to be printed on the printer with the address code <i>hac</i> .
STD	Printer output is defined for each terminal individually ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
HOLD	defines the behavior of OMNIS after the connection to the terminal has been lost. If <i>SECURITYLEVEL</i> ≠ <i>LOW</i> , specification of the <i>HOLD</i> operand in the <i>OPNCON</i> , <i>OPTION</i> , <i>PARAM</i> and <i>SET</i> commands is rejected with the message OMS0191.
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 specification <i>HOLD</i> = <i>YES</i> is accepted only if <i>OPASS</i> is also specified or is already defined (with <i>DECLARE-PARTNER</i> or <i>PARAM</i> ).
	Connections to Kerberos-authenticated partners (previously chip card) are always cleared down if the connection to the terminal is lost.
	If <i>SECURITYLEVEL</i> ≠ <i>LOW</i> , partners are not switched to the HOLD state.
STD	The behavior after a loss of connection is defined individually for each terminal ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
KPAC	defines a K key to be assigned to the partner. This key is used to switch between various partners in formatted mode.
number	is the number of this K key in the range 1 to 14. With UTM partners, this key triggers a <i>KDCLAST</i> . This applies to both direct and indirect connections to UTM partners but not to multiplex connections: if <i>ROUTE</i> = <i>MUX</i> , the K key defined with <i>KPAC</i> triggers a format restart. With non-UTM partners, it simulates the terminal being switched off and on again, followed by input of K3.
NO	No K key is defined.

LCASE	specifies whether lowercase letters are 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 also passed on to the partner.
NO	Lowercase letters are not passed on to the partner.
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). <i>LINE25</i> is not evaluated for terminals of type 8160.
STD	The use of the 25th line on the screen is defined on a terminal-specific basis ( <i>OPTION</i> , <i>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.
YES	Partner information is written in the in 25th line on the screen. OMNIS generates the 25th line on the screen <b>only</b> on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will not be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, for instance from formatted to line mode and, for the 9763 terminal, as a result of a status query, for instance on <i>LOGON</i> or <i>START-EDT</i> .
OMNIS	Partner information is written in the in 25th line on the screen. OMNIS generates the 25th line on the screen on changing partners <b>and</b> whenever this line is modified by the set partner (at the time of the next output).
LMSG	See CMSG.
LOGGING	controls the logging in the logging file of messages sent to the partner and/or received from the partner.
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 individually for each terminal individually ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
LPASS	See CPASS.

MTAB	defines a message table with which the partner's output is to be processed. The <i>MTAB</i> entry is relevant only for UCON partners.
NO	The partner's output is not to be processed with any message table.
name	The partner's output is to be processed with the message table <i>name</i> .
STD	Processing of the partner's output with a message table is defined individually for each terminal ( <i>OPTION, DECLARE-TERMINAL</i> ).
OPASS	defines an <i>OCCUPY</i> password. This is needed for taking over partners from another terminal with the <i>OCCUPY</i> command.
passwd	is the <i>OCCUPY</i> password which must be specified in an <i>OCCUPY</i> command in order to take over the partner.
NONE	If <i>OPASS=NONE</i> is specified, no <i>OPASS</i> is defined.
?	specifies that OMNIS is to request the <i>OCCUPY</i> password in blanking mode.
PAC	defines the output of the partner address code for line mode. The PAC may not already have been assigned to another partner or another partner group.
PREFIX	The PAC is to be displayed in the same line as the partner's message (as a prefix).
LINE	The PAC is to be displayed in a line on its own (together with the partner's name and type) before the message itself.
NO	The PAC is not to be displayed.
STD	Output of the partner address code is defined on a terminal-specific or OMNIS-global basis ( <i>OPTION, DECLARE-TERMINAL</i> ).
PRONAME	specifies (together with <i>PTNNAME</i> ) the symbolic network address of the partner. If <i>PRONAME</i> is not specified, OMNIS assumes that the partner is in the same processor in which OMNIS is loaded.
*ANY	<p><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.</p> <p>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.</p>

PROTOCOL	defines the behavior of OMNIS with respect to a DCAM application. The default is <i>OMNIS</i> .
OMNIS	OMNIS behaves like an application.
DSSIM	OMNIS behaves like a real terminal.
PTNNAME	specifies (together with <i>PRONAME</i> ) the symbolic network address of the partner. The default assumption is <i>\$DIALOG</i> for <i>TYP=TIAM</i> , and <i>\$CONSOLE</i> for <i>TYP=UCON</i> . <i>PTNNAME</i> is a mandatory operand for all other partner types.
ROUTE	specifies whether a direct, indirect or multiplex connection is to be set up to the partner.
DIRECT	A direct connection to the partner is to be set up.
INDIRECT	An indirect connection to the partner is to be set up (i.e. via an OMNIS present in the target computer).
MUX	A multiplex connection to the partner is to be set up (only where <i>TYP=UTM</i> ).
SAVE	specifies whether screen contents and/or P keys are saved on quitting the 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.)
PKEY	The P keys are saved individually for each application.
ALL	Both the screen contents and also the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
STD	The <i>OPTION</i> or <i>SET</i> value applies for <i>PAR</i> statements. The <i>DECLARE-TERMINAL</i> value applies for <i>DECLARE-PARTNER</i> statements. The default value for <i>SET</i> is <i>NO</i> .
SNMP	This operand enables the partner connection to be monitored via an SNMP terminal or OMNIS. Only one such SNMP terminal can be logged on via OMNIS at a time.
YES	The partner connection opened with <i>OPNCON</i> is monitored via the SNMP terminal or OMNIS.
NO	The partner connection opened with <i>OPNCON</i> is not monitored.



## START-SEQUENCE

*number* 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 with the start parameter *LENGTH TEXTKEY=n*. When specifying *START-SEQUENCE*, leading zeros can be omitted. If  $n=7$ , values 1 through 9999999 are permissible for *START-SEQUENCE*; if  $n=4$ , values between 1 and 9999 are permitted. The *START-SEQUENCE* operand can be specified in the *OPNCON* command only if the *OPNSTART=YES* operand has been specified in the *SET* command. Using the OMNISTC utility routine, text files can be converted to different key lengths (see the "Administration and Programming" manual [1]).

## TYP

defines the type of partner.

## TIAM

The partner is the system application \$DIALOG (timesharing).

## UCON

The partner is the system application \$CONSOLE (operating) or \$VMCONS (virtual console).

## DCAM

The partner is a DCAM or CMX application.

## UTM

The partner is a UTM application.

## CMMSG

(alternative: *LMSG*) defines a connection message to be sent to the partner by OMNIS on establishing the connection. The *CMMSG/LMSG* operand is not a genuine keyword operand and must always be the last operand in the operand list.

## connection-message

is partner-specific and can be up to 80 characters long; it must be specified in the case of UCON partners and may contain, for example, the prefix "\$CONSOLE", the authorization name and the authorization password (see also [page 61](#)).

## ?

Specifies that OMNIS is to prompt for the connection message in blanking mode.

Which operands must (may) be specified and which settings are effective depends on the partner type, the declaration and the settings made with the *PRIORITY* command, see the following section.

### Dependency of the operands on the 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

For partners declared with *CHANGE=NO*, only the operand *pac* may be specified; specifying any other operand(s) leads to an error message.

For declared terminals, the values (if any) in the configuration file are used for operands which are not specified. If no values are specified in the configuration file, and for dynamic connections, the defaults underlined in the command description are used.

In addition to this, the settings of the operands *EXIT* and *LOGGING* defined with the *PRIORITY* command apply with the highest priority.

The specifications *CLASS*, *COLOR*, *EXIT*, *HCOPY*, *HOLD*, *KPAC*, *LINE25*, *LOGGING*, *MTAB*, *OPASS* and *PAC* can be modified at any time with the *PARAM* command (but only, in the case of declared partners, if they have been declared with *CHANGE=YES*).



The *INF* command can be used to determine which connections currently exist for a terminal.

#### Examples

```
@: /INF P
```

```
@: NO PARTNER TO DISPLAY.
```

```
@: /OPNCON A
```

```
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=A, PID=0013, APPNAME=
OM@00013/HOST).
```

```
@: /
```

```
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY (PARTNER=$DIALOG/
HOST, PAC=A).
```

```
A:% JMS0150 INSTALLATION 7.580-D, BS2000 VERSION BS2H303 V090:
PLEASE LOGON.
```

```

A: /@:OPNCON B ,TYP=UCON,PRO=REMOTE,LM=CONS,X'56FF34081364'
@: OMS0012 'OPNCON' COMMAND ACCEPTED(PAC=B, PID=17, APPNAME=
      OM@00017/HOST).

@: /
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY (PARTNER=$CONSOLE/
      REMOTE, PAC=B).
B:%CONS 00.083314 % CONS CONNECTED

@:OPNCON C ,TYP=UTM,APP=XXX,PTN=UTMANW
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=C, PID=17, APPNAME=
      XXX/HOST).

@: /
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY (PARTNER=UTMANW/
      HOST, PAC=C).
K000 CONNECTED WITH APPLICATION UTMANW, PLEASE KDCSIGN

@:OPNCON D ,TYP=DCAM,PTN=DCAMANW,LP=X'11443322'
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=D, PID=18, APPNAME=
      OM@00019/HOST).

@: /
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY(PARTNER=DCAMANW/
      HOST, PAC=D).

@: /OPNCON U ,TYP=UCON,HOLD=YES,OP=?,LM=?
@: OMS0138 PLEASE ENTER OPASS:
@: ?          (blanked)
@: OMS0141 PLEASE ENTER LMSG:
@: ?          (blanked)
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=U,PID=30, APPNAME=
      OM@00030/HOST).

@: /
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY (PARTNER=$CONSOLE/
      HOST, PAC=U).

@: /INF P
@: PAC  PID  PTNNAME /PRONAME  TYPE STATUS  ROUTE  KPAC  TID=12
@: A   13   $DIALOG /HOST      TIAM ACT   DIR    NO
@: B   17   $CONSOLE/HOST     UCON ACT   DIR    NO
@: C   27   UTMANW  /REMOTE     UTM  ACT   DIR    NO
@: D   28   DCAMANW /HOST      DCAM ACT   DIR    NO
@: U   30   $DIALOG /HOST     UCON ACT   DIR    NO
@: /

```

## OPTION - Define terminal-specific attributes

The *OPTION* command can be used to set values which apply to all the partners of a terminal or to the terminal itself (see also the *SET* and *PARAM* commands for the differences).



The default values (which are shown underlined) are the settings for terminals that are not predefined.

When *OPTION* is entered for a terminal, the applicable values for all operands which are not explicitly specified remain unchanged.


Operation	Operands
OPT[ION]	$[,ACK=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ \underline{L[INE]} \end{array} \right\}]$ $[,ADM=\left\{ \begin{array}{l} Y[ES] \\ \underline{N[O]} \end{array} \right\}]$ $[,AP[ASS]=\left\{ \begin{array}{l} 'passwd' \\ ? \end{array} \right\}]$ $[,B[REAK]-K[EY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ num \end{array} \right\}]$ $[,C[ALL]-I[NF]=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array} \right\}]$ $[,C[ALL]-K[EY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ num \end{array} \right\}]$

Operation	Operands
	$[,COL[OR]=\left. \begin{array}{l} N[O] \\ B[LUE] \\ C[YAN] \\ G[REEN] \\ M[AGENTA] \\ R[ED] \\ W[HITE] \\ Y[ELLOW] \end{array} \right\} ]$ $[,D[ISPLAY]-M[ODE]=\left. \begin{array}{l} S[TD] \\ SY[STEM] \\ O[MNIS] \\ M[IXED] \end{array} \right\} ]$ $[,E[XCLUSIVE]-P[ARTNER]=\left. \begin{array}{l} N[O] \\ Y[ES] \\ S[TD] \end{array} \right\} ]$ $[,EX[IT]=\left. \begin{array}{l} S[TD] \\ N[O] \\ eac \end{array} \right\} ]$ $[,HC[OPY]=\left. \begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \\ hac \\ (D[EVICE]=devname,K[EY]=number \\ ,F[ORM]=STD   <form-name>) \end{array} \right\} ]$ $[,HO[LD]=\left. \begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \\ A[UTO] \end{array} \right\} ]$

Operation	Operands
	$[,I[INPUT]-L[OGGING]=\left. \begin{array}{l} \underline{S[TD]} \\ R[EC] \\ \\ SE[ND] \\ B[OTH] \end{array} \right\} ]$
	$[,I[INPUT]-S[AVE]=\left. \begin{array}{l} [K]num1..14 \\ Fnum1..24 \\ ([K]num1..14[,qty1..16]) \\ (Fnum1..24[,qty1..16]) \\ \underline{S[TD]} \\ N[O] \end{array} \right\} ]$
	$[,KP[AC]=\left. \begin{array}{l} \underline{S[TD]} \\ num \\ N[O] \end{array} \right\} ]$
	$[,LIN[E25]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ O[MNIS] \end{array} \right\} ]$
	$[,LIS[TENING]=\left. \begin{array}{l} \underline{N[O]} \\ hac \end{array} \right\} ]$
	$[,LO[GGING]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array} \right\} ]$
	$[,MT[AB]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ name \end{array} \right\} ]$
	$[,OP[ASS]=\left. \begin{array}{l} 'passwd' \\ NONE \\ ? \end{array} \right\} ]$

Operation	Operands
	$[,O[UTPUT]-L[OGGING]=\left. \begin{array}{l} S[TD] \\ R[EC] \\ SE[ND] \\ B[OTH] \end{array} \right\}]$ $[,PAC=\left. \begin{array}{l} S[TD] \\ N[O] \\ L[INE] \\ P[REFIX] \end{array} \right\}]$ $[,R[EPLY]=\left. \begin{array}{l} A[LL] \\ R[ESTRICTED] \\ S[TD] \end{array} \right\}]$ $[,SA[VE]=\left. \begin{array}{l} SC[REEN] \\ P[KEY] \\ A[LL] \\ N[O] \\ S[TD] \end{array} \right\}]$ $[,TESTMODE=\left. \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}]$

- ACK defines screen overflow control.
- NO Screen overflow control is deactivated; the messages from the partners are output asynchronously at the terminal.
- YES Screen overflow control is activated; messages from the partners are output at the terminal until there is a message which would clear the current screen contents or until the screen is full. In these cases the short message ACK is output and the system waits until an input is made by the user.
- STD Message flow control is not defined specifically for the terminal; the specification made in the SET command and valid globally in OMNIS applies.

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 the command authorization.
YES	All commands (including administrator's) may be entered at the terminal.
	Only the administrator may specify <i>ADM=YES</i> . In this case, <i>APASS</i> must also be specified. If <i>SECURITY LEVEL=HIGH</i> is set, the <i>APASS</i> operand is rejected with the message OMS0190. <i>SECURITY LEVEL=ADM</i> can therefore be entered only on the BS2000 console in this case.
NO	No administrator commands may be entered at the terminal.
APASS	specifies the valid administration password. <i>APASS</i> is required if <i>ADM=YES</i> is specified. If <i>SECURITY LEVEL=HIGH</i> is set, the <i>APASS</i> operand is rejected with the message OMS0190.
passwd	specifies the administration password directly.
?	After the command has been input, the administration password is to be requested by OMNIS in blanking mode.
BREAK-KEY	defines a K key which is to generate a <i>BREAK</i> for the \$DIALOG partner to which the last input was sent.
num	The value range for <i>num</i> (number) is 1-14.
NO	No <i>BREAK-KEY</i> is defined.
STD	The value defined globally for <i>BREAK-KEY</i> using the <i>SET</i> command applies.
CALL-INF	specifies whether or not the CALL status is to be indicated by a message.
YES	CALL status is to be indicated by a message.
NO	CALL status is not to be indicated by a message.
STD	The value specified using the <i>SET</i> command applies.



CALL-KEY	defines a K key which is to generate a CALL.
num	is the K key which is to generate a CALL. The value range for num is 1-14.
NO	No <i>CALL-KEY</i> is defined.
STD	The <i>CALL-KEY</i> OMNIS-wide defined in the <i>SET</i> command applies.
COLOR	specifies the color in which OMNIS messages are to be output:
NO	OMNIS is not to 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.
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.
OMNIS	The entire message is output in the partner's color.
MIXED	The message prefix is output in the partner's color, and the remainder in accordance with system conventions.
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 <i>SET</i> value for OMNIS applies.
YES	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. All messages requesting inputs are announced (by means of message <i>OMS0215</i> in the 25th screen line). Only the first output message is announced in each case (by means of message <i>OMS0215</i> in the 25th screen line).

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.


When *EXCLUSIVE-PARTNER=YES* is specified, all partners behave as if *CLASS=OUTPUT*, regardless of their *CLASS* value.

EXIT	defines whether or not an exit group is used. An <i>EXIT</i> specification is relevant only for the messages of those partners for which <i>EXIT=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
NO	The messages are not to be passed to an exit routine of an exit group.
eac	The terminal's message traffic is to be passed through the exit group <i>eac</i> .
STD	The passing of messages to an exit routine is defined globally ( <i>SET</i> ).
HCOPY	controls the output of messages on a terminal printer (hard copy) or an RSO printer. The hardcopy behavior defined by <i>HCOPY</i> is only relevant for partners for which <i>HCOPY=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
YES	All outputs from the partners are to be printed out on the locally connected terminal printer at the same time as they are displayed at the terminal.
NO	Outputs from the partner are displayed on the terminal only; printer output is suppressed even if a hard copy is explicitly requested.
hac	The terminal's messages are to be output on the directly connected printer with the address code <i>hac</i> .
STD	Output to the terminal printer is defined globally ( <i>SET</i> ).
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:
	$S. \left. \begin{array}{l} user \\ devname \end{array} \right\} .date.time.tid$
	where the user name <i>user</i> is needed when using OMNIS-MENU, otherwise the printer name <i>devname</i> . <i>date</i> and <i>time</i> refer to the time of file opening.
KEY	By specifying <i>number</i> you define a K key which you can use to initiate a screen dump and printout on the printer <i>devname</i> .


FORM	<p>This operand is for setting the form for the printer. By specifying <i>STD</i> or <i>form-name</i> (max. 6 characters), you can specify which format of the print form is 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 switch is made from the specified printer to a directly connected printer with <i>HCOPY=hac</i>, or the printer output is suppressed with <i>HCOPY=STD</i>, the form name is automatically set to the standard form <i>STD</i>.</p>
HOLD	<p>controls the way in which OMNIS behaves if it detects that the connection to the terminal has been lost (e.g. failure of a communication computer on the transmission path). The response to a lost connection as defined by <i>HOLD</i> is only relevant for partners for which <i>HOLD=STD</i> is specified (<i>OPNCON</i>, <i>PARAM</i>, <i>DECLARE-PARTNER</i>).</p>
NO	<p>If the connection to the terminal is lost, all open connections to the terminal's partners are closed (implicit <i>CLSCON</i>).</p>
YES	<p>If the connection to the terminal is lost, the connections to the terminal's partners are not closed, but are retained. The partners are not automatically taken over when the terminal is again connected to OMNIS; explicit <i>OCCUPY</i> commands are required. <i>HOLD=YES</i> is accepted only if <i>OPASS</i> is also specified or if <i>OPASS</i> is already defined (with <i>DECLARE-TERMINAL</i> or with a preceding <i>OPTION</i>).</p> <p>The connections to Kerberos-authenticated partners (previously chip card) can only be held by Kerberos-authenticated users when <i>SECURITY LEVEL=LOW</i>.</p> <p>In the case of a <b>non-unique user</b> and <i>SECURITY LEVEL≠LOW</i>, the specification <i>HOLD</i> is ignored; <i>HOLD=NO</i> always applies. Whether or not a user is unique is checked by OMNIS-MENU.</p>
AUTO	<p>If the connection to the terminal is lost, the connections to the terminal's partners are not closed, but are retained. The partners are automatically taken over when the terminal is again connected to OMNIS; no <i>OCCUPY</i> is required. The settings made with <i>SET</i> and <i>PRIORITY</i> should be noted.</p> <p>The connections to Kerberos-authenticated partners (previously chip card) can only be held by Kerberos-authenticated users when <i>SECURITY LEVEL=LOW</i>.</p> <p>In the case of a <b>non-unique user</b> and <i>SECURITY LEVEL≠LOW</i>, the specification <i>HOLD</i> is ignored; <i>HOLD=NO</i> always applies. Whether or not a user is unique is checked by OMNIS-MENU.</p>

STD	The response to the loss of the connection to the terminal is defined globally ( <i>SET</i> ).
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. If <i>LOGGING=NO</i> , no data is logged. If <i>LOGGING=STD</i> , the <i>LOGGING</i> setting specified in <i>SET</i> is evaluated.
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 using 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 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 for the first time but no output quantity specified, the default value 1 also applies. If the K or F key is only 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 reasons of security. If the K key is pressed for a 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 with which a switch to OMNIS command mode can be forced, even when a format is currently 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 using the <i>SET</i> command is applies.
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). <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 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.
YES	Partner information is written in the in 25th line on the screen. OMNIS generates the 25th line on the screen <b>only</b> on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will not be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, for instance from formatted to line mode and, for the 9763 terminal, as a result of a status query, for instance on <i>LOGON</i> or <i>START-EDT</i> .
OMNIS	Partner information is written in the in 25th line on the screen. OMNIS generates the 25th line on the screen on changing partners <b>and</b> whenever this line is modified by the set partner (at the time of the next output).
LISTENING	controls dialog output on another, "listening" terminal (see also <a href="#">page 120</a> ).
NO	Output is restricted to the current terminal.
hac	Dialog output appears both on the current terminal and simultaneously on the terminal identified by <i>hac</i> .

LOGGING	controls the logging in the logging file of messages passing via OMNIS. <i>LOGGING</i> is only relevant for messages to/from partners for which <i>LOGGING=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
NO	The messages are not to be written to the logging file. If <i>SECURITY LEVEL≠LOW</i> , all OMNIS commands and messages are written into the logging file, even if <i>LOGGING=NO</i> is specified.
YES	The messages are to be written to the logging file.
STD	Terminal-specific logging is not specified.
	If the logging priority <i>PAR-OPT-SET</i> is set (see <a href="#">page 170</a> ), logging behavior is specified globally (see the <i>SET</i> command). If the logging priority <i>SET-PAR-OPT</i> is set, the default value <i>NO</i> applies.
MTAB	defines a message table with which all UCON outputs are to be processed (message suppression, automatic message answering). <i>MTAB</i> is only relevant for UCON partners for which <i>MTAB=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
NO	UCON outputs are not to be processed via a message table.
name	UCON outputs are to be processed via the message table <i>name</i> . The specified message table must have been generated beforehand with the <i>MDEF</i> command.
STD	The processing of UCON outputs via a message table is defined globally ( <i>SET</i> ).
OPASS	defines an <i>OCCUPY</i> password. Assignment of an <i>OCCUPY</i> password is a prerequisite for taking over a connection with the <i>OCCUPY</i> command.
passwd	is the <i>OCCUPY</i> password.
NONE	The OPASS is deleted.
?	OMNIS requests the <i>OCCUPY</i> password in blanking mode.
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. If <i>LOGGING=NO</i> , no data is logged. If <i>LOGGING=STD</i> , the <i>LOGGING</i> setting specified in <i>SET</i> is evaluated.
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 for <i>OUTPUT-LOGGING</i> set with the <i>SET</i> command is used.
PAC	specifies whether and how OMNIS is to prefix line mode messages from partners with the partner address code. The partner address code handling defined by PAC is only relevant to partners for which <i>PAC=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
PREFIX	The partner address code is written in the same line (before the net message).
LINE	The partner address code is output in a separate line (before the line containing the net message).
NO	The partner address code is not output.
STD	The way in which the partner address code is to be output is defined globally ( <i>SET</i> ).
REPLY	controls replies to UCON applications.
ALL	OMNIS forwards all input to UCON.
RESTRICTED	OMNIS checks each reply to UCON to establish whether the corresponding query was output on the associated terminal. If so, OMNIS forwards the reply to UCON. If not, it rejects the reply and issues message <i>OMS0236</i> .
STD	The value defined in the <i>SET</i> command applies.
SAVE	specifies whether screen contents and/or P keys are saved on quitting the 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.)
PKEY	The P keys are saved individually for each application.
ALL	Both the screen contents and also the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
STD	The <i>OPTION</i> or <i>SET</i> value applies for <i>PAR</i> statements. The <i>DECLARE-TERMINAL</i> value applies for <i>DECLARE-PARTNER</i> statements. The default value for <i>SET</i> is <i>NO</i> .

TESTMODE	Switches the test mode on or off.
YES	Test mode is switched on.
	<i>TESTMODE=YES</i> is reserved for use by the administrator. If <i>SECURITY LEVEL=HIGH</i> , test mode is not permitted and <i>TESTMODE=YES</i> is rejected with the message OMS0189.
NO	Test mode is switched off.

The *OPTION* command is permitted for a declared terminal only if the terminal has been declared with *CHANGE=YES*.

The settings of the operands *EXIT*, *HOLD* (if *HOLD=AUTO*) and *LOGGING* made with the *PRIORITY* command have the highest priority.



*Examples*

@: / **INF T,TID=OWN**

```
@: ZVT-ENTRY FOR PID= 14:
@: PAC=          STD, PTNNAME=      MN303028, PRONAME=      V214H303,
@: TID=          14, TYP=          TERM, ADM=          NO,
@: STATUS=      ACT, ROUTE=        DIR, OPASS=        NO,
@: MTAB=        STD, EXIT=         STD, HOLD=         STD,
@: KPAC=        STD, CHANGE=       YES, HCOPIY=       STD,
@: ACK=         STD, LISTENING=    STD, COLOUR=      YELLOW,
@: LOGGING=     STD, BERID=        NO, DECLARED=     YES,
@: CALL-KEY=    STD, BREAK-KEY=    STD, CALL-INF=    STD,
@: DIS-MOD=     SYSTEM, CONNECT=   LOGON, OPNCON=   STD,
@: PACANZ=      0000, INPUT-LOGGING= SEND, OUTPUT-LOGGING= SEND,
@: AUTO-LOGOFF= STD, LINE25=       STD, USER=       ,
@: EXCL-PARTNER= STD, SAVE =       STD, REPLY =      STD,
@: INPUT-SAVE=  F4, USER-PROT=    OMNIS, TESTMODE=  NO,
@: SNMP =       NO, TRANS-PROT=   NEA, HCY-FORM=    STD.
@: /
```

@: / **OPTION HC=YES,KPAC=5**

@: / **INF T,TID=OWN**

```
@: ZVT-ENTRY FOR PID= 14:
@: PAC=          STD, PTNNAME=      MN303028, PRONAME=      V214H303,
@: TID=          14, TYP=          TERM, ADM=          NO,
@: STATUS=      ACT, ROUTE=        DIR, OPASS=        NO,
@: MTAB=        STD, EXIT=         STD, HOLD=         STD,
@: KPAC=        K5, CHANGE=       YES, HCOPIY=       YES,
@: ACK=         STD, LISTENING=    STD, COLOUR=      YELLOW,
@: LOGGING=     STD, BERID=        NO, DECLARED=     YES,
@: CALL-KEY=    STD, BREAK-KEY=    STD, CALL-INF=    STD,
@: DIS-MOD=     SYSTEM, CONNECT=   LOGON, OPNCON=   STD,
@: PACANZ=      0000, INPUT-LOGGING= SEND, OUTPUT-LOGGING= SEND,
@: AUTO-LOGOFF= STD, LINE25=       STD, USER=       ,
@: EXCL-PARTNER= STD, SAVE =       STD, REPLY =      STD,
@: INPUT-SAVE=  F4, USER-PROT=    OMNIS, TESTMODE=  NO,
@: SNMP =       NO, TRANS-PROT=   NEA, HCY-FORM=    STD.
@: /
```

## PARAM - Define partner-specific attributes

The *PARAM* command can be used to define or modify the attributes of specific partners.



The default values (which are shown underlined>) are the settings for partners that are not predefined.

When *PARAM* is entered for a partner, the applicable values for all operands which are not explicitly specified remain unchanged.

Operation	Operands
PAR[AM]	$\left. \begin{array}{l} \text{pac} \\ \text{gac} \\ *C \\ *D \\ *T \\ *U \\ * \end{array} \right\}$ $[,CL[ASS]=\left\{ \begin{array}{l} \underline{O[UTPUT]} \\ D[ELETE] \\ S[AVE] \end{array} \right\}]$ $[,COL[OR]=\left\{ \begin{array}{l} \underline{N[O]} \\ B[LUE] \\ C[YAN] \\ G[REEN] \\ M[AGENTA] \\ R[ED] \\ W[HITE] \\ Y[ELLOW] \end{array} \right\}]$ $[,EX[IT]=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{eac} \end{array} \right\}]$ $[,HC[OPY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ \text{hac} \end{array} \right\}]$

Operation	Operands
	$[,HO[LD]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array} \right\} ]$
	$[,KP[AC]=\left. \begin{array}{l} \underline{N[O]} \\ \text{num} \end{array} \right\} ]$
	$[,LIN[E25]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ O[MNIS] \end{array} \right\} ]$
	$[,LO[GGING]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \end{array} \right\} ]$
	$[,MT[AB]=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ \text{name} \end{array} \right\} ]$
	$[,OP[ASS]=\left. \begin{array}{l} \text{'passwd'} \\ ? \end{array} \right\} ]$
	$[,PAC=\left. \begin{array}{l} \underline{S[TD]} \\ N[O] \\ L[INE] \\ P[REFIX] \end{array} \right\} ]$
	$[,SA[VE] = \left. \begin{array}{l} SC[REEN] \\ P[KEY] \\ A[LL] \\ N[O] \\ \underline{S[TD]} \end{array} \right\} ]$
	$[,SN[MP]=\left. \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\} ]$

---

pac	is the partner address code of the partner whose attributes are to be modified.
gac	The attributes of all partners in the group with the address code <i>gac</i> are to be modified.
*C,*D,*T,*U,*	The attributes of all partners of the type UCON (*C), DCAM (*D), TIAM (*T) or UTM (*U), or of all declared partners (*), are to be modified.
CLASS	specifies the output class (message flow control) to which the partner is to be assigned.
OUTPUT	Messages from partners in the OUTPUT class are output immediately at the terminal.
DELETE	Messages from partners in the DELETE class are not output at the terminal and are lost.
SAVE	Messages from partners in the SAVE class are not output immediately at the terminal but are written to the logging file and are only output when the partner is transferred from the SAVE class to a different class.
COLOR	specifies the color in which the partner's messages are to be output:
NO	OMNIS is not to 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.
EXIT	defines whether or not an exit routine is used.
NO	The partner's message traffic is not to proceed via an exit routine from an exit group.
eac	The partner's message traffic is to be passed through the routine of the exit group <i>eac</i> .
STD	Messages are passed to an exit routine on a terminal-specific basis ( <i>OPTION, DECLARE-TERMINAL</i> ).

HCOPY	controls the output of messages on a terminal printer (hard copy) or on an RSO printer as an independent station.
YES	All outputs from the partner are to be printed out on the locally connected terminal printer at the same time as they are displayed on the terminal.
NO	Messages from the partner are to be displayed on the terminal only; printer output is suppressed even if it is explicitly requested by the partner application.
hac	The partner's messages are to be printed out on the directly connected printer with the address code <i>hac</i> .
STD	The output to a printer is defined for each terminal individually ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
HOLD	controls the way in which OMNIS behaves if it detects that the connection to the terminal has been lost (e.g. failure of a communication computer on the transmission path).
NO	If the connection to the terminal is lost, the connection to the partner is to be closed as well (implicit <i>CLSCON</i> ).
YES	If the connection to the terminal is lost, the connection to the partner is not to be closed, but is to remain open. <i>HOLD=YES</i> is accepted only if <i>OPASS</i> is also specified or if <i>OPASS</i> is already defined (with <i>DECLARE-PARTNER</i> , <i>OPNCON</i> or a preceding <i>PARAM</i> ).
	The connections to Kerberos-authenticated partners (previously chip card) can only be held by Kerberos-authenticated users when <i>SECURITY LEVEL=LOW</i> .
	Partners do not switch to the <i>HOLD</i> state if <i>SECURITY LEVEL≠LOW</i> .
STD	The behavior when the connection to the terminal is lost is defined for each terminal individually ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
KPAC	defines a K key to be assigned to the partner as a partner address code and to be used to switch partners.
NO	No K key is defined.
num	is the number of this key in the range 1 to 14. With UTM partners, this key triggers a <i>KDCLAST</i> . This applies to both direct and indirect connections to UTM partners but not to multiplex connections. If <i>ROUTE=MUX</i> , the K key defined with <i>KPAC</i> triggers a format restart. With non-UTM partners, it simulates the terminal being switched off and on again, followed by input of K3.

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.
YES	Partner information is written in the in 25th line on the screen.  OMNIS generates the 25th line on the screen <b>only</b> on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will not be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, for instance from formatted to line mode and, for the 9763 terminal, as a result of a status query, for instance on <i>LOGON</i> or <i>START-EDT</i> .
OMNIS	Partner information is written in the in 25th line on the screen.  OMNIS generates the 25th line on the screen on changing partners <b>and</b> whenever this line is modified by the set partner (at the time of the next output).
LOGGING	controls the logging in the logging file of messages passing via OMNIS.
NO	The messages are not to be written to the logging file.
YES	The messages are to be written to the logging file.
STD	The behavior of OMNIS with regard to logging is defined for each terminal individually ( <i>OPTION, DECLARE-TERMINAL</i> ).
MTAB	defines a message table with which all outputs of the partner are to be processed (message suppression, automatic message answering). The <i>MTAB</i> operand is only relevant to UCON partners.
NO	Outputs are not to be processed via a message table.
name	Outputs are to be processed via the message table <i>name</i> . The specified message table must have been generated beforehand with the <i>MDEF</i> command.
STD	The processing of outputs via a message table is defined for each terminal individually ( <i>OPTION, DECLARE-TERMINAL</i> ).
OPASS	This operand allows an <i>OCCUPY</i> password to be assigned / modified / deleted. This password must (if declared) be specified in the <i>OCCUPY</i> command in order to take over a partner from another terminal.
passwd	is the <i>OCCUPY</i> password.

NONE	<i>OPASS</i> is deleted.
?	specifies that OMNIS is to request <i>OPASS</i> in blanking mode.
PAC	specifies whether and how OMNIS is to prefix line mode messages from partners with the partner address code.
PREFIX	The partner address code is written in the same line (before the net message). Some lines of the message may be shifted if this is done.
LINE	The partner address code is output in a separate line (before the line containing the net message).
NO	The partner address code is not output. It is normally not advisable to switch off output of the partner address codes for more than one partner, since the originator of a message cannot be identified if this is done.
STD	The way in which the partner address code is to be output is defined for each terminal individually ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
SAVE	specifies whether screen contents and/or P keys are saved on quitting the 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.)
PKEY	The P keys are saved individually for each application.
ALL	Both the screen contents and also the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
STD	The <i>OPTION</i> or <i>SET</i> value applies for <i>PAR</i> statements. The <i>DECLARE-TERMINAL</i> value applies for <i>DECLARE-PARTNER</i> statements. The default value for <i>SET</i> is <i>NO</i> .
SNMP	This operand enables the partner connection to be monitored via an SNMP terminal or OMNIS. Only one such SNMP terminal can be logged on via OMNIS at a time.
YES	The partner connection opened with <i>OPNCON</i> is monitored via the SNMP terminal or OMNIS.
NO	The partner connection opened with <i>OPNCON</i> is not monitored.

The *PARAM* command is permitted for a declared partner only if the partner has been declared with *CHANGE=YES*.

The settings of the operands *EXIT* and *LOGGING* made with the *PRIORITY* command have the highest priority.

*Examples*

@: /INF P,PID=0037

```
@: ZVT-ENTRY FOR PID= 37, TID= 27:
@: PAC= A, PTNNAME= $DIALOG, PRONAME= V214H303,
@: PID= 37, TYP= TIAM, APPNAME= MAN00037,
@: STATUS= ACT, ROUTE= DIR, OPASS= NO,
@: MTAB= STD, EXIT= STD, HOLD= STD,
@: KPAC= NO, CHANGE= YES, HCOPY= STD,
@: CLASS= OUT, COLOUR= YELLOW, PROTOCOL= OMNIS,
@: LOGGING= STD, LPASS= NO, DECLARED= NO,
@: AUTO-LOGOFF= STD, LINE25= STD, START-SEQ= NO,
@: LMSG = NO, LCASE = NO, SAVE= STD,
@: PAC-PREFIX= STD, BERID= NO, CONECT= OPNCON,
@: SNMP = NO.
```

@: /PARAM A,CLASS=D,OP='OPPWD',HOLD=YES

@: /INF P,PID=0037

```
@: ZVT-ENTRY FOR PID= 37, TID= 27:
@: PAC= A, PTNNAME= $DIALOG, PRONAME= V214H303,
@: PID= 37, TYP= TIAM, APPNAME= MAN00037,
@: STATUS= ACT, ROUTE= DIR, OPASS= YES,
@: MTAB= STD, EXIT= STD, HOLD= YES,
@: KPAC= NO, CHANGE= YES, HCOPY= STD,
@: CLASS= DEL, COLOUR= YELLOW, PROTOCOL= OMNIS,
@: LOGGING= STD, LPASS= NO, DECLARED= NO,
@: AUTO-LOGOFF= STD, LINE25= STD, START-SEQ= NO,
@: LMSG = NO, LCASE = NO, SAVE= STD,
@: PAC-PREFIX= STD, BERID= NO, CONECT= OPNCON,
@: SNMP = NO.
```

@: /INF P,PID=0019

```
@: ZVT-ENTRY FOR PID= 19, TID= 27:
@: PAC= UCON, PTNNAME= $CONSOLE, PRONAME= V214H303,
@: PID= 19, TYP= UCON, APPNAME= $$$$$$$$,
@: STATUS= INACT, ROUTE= DIR, OPASS= NO,
@: MTAB= STD, EXIT= STD, HOLD= STD,
@: KPAC= NO, CHANGE= YES, HCOPY= STD,
@: CLASS= OUT, COLOUR= YELLOW, PROTOCOL= OMNIS,
@: LOGGING= STD, LPASS= NO, DECLARED= YES,
@: AUTO-LOGOFF= STD, LINE25= STD, START-SEQ= NO,
@: LMSG = NO, LCASE = NO, SAVE= STD,
@: PAC-PREFIX= STD, BERID= NO, CONECT= OPNCON,
@: SNMP = NO.
@: /PARAM UCON,EXIT=EX1,LOGGING=YES,HC=YES
```



```
@:/INF P,PID=0019
@: ZVT-ENTRY FOR PID= 19, TID= 27:
@: PAC= UCON, PTNNAME= $CONSOLE, PRONAME= V214H303,
@: PID= 19, TYP= UCON, APPNAME= $$$$$$$$,
@: STATUS= INACT, ROUTE= DIR, OPASS= NO,
@: MTAB= STD, EXIT= EX1, HOLD= STD,
@: KPAC= NO, CHANGE= YES, HCOPY= YES,
@: CLASS= OUT, COLOUR= YELLOW, PROTOCOL= OMNIS,
@: LOGGING= YES, LPASS= NO, DECLARED= YES,
@: AUTO-LOGOFF= STD, LINE25= STD, START-SEQ= NO,
@: LMSG = NO, LCASE = NO, SAVE= STD,
@: PAC-PREFIX= STD, BERID= NO, CONECT= OPNCON,
@: SNMP = NO.
@:/
```

## PRIORITY - Define scope of commands

The *PRIORITY* command defines whether *PARAM* is to apply before *OPTION* before *SET* or *SET* before *PARAM* before *OPTION* in the *EXIT*, *HOLD* (if *HOLD=AUTO*) and *LOGGING* operands.

It can also be used to define whether the *EXIT* and *MDEF* commands are to be reserved for the OMNIS administrator or whether every OMNIS user can issue these commands.



The *PRIORITY* command is reserved for use by the administrator.

The default values (which are shown underlined> are the settings when OMNIS is first started.

On entering *PRIORITY*, the applicable values for all operands which are not explicitly specified remain unchanged.

Operation	Operands
PR[ <u>IORITY</u> ]	[,AP[ASS]= $\left. \begin{matrix} \text{'passwd'} \\ ? \end{matrix} \right\} ]$
	[,E[XIT]-A[UTHORIZATION]= $\left. \begin{matrix} \text{ADM} \\ \text{ALL} \end{matrix} \right\} ]$
	[E[XIT]-P[R <u>IORITY</u> ]= $\left. \begin{matrix} \text{P[AR-OPT-SET]} \\ \text{S[ET-PAR-OPT]} \end{matrix} \right\} ]$
	[,H[OLD]-P[R <u>IORITY</u> ]= $\left. \begin{matrix} \text{O[PT-SET]} \\ \text{S[ET-OPT]} \end{matrix} \right\} ]$
	[,L[OGGING]-P[R <u>IORITY</u> ]= $\left. \begin{matrix} \text{P[AR-OPT-SET]} \\ \text{S[ET-PAR-OPT]} \end{matrix} \right\} ]$
	[,M[DEF]-A[UTHORIZATION]= $\left. \begin{matrix} \text{ADM} \\ \text{ALL} \end{matrix} \right\} ]$

- APASS** specifies the valid administration password. This operand is mandatory if the terminal is not defined as administrator. (A terminal is defined as administrator with the *ADM=YES* operand in the *OPTION* or *DECLARE-TERMINAL* command.) If *SECURITY LEVEL=HIGH* is set, the *APASS* operand is rejected with the message OMS0190.
- The command can, therefore, only be entered from the BS2000 console in this case if *SECURITY LEVEL=HIGH* is set.
- passwd** specifies the administration password.
- ?** After the command has been input, the administration password is requested by OMNIS in blanking mode.
- EXIT-AUTHORIZATION** specifies whether the *EXIT* command is reserved for the administrator or whether it may be issued by all users.
- ADM** Only the administrator may issue the *EXIT* command.
- ALL** All users are permitted to issue the *EXIT* command.
- EXIT-PRIORITY** defines the priority of the *SET*, *OPTION* and *PARAM* commands with regard to the *EXIT* operand.
- PAR-OPT-SET** The default order of priority - *PARAM* before *OPTION* before *SET* - applies.
- SET-PAR-OPT** *SET* applies before *PARAM* before *OPTION*.
- HOLD-PRIORITY** defines the priority of the *SET* and *OPTION* commands with regard to the *HOLD=AUTO* operand.
- OPT-SET** The value specified for *HOLD=AUTO* at the terminal by the user applies. This is the default value.
- SET-OPT** The value specified by the administrator for *HOLD=AUTO* with *SET* applies.
- LOGGING-PRIORITY** defines the priority of the *SET*, *OPTION* and *PARAM* commands with regard to the *LOGGING* operand.
- PAR-OPT-SET** The default order of priority - *PARAM* before *OPTION* before *SET* - applies.
- SET-PAR-OPT** *SET* applies before *PARAM* before *OPTION*.
- The OMNIS administrator can use the OMNIS commands *SET LOGGING=NO* and *SET LOGGING-PRIORITY=SET-PAR-OPT* to prevent OMNIS commands and messages from being recorded in the logging file when the *SECURITY-LEVEL=LOW*.

MDEF-AUTHORIZATION

specifies whether the *MDEF* command is reserved for the administrator or whether it may be issued by all users.

ADM Only the administrator may issue the *MDEF* command.

ALL All users are permitted to issue the *MDEF* command.

*Examples*

```
@: /INF S
@: DSTMAX =          18, PTNMAX =          36, PACMAX =          36,
@: APASS =          YES, HOLD =          *NO, HCY-FORM = STD    ,
@: HCOPY =          *NO, LOGGING =          *NO, ACK =          *NO,
@: MTAB =          *NO, EXIT =          *NO, OPNCON =          *FREE,
@: BREAK-KEY =          *NO, CALL-KEY =          *NO, CALL-INF =          *NO,
@: PAC =          *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG =*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =          *NO,
@: EXIT-PRIO = PAR-OPT-SET, REPLY =          ALL, EXIT-AUTH = ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT =          OFF, MDEF-AUTH = ADM,
@: HOLD-PRIO = OPT-SET    , INPUT-SAVE = *NO, OPNSTART = NO,
@: EXCL-PARTNER = NO, SAVE =          *NO, RTIMER =          ON,
@: INPUT-SAVE-NUMBER = *NO.
@: /PRIORITY EXIT-PRIORITY=SET-PAR-OPT,AP='SCH'
@: /INF S
@: DSTMAX =          18, PTNMAX =          36, PACMAX =          36,
@: APASS =          YES, HOLD =          *NO, HCY-FORM = STD    ,
@: HCOPY =          *NO, LOGGING =          *NO, ACK =          *NO,
@: MTAB =          *NO, EXIT =          *NO, OPNCON =          *FREE,
@: BREAK-KEY =          *NO, CALL-KEY =          *NO, CALL-INF =          *NO,
@: PAC =          *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG =*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =          *NO,
@: EXIT-PRIO = SET-PAR-OPT, REPLY =          ALL, EXIT-AUTH = ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT =          OFF, MDEF-AUTH = ADM,
@: HOLD-PRIO = OPT-SET    , INPUT-SAVE = *NO, OPNSTART = NO,
@: EXCL-PARTNER = NO, SAVE =          *NO, RTIMER =          ON,
@: INPUT-SAVE-NUMBER = *NO.
```

## RESTART - Restart OMNIS-OMNIS connection/printer connection

The *RESTART* command activates an OMNIS-OMNIS connection or a printer connection after a line failure.

Operation	Operands
R[ESTART]	HCOPY=hac
R[ESTART]	OMNIS=oac

HCOPY specifies the HCOPY address code of the printer to be activated.

OMNIS specifies the OMNIS address code of the OMNIS-OMNIS connection to be activated.

### *Example*

```
@: /RESTART OMNIS=@OM3
```

## SET - Set operating parameters

The *SET* command can be used

- to set maximum values which may not be exceeded during operation of OMNIS (maximum number of partners, terminals or partners per terminal);
- to declare/modify the administration password;
- to redefine most of the default values set in OMNIS (see *PRIORITY*).



The *SET* command is reserved for use by the administrator.

The default values (which are shown underlined> are the settings when OMNIS is first started.

On entering *SET*, the applicable values for all operands which are not explicitly specified remain unchanged.

Operation	Operands
S[ET]	$[,ACK=\left\{ \begin{array}{l} \underline{S[TD]} \\ N[O] \\ Y[ES] \\ \underline{L[INE]} \end{array} \right\}]$ $[,AP[ASS]=\left\{ \begin{array}{l} 'passwd' \\ ? \end{array} \right\}]$ $[,AU[DIT]=\left\{ \begin{array}{l} ON \\ \underline{OF[F]} \end{array} \right\}]$ $[,B[REAK]-K[EY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ num \\ \underline{N[O]} \end{array} \right\}]$ $[,C[ALL]-I[NF]=\left\{ \begin{array}{l} \underline{S[TD]} \\ Y[ES] \\ \underline{N[O]} \end{array} \right\}]$ $[,C[ALL]-K[EY]=\left\{ \begin{array}{l} \underline{S[TD]} \\ num \\ \underline{N[O]} \end{array} \right\}]$

Operation	Operands
	<p data-bbox="397 215 833 378">[,D[ISPLAY]-M[ODE]=<math>\left. \begin{array}{l} \text{ST[<u>D</u>] } \\ \text{SY[<u>STEM</u>] } \\ \text{O[<u>MNIS</u>] } \\ \text{M[<u>IXED</u>] } \end{array} \right\} ]</math></p> <p data-bbox="397 416 615 450">[,DSTM[AX]=num]</p> <p data-bbox="397 495 873 588">[,E[XCLUSIVE]-P[ARTNER]=<math>\left. \begin{array}{l} \text{N[<u>O</u>] } \\ \text{Y[<u>ES</u>] } \\ \text{S[<u>TD</u>] } \end{array} \right\} ]</math></p> <p data-bbox="397 635 618 727">[,EX[IT]=<math>\left. \begin{array}{l} \text{S[<u>TD</u>] } \\ \text{N[<u>O</u>] } \\ \text{eac} \end{array} \right\} ]</math></p> <p data-bbox="397 772 1048 999">[,HC[OPY]=<math>\left. \begin{array}{l} \text{S[<u>TD</u>] } \\ \text{N[<u>O</u>] } \\ \text{Y[<u>ES</u>] } \\ \text{hac} \\ \text{(D[EVIC<u>E</u>]=devname} \\ \text{,K[EY]=number} \\ \text{,F[OR<u>M</u>]=STD   &lt;form-name1..6&gt;)} \end{array} \right\} ]</math></p> <p data-bbox="397 1041 658 1201">[,HO[LD]=<math>\left. \begin{array}{l} \text{S[<u>TD</u>] } \\ \text{N[<u>O</u>] } \\ \text{Y[<u>ES</u>] } \\ \text{A[<u>UTO</u>] } \end{array} \right\} ]</math></p> <p data-bbox="397 1243 813 1402">[,I[NPUT]-L[OGGING]=<math>\left. \begin{array}{l} \text{S[<u>TD</u>] } \\ \text{R[<u>EC</u>] } \\ \text{SE[<u>ND</u>] } \\ \text{B[<u>OTH</u>] } \end{array} \right\} ]</math></p>

Operation	Operands
	$[,I[INPUT]-S[AVE]= \left. \begin{array}{l} ([K]num1..14 \\ Fnum1..24 \\ ([K]num1..14[,qty1..16]) \\ (Fnum1..24[,qty1..16]) \\ S[TD] \\ N[O] \end{array} \right\} ]$
	$[,KPAC= \left. \begin{array}{l} S[TD] \\ num \\ N[O] \end{array} \right\} ]$
	$[,LIN[E25]= \left. \begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \\ O[MNIS] \end{array} \right\} ]$
	$[,LO[GGING]= \left. \begin{array}{l} S[TD] \\ N[O] \\ Y[ES] \end{array} \right\} ]$
	$[,MT[AB]= \left. \begin{array}{l} S[TD] \\ N[O] \\ name \end{array} \right\} ]$
	$[,NP[ASS]= \left. \begin{array}{l} 'passwd' \\ ? \end{array} \right\} ]$
	$[,OP[NCON]= \left. \begin{array}{l} S[TD] \\ D[ECLARED] \\ F[REE] \end{array} \right\} ]$
	$[,OPNS[TART]= \left. \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\} ]$



Operation	Operands
	<p>[,O[UTPUT]-L[OGGING]=<math>\left. \begin{array}{l} \text{S[TD]} \\ \text{R[EC]} \\ \text{SE[ND]} \\ \text{B[OTH]} \end{array} \right\} ]</math></p> <p>[,PAC=<math>\left. \begin{array}{l} \text{S[TD]} \\ \text{N[O]} \\ \text{L[INE]} \\ \text{P[REFIX]} \end{array} \right\} ]</math></p> <p>[,PACM[AX]=number]</p> <p>[,PTNM[AX]=number]</p> <p>[,R[EPLY]=<math>\left. \begin{array}{l} \text{A[LL]} \\ \text{R[ESTRICTED]} \\ \text{S[TD]} \end{array} \right\} ]</math></p> <p>[,RT[IMER]=<math>\left. \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} ]</math></p> <p>[,SA[VE] = <math>\left. \begin{array}{l} \text{SC[REEN]} \\ \text{P[KEY]} \\ \text{A[LL]} \\ \text{N[O]} \\ \text{S[TD]} \end{array} \right\} ]</math></p>

ACK	defines the screen overflow control. <i>ACK</i> is only relevant for terminals for which <i>ACK=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
NO	Screen overflow control is deactivated; the messages from the partners are output asynchronously at the terminal.
YES	Screen overflow control is activated; messages from the partners are output at the terminal until there is a message which would clear the current screen contents or until the screen is full. In these cases the short message <i>@:ACK</i> is output and the system waits until an input is made by the user.
STD	Message flow control is not defined; the default value <i>NO</i> applies.
LINE	The behavior of screen overflow control is brought into line with the behavior of <i>TIAM</i> . In contrast to <i>TIAM</i> , the <i>ACK=YES</i> setting results in <i>@:ACK</i> 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 <i>BLS0500</i> and <i>BLS0552</i> after the startup of the <i>EDT</i> program which immediately switches to format mode, the <i>ACK=LINE</i> setting enables <i>@:ACK</i> to be restricted to the overflow in line mode.
APASS	specifies the valid administration password (if one was previously declared with <i>NPASS</i> ). <i>APASS</i> is mandatory if the terminal is not defined as an administrator. If <i>SECURITY LEVEL=HIGH</i> is set, the <i>APASS</i> operand is rejected with the message <i>OMS0190</i> .  The command can, therefore, only be entered from the <i>BS2000</i> console in this case.
passwd	specifies the administration password.
?	After the command has been input, the administration password is to be requested by <i>OMNIS</i> in blanking mode.
AUDIT	specifies whether the audit trace is to be used to obtain diagnostic information. This is advisable only in the case of reproducible <i>OMNIS</i> errors. This information can be made available to the system diagnosis personnel only via an <i>OMNIS</i> user dump (see the <i>DUMP</i> command). Note that activating the audit trace will considerably reduce the performance of <i>OMNIS</i> (by a factor of 50)! Default value when <i>OMNIS</i> is started: <i>OFF</i> .
ON	The audit trace executes.
OFF	The audit trace does not execute.

BREAK-KEY	defines a K key which is to generate a <i>BREAK</i> for the \$DIALOG partner to which the last input was sent. Specification of <i>BREAK-KEY</i> is relevant only for terminals for which <i>BREAK-KEY=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
num	The value range for this number is 1-14.
NO	No <i>BREAK-KEY</i> is defined.
STD	No <i>BREAK-KEY</i> is defined; the default value <i>NO</i> applies.
CALL-INF	specifies whether or not CALL status is to be indicated by a message. The value set for <i>CALL-INF</i> with <i>SET</i> applies only to terminals for which <i>CALL-INF=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
YES	CALL status is to be indicated by a message.
NO	CALL status is not to be indicated by a message.
STD	The default value <i>NO</i> applies.
CALL-KEY	defines a K key which is to generate a <i>CALL</i> . The <i>CALL-KEY</i> defined with <i>SET</i> applies only to terminals for which <i>CALL-KEY=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
num	is the K key which is to produce a <i>CALL</i> . The value range for this number is 1-14.
NO	No <i>CALL-KEY</i> is defined.
STD	The default value <i>NO</i> applies.
DISPLAY-MODE	defines the colors for UCON outputs. Default value when OMNIS is started: <i>SYSTEM</i> . The <i>DISPLAY-MODE</i> specification is relevant only for terminals for which <i>DISPLAY-MODE=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
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.
OMNIS	The entire message is output in the partner's color.
MIXED	The message prefix is output in the partner's color, and the remainder in accordance with system conventions.
STD	The default value <i>SYSTEM</i> applies.

DSTMAX	defines the maximum number of terminals (in the range 0..32767) which can operate simultaneously with OMNIS. Default value when OMNIS is started: 18. Consoles used as terminals are included in the count of terminals; this must be taken into account when specifying DSTMAX. <i>DSTMAX=0</i> can be used to prevent the connection of further terminals to OMNIS. Existing connections to terminals are not affected by this specification.
EXCLUSIVE-PARTNER	controls the message flow.
NO	All messages relating to all partners are output on the terminal.
STD	The message flow control is not defined; the default value <i>NO</i> applies.
YES	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. All messages requesting inputs are announced, by means of message <i>OMS0215</i> in the 25th screen line. Only the first output message is announced in each case, by means of message <i>OMS0214</i> in the 25th screen line.  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.  When <i>EXCLUSIVE-PARTNER=YES</i> is specified, all partners behave as if <i>CLASS=OUTPUT</i> , regardless of their <i>CLASS</i> value.
EXIT	defines an exit routine. An <i>EXIT</i> specification is only relevant for the messages of terminals for which <i>EXIT=STD</i> is specified ( <i>OPTION, DECLARE-TERMINAL</i> ).
NO	The messages are not to be passed to an exit routine.
eac	The messages are to be passed to the exit group <i>eac</i> .
STD	The passing of messages to an exit routine is not defined; the default value <i>NO</i> applies.
HCOPY	controls the output of messages on a terminal printer (hard copy). The hardcopy behavior defined by <i>HCOPY</i> is only relevant for terminals for which <i>HCOPY=STD</i> is specified ( <i>OPTION, DECLARE-TERMINAL</i> ).
YES	All outputs from the partners are to be printed out on the locally connected terminal printer at the same time as they are displayed on the relevant terminal.
NO	Outputs from the partner are displayed on the terminal only.

hac	All messages are to be output on the directly connected printer with the address code <i>hac</i> .
STD	Output to the terminal printer is not defined; there is no printer output.
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 printer name <i>devname</i>. <i>date</i> and <i>time</i> refer to the time of file opening.</p>
KEY	By specifying <i>number</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 is for setting the form for the printer. By specifying <i>STD</i> or <i>form-name</i> (max. 6 characters), you can specify which format of the print form is 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 switch is made from the specified printer to a directly connected printer with <i>HCOPY=hac</i> , or the printer output is suppressed with <i>HCOPY=STD</i> , the form name is automatically set to the standard form <i>STD</i> .
HOLD	controls the way in which OMNIS behaves if it detects that the connection to the terminal has been lost (e.g. failure of a communication computer on the transmission path). The response in the event of a connection loss as defined with <i>HOLD</i> is only relevant for terminals for which <i>HOLD=STD</i> is specified ( <i>OPNCON</i> , <i>PARAM</i> , <i>DECLARE-PARTNER</i> ).
NO	If the connection to the terminal is lost, all open connections to the terminal's partners are closed (implicit <i>CLSCON</i> ).
YES	If the connection to the terminal is lost, the connections to the terminal's partners are not closed, but are retained. The partners are not automatically taken over when the terminal is again connected to OMNIS; explicit <i>OCCUPY</i> commands are required. <i>HOLD=YES</i> is only effective for partners for whom OMNIS can assign an <i>OPASS</i> .


---

	<p>The connections to Kerberos-authenticated partners (previously chip card) can only be held by Kerberos-authenticated users when <i>SECURITY LEVEL=LOW</i>.</p> <p>Partners do not switch to the HOLD state if <i>SECURITY LEVEL≠LOW</i>.</p>
AUTO	<p>If the connection to the terminal is lost, the connections to the terminal's partners are not closed, but are retained. The partners are automatically taken over when the terminal is again connected to OMNIS (no <i>OCCUPY</i> required).</p> <p>The connections to Kerberos-authenticated partners (previously chip card) can only be held by Kerberos-authenticated users when <i>SECURITY LEVEL=LOW</i>.</p> <p>Partners do not switch to the HOLD state if <i>SECURITY LEVEL≠LOW</i>.</p>
STD	<p>The response when a connection to a terminal is lost is not defined; the default value <i>NO</i> applies.</p>
INPUT-LOGGING	<p>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. If <i>LOGGING=NO</i>, no data is logged. If <i>LOGGING=STD</i>, the <i>LOGGING</i> setting specified in <i>SET</i> is evaluated. The value set for <i>INPUT-LOGGING</i> is (in addition to the condition <i>LOGGING=YES</i>) relevant only for terminals for which <i>INPUT-LOGGING=STD</i> is set (<i>OPTION, DECLARE-TERMINAL</i>).</p>
REC	<p>The messages are written to the logging file after being entered at the terminal.</p>
SEND	<p>The messages are written to the logging file before being sent to the partner.</p>
BOTH	<p>The messages are written to the logging file both when they are entered at the terminal and when they are sent to the partner.</p>
STD	<p>The messages are written to the logging file before being sent to the partner (<i>SEND</i>).</p>
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 maximum capacity (1000 characters), the oldest input is deleted. If no quantity is specified, the default value 1 applies. If a key is assigned for the first time but no output quantity specified, the default value 1 also applies. If the K or F key is only reassigned, without the quantity being specified, the quantity that was previously set continues to be valid.</p>

If the default value *STD* is set or the functionality is switched off with *NO*, the quantity that was previously specified is reset to the default value of 1.

LINE entries that are displayed in blanking mode are not buffered for reasons of security. If the K key is pressed for a 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 *SAVE=SCREEN* is set for the terminal, the saved screen cannot be restored if you return to the original partner. The screen is deleted.

[K] <i>num1..14</i>	is the number of the K key in the range 1..14.
<i>Fnum1..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 default value <i>NO</i> is used.
KPAC	defines a K key which can be used to force output of a line-oriented message when a format is displayed on the screen. The value set for <i>KPAC</i> is relevant only for the messages of terminals for which <i>KPAC=STD</i> is set ( <i>OPTION, DECLARE-TERMINAL</i> ).
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	No K key is used to switch to OMNIS command mode.
LINE25	controls how the 25th line on the screen is used for partner information (PAC, PID, partner/processor name, route, K key, user ID). The value set for <i>LINE25</i> is relevant only for the messages of terminals for which <i>LINE25=STD</i> is set ( <i>OPTION, DECLARE-TERMINAL</i> ).
STD	The behavior of OMNIS with regard to the 25th line on the screen is not defined; the default value <i>NO</i> applies.
NO	The 25th line on the screen is not used for partner information.
YES	Partner information is written in the 25th line on the screen.
	OMNIS generates the 25th line on the screen <b>only</b> on changing partners. If the 25th line of the screen is modified (i.e. deleted or overwritten) by the set partner, it will not be altered by OMNIS. In addition to these explicit reasons, the 25th line can also be deleted by changing the output mode, for instance from formatted to line mode and, for the 9763 terminal, as a result of a status query, for instance on <i>LOGON</i> or <i>START-EDT</i> .

OMNIS	Partner information is written in the in 25th line on the screen. OMNIS generates the 25th line on the screen on changing partners <b>and</b> whenever this line is modified by the set partner (at the time of the next output).
LOGGING	controls for the whole of OMNIS the logging in the logging file of messages passing via OMNIS.
NO	The messages are not to be written to the logging file. If <i>SECURITY LEVEL≠LOW</i> , OMNIS commands and messages are always written into the logging file, even if <i>LOGGING=NO</i> is specified.
YES	The messages are to be written to the logging file.
STD	The global logging behavior is not defined.
	 If the logging priority <i>PAR-OPT-SET</i> is set (see <a href="#">page 170</a> ), <i>NO</i> is taken as default. If the logging priority <i>SET-PAR-OPT</i> is set, logging behavior is defined on a terminal-specific basis.
MTAB	defines a message table with which all UCON outputs are to be processed (message suppression, automatic message answering). <i>MTAB</i> is only relevant for UCON partners on terminals for which <i>MTAB=STD</i> is specified ( <i>OPTION</i> , <i>DECLARE-TERMINAL</i> ).
NO	UCON outputs are not to be processed via a message table.
name	UCON outputs are to be processed via message table <i>name</i> . The specified message table must have been generated beforehand with the <i>MDEF</i> command.
STD	The processing of UCON outputs via a message table is not defined; the default value <i>NO</i> applies.
NPASS	defines the administration password.
passwd	specifies the new administration password.
?	After the command has been input, the new administration password is to be requested by OMNIS in blanking mode.
OPNCON	defines the freedom of connection.
DECLARED	OMNIS accepts connections only from declared terminals. Requests for connection from undeclared terminals are rejected.
FREE	OMNIS accepts all requests for connections, even from undeclared terminals.
STD	The connection freedom is undefined; the default value <i>FREE</i> applies.



OPNSTART	specifies whether the <i>START-SEQUENCE</i> operand is to be permitted in the <i>OPNCON</i> command. Default value: <i>NO</i> .
YES	The <i>START-SEQUENCE</i> operand is permitted in the <i>OPNCON</i> command.
NO	The <i>START-SEQUENCE</i> operand is not permitted in the <i>OPNCON</i> command.
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. If <i>LOGGING=NO</i> , no data is logged. The value set for <i>OUTPUT-LOGGING</i> is (in addition to the condition <i>LOGGING=YES</i> ) relevant only for terminals for which <i>OUTPUT-LOGGING=STD</i> is set ( <i>OPTION, DECLARE-TERMINAL</i> ).
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 messages are written to the logging file before being sent to the terminal ( <i>=SEND</i> ).
PAC	specifies whether and in what way OMNIS is to prefix line mode messages from partners with the partner address code. The value set for <i>PAC</i> is relevant only for messages of terminals for which <i>PAC=STD</i> is set ( <i>OPTION, DECLARE-TERMINAL</i> ).
LINE	The partner address code is output in a separate line (before the line containing the net message).
PREFIX	The partner address code is written in the same line (before the net message).
NO	The partner address code is not output.
STD	The way in which the partner address code is to be output is not defined. <i>PAC=PREFIX</i> applies.
PACMAX	defines the maximum number of partners (in the range 0..32767) to which a terminal may be connected at the same time. Default value when OMNIS is started: 36.

PTNMAX	defines the maximum number of partners (in the range 0..32767) which may be connected to OMNIS at the same time. Default value when OMNIS is started: 36. <i>PTNMAX=0</i> can be used to prevent new connections to partners. Existing connections to partners are not affected by this specification.
REPLY	controls replies to UCON applications.
ALL	OMNIS forwards all input to UCON.
RESTRICTED	OMNIS checks each reply to UCON to establish whether the corresponding query was output on the associated terminal. If so, OMNIS forwards the reply to UCON. If not, it rejects the reply and issues message <i>OMS0236</i> .
STD	<i>ALL</i> applies.
RTIMER	This parameter allows users to activate or deactivate the real timer. By default, the STXIT routine defined by OMNIS is called every 10 seconds. This routine triggers realtime-critical operations. If this mechanism is deactivated, none of the realtime-critical operations are carried out: <ul style="list-style-type: none"><li>– Cyclical calling of the EXIT routines after expiry of the <i>OMEOTIME</i> interval</li><li>– Monitoring of the setup and cleardown of indirect connections and multiplex partner connections</li><li>– Monitoring of DCAM memory bottlenecks for: HCOPY, terminal, OMNIS-to-OMNIS connection and MUX connections</li><li>– Monitoring of an OMNIS-to-OMNIS connection in the period during which message traffic must take place.</li></ul>
SAVE	specifies whether screen contents and/or P keys are saved on quitting the 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.)
PKEY	The P keys are saved individually for each application.
ALL	Both the screen contents and also the P keys are saved.
NO	Neither the screen contents nor the P keys are saved.
STD	The default value is <i>NO</i> .

The value specified in the *SET* command applies for the following operands unless some other value is specified with *DECLARE-TERMINAL* or *OPTION*:

ACK,	INPUT-LOGGING,
BREAK-KEY,	KPAC,
CALL-INF,	LINE25,
CALL-KEY,	LOGGING,
DISPLAY-MODE,	MTAB,
EXCLUSIVE-PARTNER,	OUTPUT-LOGGING,
EXIT,	PAC,
HCOPY,	REPLY,
HOLD,	SAVE.

In addition, the settings for the operands *EXIT*, *HOLD* (if *HOLD=AUTO*) and *LOGGING* set with the *PRIORITY* command have highest priority.

*Examples*

```

@:/INF S
@: DSTMAX =          18, PTNMAX =          36, PACMAX =          36,
@: APASS =           YES, HOLD =           *NO, HCY-FORM = STD   ,
@: HCOPY =           *NO, LOGGING =        *NO, ACK =           *NO,
@: MTAB =            *NO, EXIT =           *NO, OPNCON =        *FREE,
@: BREAK-KEY =       *NO, CALL-KEY =       *NO, CALL-INF =      *NO,
@: PAC =             *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG ==*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =            *NO,
@: EXIT-PRIO = SET-PAR-OPT, REPLY =        ALL, EXIT-AUTH =    ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT =        OFF, MDEF-AUTH =    ADM,
@: HOLD-PRIO = OPT-SET   , INPUT-SAVE =    *NO, OPNSTART =     NO,
@: EXCL-PARTNER =      NO, SAVE =          *NO. RTIMER =       ON,
@: INPUT-SAVE-NUMBER = *NO.
@:/SET DSTMAX=3,ACK=YES,HOLD=YES,AP='sch'
@:/SET NPASS='HOME',AP='sch'
@:/SET PACMAX=5,AP='HOME',LOGGING=YES
@:/INF S
@: DSTMAX =          3, PTNMAX =          36, PACMAX =          5,
@: APASS =           YES, HOLD =           YES, HCY-FORM = STD   ,
@: HCOPY =           *NO, LOGGING =        YES, ACK =           YES,
@: MTAB =            *NO, EXIT =           *NO, OPNCON =        *FREE,
@: BREAK-KEY =       *NO, CALL-KEY =       *NO, CALL-INF =      *NO,
@: PAC =             *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG ==*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =            *NO,
@: EXIT-PRIO = SET-PAR-OPT, REPLY =        ALL, EXIT-AUTH =    ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT =        OFF, MDEF-AUTH =    ADM,
@: HOLD-PRIO = OPT-SET   , INPUT-SAVE =    *NO, OPNSTART =     NO,
@: EXCL-PARTNER =      NO, SAVE =          *NO, RTIMER =       ON,
@: INPUT-SAVE-NUMBER = *NO.
@:/SET NPASS='SCH',AP='HOME'
@:/SET ACK=NO,AP='SCH'
@:/INF S
@: DSTMAX =          3, PTNMAX =          36, PACMAX =          5,
@: APASS =           YES, HOLD =           YES, HCY-FORM = STD   ,
@: HCOPY =           *NO, LOGGING =        YES, ACK =           NO,
@: MTAB =            *NO, EXIT =           *NO, OPNCON =        *FREE,
@: BREAK-KEY =       *NO, CALL-KEY =       *NO, CALL-INF =      *NO,
@: PAC =             *PREFIX, INPUT-LOG = *SEND, OUTPUT-LOG ==*SEND,
@: LINE25 =          *NO, DIS-MOD = *SYSTEM, KPAC =            *NO,
@: EXIT-PRIO = SET-PAR-OPT, REPLY =        ALL, EXIT-AUTH =    ADM,
@: LOGG-PRIO = PAR-OPT-SET, AUDIT =        OFF, MDEF-AUTH =    ADM,
@: HOLD-PRIO = OPT-SET   , INPUT-SAVE =    *NO, OPNSTART =     NO,
@: EXCL-PARTNER =      NO, SAVE =          *NO, RTIMER =       ON,
@: INPUT-SAVE-NUMBER = *NO.

```

## TRACE - Switch trace on/off

The *TRACE* command can be used to switch on and off the trace for connection, transport and exit routine trace entries. The trace data assists system service personnel and the administrator with error diagnosis.



The *TRACE* command is accepted only when test mode is active (*OPTION TESTMODE=YES*) and thus only from the OMNIS administrator if *SECURITY LEVEL=LOW* or *SECURITY LEVEL=MEDIUM*.

The default values (which are shown underlined) are only applicable when the *TRACE* is called for the first time (after starting OMNIS).

When *TRACE* is called again, the applicable values for all operands which are not explicitly specified remain unchanged.

Operation	Operands
TR[ACE]	C[ONNECTION]= $\left. \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}$ ,E[XIT]= $\left. \begin{array}{l} \underline{N[O]} \\ Y[ES] \end{array} \right\}$ ,T[RANSPORT]= $\left. \begin{array}{l} \underline{N[O]} \\ Y[ES] \\ S[ELECT] ( [TERM=( [ptn] [,pro] ) \\ [ ,HCOPY=hac] \\ [ ,MID=mid][ \\ [ ,OMNIS=oac] ) \end{array} \right\}$

- CONNECTION** specifies whether the trace is to be activated/deactivated for logging DCAM connections.
- YES** Activates the trace for all OMNIS objects, i.e. a trace entry is written to the trace file when the following events occur:
- connection cleardown, setup or loss
  - connection response by the transport system
  - connection information and modification
- NO** Deactivates the trace for logging DCAM connections.

EXIT	specifies whether the trace is to be activated/deactivated for logging exit routines.
YES	<p>Activates the trace, i.e. a trace entry is written to the trace file whenever the following events occur:</p> <ul style="list-style-type: none"> <li>– "OMNIS calls EXIT"</li> <li>– "EXIT calls OMNIS"</li> <li>– at every exchange of information between EXIT routines that pass parameters in a group</li> </ul>
NO	Deactivates the trace (logging of exit routines).
TRANSPORT	specifies whether the trace is to be activated/deactivated for logging DCAM messages.
YES	<p>Activates the trace, i.e. a trace entry is written to the trace file when the following events occur:</p> <ul style="list-style-type: none"> <li>– a message is sent via a DCAM connection</li> <li>– a message is received via a DCAM connection</li> </ul>
NO	Deactivates the trace for DCAM messages.
SELECT	The trace for logging DCAM messages is activated only for the specified OMNIS objects. A trace for DCAM messages at terminals ( <i>TERM</i> ), printers ( <i>HCOPY</i> ), MUX connections ( <i>MID</i> ) or OMNIS-OMNIS connections ( <i>OAC</i> ) can be chosen.
TERM	is the network address of the terminal whose connections are to be traced.
ptn	is the partner name of the terminal.
pro	is the processor name of the terminal.
HCOPY	With <i>hac</i> , you can specify the hardcopy address code of the printer for which a transport trace is to be activated.
MID	With <i>mid</i> , you can specify the ID number of the multiplex connection for which the trace is to be activated.
OMNIS	With <i>oac</i> , you can specify the OMNIS address code of the OMNIS-OMNIS connection for which a transport trace is to be activated.



The administrator can evaluate the trace file only if the trace file is closed, i.e. when all traces are deactivated. The trace file can be evaluated with the TEDDY program, which is also used for the trace files of the transport system (UTM, DCAM). This provides the administrator with a good general overview of the system status.

## TURN - Page in the dialog logging file

The *TURN* command is used to activate paging mode.

Operation	Operands
T[URN]	[pac]

*pac* specifies the partner address code of the partner in whose dialog the user wishes to page (partner-specific paging mode).

If *pac* is not specified, the user may page through the entire dialog recorded for this terminal (terminal-specific paging mode).

### Paging commands

A paging command is requested by the short message @:TURN. The paging commands are as follows:

- + pages one screen toward the end of the file
- pages one screen toward the start of the file
- + n pages n screen lines toward the end of the file
- n pages n screen lines toward the start of the file
- ++ pages to the end of the file
- pages to the start of the file
- 0 terminates paging mode.

It is only possible to page in the log of partners assigned to the current terminal.

While paging mode is activated, messages currently being received are not sent to the terminal, but are buffered.

If the logging file is switched (with the *CHANGELOG* command) while paging mode is active, OMNIS switches to command mode.

In terminal-specific paging mode, when examining messages of a partner taken over with *OCCUPY*, there is no access to messages which were written to the logging file before the *OCCUPY*. In partner-specific paging mode, on the other hand, it is possible to page through all of the partner's messages.

### Examples

@: /TURN

@: /TURN XY





---

## 10 OMNIS Messages

The purpose of messages is to inform users that actions which they have initiated have been completed (command acknowledgment) or to indicate to them any unexpected events which occur (e.g. loss of connection).

### 10.1 Message layout

The messages are output prefixed with the OMNIS partner address code (`@` from the local OMNIS or `oac` from an OMNIS connected via an OMNIS-OMNIS connection).

OMNIS distinguishes between standard messages and short messages.

Short messages are messages which consist of just a few characters (in addition to the partner address code). They have the following formats:

```
@:identifier  
oac:identifier
```

Standard messages, on the other hand, have a longer message text which provides users with brief information on the reason for the message. They have the following formats:

```
@: OMSnnnn text  
oac: OMSnnnn text
```

You can find the standard messages 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".

With standard messages the user can use *HELP* to obtain detailed information on the reason for the message and to ascertain what action to take (if any). All OMNIS messages have a date/time header.

The meaning of the different variables in the message formats is as follows:

@:	is the partner address code of the local OMNIS
oac	is the OMNIS address code of a partner OMNIS
OMS	is the string indicating an OMNIS message.
nnnn	is the message number.
text	is the message text.
identifier	is the identifier of the short message.

DCAM return codes can be recognized by their prefix *OMY*.

## 10.2 Short messages

@: /	indicates that the next input preceded by the partner address code @: must be a command. However, this command does not have to be entered immediately.  If a command input to OMNIS is not followed by a standard message, the short message @: / should be regarded as a positive acknowledgment of command execution.
@: *	indicates that the next input from the terminal must be a response to a previous message which requires a reply. The response must be entered immediately; no inputs addressed to partners of the terminal are accepted.
@: ACK	indicates that the screen is full or that an output is to follow which would clear the screen.  <i>ACK</i> is only output if screen overflow control is activated ( <i>ACK=YES</i> in the <i>OPTION</i> , <i>SET</i> or <i>DECLARE-TERMINAL</i> commands).
@: CALL(pac)	means that the terminal is in CALL state and no outputs are made until an input occurs.  <i>CALL</i> is only output when the CALL display is activated ( <i>CALL-INF=YES</i> in the <i>OPTION</i> , <i>SET</i> or <i>DECLARE-TERMINAL</i> commands).
@: TURN	indicates that the terminal is in page-turning mode and that OMNIS is expecting the next paging command:
+	page forward one screen
+n	page forward n screen lines
++	page forward to the final message
-	page back one screen
-n	page back n screen lines
--	page back to the first message
0	terminate paging mode

---

# Glossary

## address code

Code used for screen input and output to uniquely identify partners, partner groups, the local (@:) or another (@ABC:) OMNIS, a hardcopy unit or an exit group. A distinction is thus made between partner address codes (PAC), group address codes (GAC), OMNIS address codes (OAC), hardcopy address codes (HAC) and exit address codes (EAC).

## BREAK key

K key that causes a *BREAK* with the TIAM partner to whom the last input was sent (like the K2 key without OMNIS). The BREAK key is defined with the *BREAK-KEY* operand in the OMNIS *OPTION* or *SET* command, or the OMNISKD *DECLARE-TERMINAL* statement.

## CALL key

K key that interrupts output to the terminal until the next input by the user. The CALL key is defined with the *CALL-KEY* operand in the OMNIS *OPTION* or *SET* command, or the OMNISKD *DECLARE-TERMINAL* statement.

## configuration

The configuration represents the OMNIS-specific connection structure.

Declared configuration: The connections are declared offline with the OMNISKD utility.

Dynamic configuration: The connections are set up and cleared down dynamically (successively during operation by means of OMNIS commands).

## configuration file

The configuration file contains the configuration defined with the OMNISKD utility routine.

### **connections**

OMNIS distinguishes between connections to terminals, to partners, to printers and to another OMNIS.

Connections to partners are either direct or indirect.

Direct connections (normal case): The OMNIS to which the terminal is connected is directly connected to the communication partner in the other or its own computer.

Indirect connections: The connection to a partner is set up indirectly via an OMNIS running on another computer.

Multiplex connections (special case): These make it possible to have more than one session simultaneously via a single transport connection.

### **dialog log**

A record kept of specific messages or all messages in the logging file.

### **exit group**

Sum of all the exit routines (up to 14) that are assigned to a particular address code and activated simultaneously.

### **exit routine**

A subroutine created by the OMNIS user or standard subroutine supplied by the manufacturer for extending or modifying the functions implemented by OMNIS, which is non-resident and called by OMNIS.

### **logging file**

The logging file serves as the logging medium when logging is switched on.

### **master**

The OMNIS to which the terminal is connected is referred to as the “master” and the OMNIS in the partner processor as the “slave”. These terms refer to a single indirect connection. One OMNIS can be master for some indirect connections and slave for other indirect connections at one and the same time.

### **MESSAGE (OMNIS command)**

Message sent to specific terminals or to all terminals connected to OMNIS.

### **message answering**

The automatic answering of console messages using predefined responses.

### **message table**

Summary of message declarations for message answering and/or message suppression. An entry contains a message and the responses to the message (default response/suppression identifier).

**OMNIS key**

K key for switching from formatted mode to command mode. This key is defined with the *KPAC* operand in the *OMNIS SET* or *OPTION* command, or the *OMNISKD DECLARE-TERMINAL* statement.

**paging file**

The paging file is used to buffer messages in the event of a message bottleneck. When messages saved in this file are sent on, and therefore no longer needed here, they are overwritten in the next message bottleneck. When OMNIS is terminated, this file is deleted.

**partner**

OMNIS recognizes the following as partners of a terminal:

- the system application \$DIALOG (TIAM partner)
- the system application \$CONSOLE (UCON partner)
- the system application "virtual console" \$VMCONS (UCON partner)
- UTM applications
- DCAM and CMX applications

**partner group**

A number of partners can be grouped together to form a partner group. This partner group is addressed in commands and during message input by means of a group address code.

**partner key**

Predefined K key which allows the user to switch partners in format mode from one (e.g. UTM) application to another (e.g. DCAM or TIAM) application. Each partner can be assigned a special K key with the *KPAC* operand of either of the *OMNIS* commands *SET* or *OPTION* or of the *OMNISKD DECLARE-TERMINAL* statement.

**PROP (OMNIS-PROP)**

OMNIS exit routine for administering and controlling programmed operator actions which are formulated as administrator procedures.

**scope**

The difference between the *OMNIS* commands *SET*, *OPTION*, *PARAM* and *OPNCON* lies primarily in their scope. A distinction is made between global scope (valid for all *OMNIS* users), terminal-specific scope (valid for a terminal and its partners), partner-specific scope (valid only for one partner). Each category has a different priority.

**session**

One of several data streams over a transport connection, with a defined start and a defined end.

Special case UTM: communication association between terminal and UTM application. The session starts with the (logical) connection setup to the application and ends when the connection is cleared down.

**slave**

see “master”

**standard partner groups**

Partner groups which are set up and administered automatically by OMNIS. OMNIS provides predefined group address codes for the currently assigned partners of a terminal, for all UCON, DCAM, TIAM and UTM partners. The corresponding group address codes are: \*, \*C, \*D, \*T and \*U.

**startup file**

The startup file is a special command file which is processed automatically by OMNIS immediately after startup.

---

## Abbreviations

ACK	(Positive) acknowledgement
AID	Advanced Interactive Debugger
BCAM	Basic Communication Access Method
CMX	Communication Method in SINIX
DADM	ADM user program in the host
DCAM	Data Communication Access Method
DCM	Data Communication Method
DDP	Distributed Data Processing
DDT	Data Display Terminal
DST	Data Station
EAC	Exit Address Code
EM	End Marker
ESC	Escape
FHS	Format Handling System
GAC	Group Address Code
HAC	Hardcopy Address Code
HCY	Hardcopy
HID	Hardcopy Identifier
IFG	Interactive Format Generator
IO	Input/Output
IPL	Initial Program Loader
ISAM	Index Sequential Access Method
ISD	Internal Symbol Dictionary
ITC	Inter Task Communication
KTT	Console driver task
LMS	Library Maintenance System
LTS	Local Terminal Support

## Abbreviations

---

MCLP	Macro Command Language Processor
MID	Multiplex Identifier
MSG	Message
MTAB	Message Table
MUX	Multiplex
NL	New Line character
NP	New Page character
OAC	OMNIS Address Code
OID	OMNIS Identifier
OMNISFC	OMNIS File Converter
OMNISKD	OMNIS configuration definitions
OMNISLC	OMNIS Logging Converter
OMNISTC	OMNIS Text Converter
PAC	Partner Address Code
PAM	Primary Access Method
PCB	Process Control Block
PID	Partner Identifier
PLUS	Program key load and save
PROP	Programmed Operating
PTN	Partner
SAM	Sequential Access Method
SDF	System Dialog Facility
SNMP	Simple Network Management Protocol
TAC	Transaction Code
TIAM	Terminal Interactive Access Method
TID	Terminal Identifier
TSN	Task Sequence Number
TSOS	Timesharing Operating System
UCON	Universal Console Support
UTM	Universal Transaction Monitor
VTSUCB	Virtual Terminal Support Control Block
XHCS	Extended Host Code Support
XS	Extended System Architecture



---

## 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)**  
**Administration and Programming**  
User Guide
- [2] **OMNIS-PROP (BS2000)**  
User Guide
- [3] **BS2000 OSD/BC**  
**Commands**  
User Guide
- [4] **BS2000 OSD/BC**  
**Executive Macros**  
User Guide
- [5] **BS2000 OSD/BC**  
**System Exits**  
User Guide
- [6] **BS2000 OSD/BC**  
**Introduction to System Administration**  
User Guide
- [7] **HIPLEX AF (BS2000)**  
**High Availability of Applications in BS2000**  
User Guide
- [8] **LMS (BS2000)**  
**SDF Format**  
User Guide
- [9] **VM2000 (BS2000)**  
**Virtual Machine System**  
User Guide

- [10] **SECOS** (BS2000/OSD)  
**Security Control System - Access Control**  
User Guide

---

# Index

\*  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*C  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*D  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*K  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*S  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*T  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*U  
    in CLSCON command 189  
    in OPNCON command 243  
    in PARAM command 268

\*UNLOCK 133

\$CONSOLE  
    as a partner 61, 63

\$DIALOG  
    as a partner 59

## A

access verification 161  
    connection loss 164  
    connection take-over 165  
    ID reader 161  
    Kerberos 161  
    password 161

ACK 303  
    in OPTION command 255  
    in SET command 282

acknowledgment 108

ACTION  
    in CHANGELOG command 182  
    in EXIT command 198  
    in MDEF command 230

ACTIV  
    in INF command 217, 218  
    in INF HCOPY command 219  
    in INF MUX command 220  
    in INF OMNIS command 221

ADD  
    in GROUP command 204

address code 299

ADM  
    in MESSAGE command 235  
    in OPTION command 256

administering  
    a number of applications 26  
    decentralized computer networks 31  
    large computer centers 27

administrator messages 158

AID 303

ALL  
    in MESSAGE command 235

- allocation table
  - TAC [142](#)
- APASS
  - for END command [198](#)
  - in CANCEL command [177](#)
  - in CHANGELOG command [182](#)
  - in END command [197](#)
  - in FILE command [201](#)
  - in INF command [217](#)
  - in INF GROUP command [219](#)
  - in INF TERMINAL command [218](#)
  - in MESSAGE command [235](#)
  - in OCCUPY command [238](#)
  - in OPTION command [256](#)
  - in PRIORITY command [275](#)
  - in SET command [282](#)
- Application Menu [135](#), [136](#)
- application switching [139](#), [156](#)
  - explicit [139](#)
  - implicit [141](#)
  - TAC [141](#)
  - using F keys [139](#)
  - using K keys [139](#)
- APPNAME
  - in OPNCON command [243](#)
- AUDIT
  - in SET command [282](#)
- authentication [159](#)
- authorization check
  - partner [58](#)
  - terminal [48](#)
- automatic disconnection [156](#)
- automatic LOGON [149](#)
- automatic switchover [84](#)
- automatic system operating [111](#)
- AVAILABILITY
  - HIPLEX AF [84](#)
- B**
  - BCAM [303](#)
  - BCAM message pool [109](#)
  - BELL
    - in MDEF command [230](#)
  - BERID
    - in OPNCON command [243](#)
  - BREAK [110](#)
  - BREAK key [299](#)
  - BREAK-KEY [110](#)
    - in OPTION command [256](#)
    - in SET command [283](#)
  - BS2000 console as a terminal [23](#), [50](#)
  - bulletin file [42](#)
- C**
  - CALL [109](#)
  - CALL key [299](#)
  - CALL-INF [110](#)
    - in OPTION command [256](#)
    - in SET command [283](#)
  - CALL-KEY [104](#), [109](#)
    - in OPTION command [257](#)
    - in SET command [283](#)
  - CANCEL
    - command description [175](#)
  - CHANGELOG
    - command description [180](#)
  - character substitution [113](#)
  - checks by OMNIS-MENU [143](#)
  - CLASS
    - in OPNCON command [244](#)
    - in PARAM command [268](#)
  - closing
    - connections to partners [189](#)
  - CLSCON
    - command description [189](#)
  - CMD
    - command description [191](#)
  - CMSG
    - in OPNCON command [249](#)
  - CMX [303](#)
  - CMX application
    - as a partner [66](#)

- CODE in OPNCON command 244
- COLOR
  - in OPNCON command 244
  - in OPTION command 257
  - in PARAM command 268
- color control 19, 118
- command file processing 201
- command files 43
- command input 172
- command mode 101
  - OMNIS-MENU 138
- command overview 167
- command priorities 274
- command scopes 168
- commands 167
  - OMNIS 175
- communication partners 15
- computational load
  - balancing 86
- configuration 35, 299
  - declared 35
  - dynamic 35
- configuration file 43, 299
- CONFUPDATE
  - command description 192
- CONNECT
  - in HCOPY command 208
- connection
  - of terminals to OMNIS 93
  - setting up 132
  - taking over 237
- connection clear-down
  - OMNIS-OMNIS 80
  - partner 58
  - terminal 49
- connection setup
  - explicit 80
  - for OMNIS-MENU 133
  - implicit 80
  - OMNIS-OMNIS 80
  - partner 53, 240
  - terminal 47
  - to TIAM application 149
  - to UTM application 150
- connections 300
  - direct (connection setup) 54
  - indirect (connection setup) 56
  - OMNIS-OMNIS 39, 80
  - taking over 77
  - to partners 37
- CPASS
  - in OPNCON command 244
- D**
- DADM 303
- data input
  - for OMNIS-MENU 131
- data protection 23
- DCAM 303
- DCAM application
  - as a partner 66
- DCAM return codes 298
- DCM 303
- DDP 303
- DDT 303
- DEL
  - in GROUP command 204
- DELETE
  - in CHANGELOG command 182
- dialog control
  - more than one application 25
  - unhomogeneous applications 26
- dialog display 120
- dialog log 300
- dialog logging 22
- DISPLAY
  - in FILE command 201
  - in MDEF command 232
- DISPLAY-MODE
  - in OPTION command 257
  - in SET command 283
- DST 303
- DSTMAX in SET command 284
- DUMP command description 195
- dump output 195
- dynamically authorized applications 61

### E

EAC [91](#), [303](#)  
eac  
    in EXIT command [198](#)  
EM [303](#)  
encryption [159](#)  
END  
    command description [197](#)  
ESC [303](#)  
EXCLUSIVE-PARTNER  
    with OPTION [257](#)  
    with SET [284](#)  
EXIT  
    command description [198](#)  
    in INF command [220](#), [221](#)  
    in OPNCON command [244](#)  
    in OPTION command [258](#)  
    in PARAM command [268](#)  
    in SET command [284](#)  
exit address code [91](#)  
exit group [199](#), [300](#)  
exit routine [21](#), [300](#)  
EXIT-AUTHORIZATION  
    in PRIORITY command [275](#)  
EXIT-PRIORITY  
    in PRIORITY command [275](#)  
explicit application switching [139](#)

### F

F keys  
    management [154](#)  
    switching application using [139](#)  
FAILURE  
    in CANCEL command [178](#)  
    in HCOPI command [208](#)  
FHS [303](#)  
FILE  
    command description [201](#)  
formatted mode [101](#)  
functionality  
    of OMNIS-MENU [125](#)

### G

GAC [303](#)  
gac [89](#)  
    in CLSCON command [189](#)  
    in GROUP command [204](#)  
    in OPNCON command [243](#)  
    in PARAM command [268](#)  
GROUP  
    command description [204](#)  
    in INF command [219](#)  
group address code [89](#)  
guest systems  
    administration [33](#)

### H

HAC [91](#), [303](#)  
hac  
    in HCOPI command [208](#)  
HALT  
    command description [206](#)  
hardcopy address code [91](#)  
hardcopy identifier [92](#)  
hardcopy log [20](#)  
hardcopy output [82](#)  
hardcopy unit  
    entering in table [208](#)  
HCOPI  
    command description [208](#)  
    in CANCEL command [178](#)  
    in INF command [219](#)  
    in OPNCON command [245](#)  
    in OPTION command [258](#)  
    in PARAM command [269](#)  
    in RESTART command [277](#)  
    in SET command [284](#)  
HCY [303](#)  
HELP  
    command description [210](#)  
help function  
    of OMNIS-MENU [128](#)  
HID [92](#), [303](#)  
    in CANCEL command [178](#)  
    in INF HCOPI command [219](#)  
HIPLEX AF [84](#)

HIPLEX MSCF 86

HOLD

- in HALT command 206
- in OPNCON command 245
- in OPTION command 259
- in PARAM command 269
- in SET command 285

HOLD-PRIORITY

- in PRIORITY command 275

HSMS 43

**I**

ID

- in MDEF command 231

ID card information 158

IFG 303

implementations of OMNIS 25

implicit application switching 141

INF

- command description 214

INPUT

- in FILE command 201
- in MESSAGE command 235

input message

- checking by OMNIS-MENU 144

INPUT-LOGGING

- in OPTION command 260
- in SET command 286

INPUT-SAVE

- in OPTION command 260
- in SET command 286

IPL 303

ISAM 303

ISD 303

ITC 303

**K**

K keys

- CALL key 109
- ESC sequences 140
- management 154
- special points relating to OMNIS 104
- switching application using 139
- switching between partners in formatted mode 98
- switching from formatted mode to line mode 101
- switching partners in formatted mode 18

KDCSIGN 150

Kerberos 159

- connection loss 164
- connection take-over 165

key 160

Key Distribution Center 160

KILL

- in CANCEL command 177

KPAC

- in OPNCON command 245
- in OPTION command 261
- in PARAM command 269
- in SET command 287

KTT 303

**L**

LCASE

- in OPNCON command 246

LINE25

- in OPNCON command 246
- in OPTION command 261, 270
- in SET command 287

LISTENING

- description of function 120
- in OPTION command 261

listening 120

LMS 303

LMSG

- in OPNCON command 249

LOAD

- in EXIT command 199

load balancing 86

load distribution 40

### LOGGING

in OPNCON command 246

in OPTION command 262

in PARAM command 270

in SET command 288

logging 122

scope and time 122

logging file 22, 43, 300

analysis 124

control 122

managing 180

paging 123

paging in 295

switching 123, 180

### LOGGING-PRIORITY

in PRIORITY command 275

login 132

login mask 133

### LOGON

automatic 149

logout 145

logout mask 145

loss of connection

OMNIS-OMNIS 80

partner 76

terminal 74

with HOLD=AUTO 75

with HOLD=NO 74

with HOLD=YES 75

### LPASS

in OPNCON command 244

LTS 303

## M

master 38, 300

maximum length

TIAM messages 60

MCLP 304

### MDEF

command description 229

MDEF command sequence 114

### MDEF-AUTHORIZATION

in PRIORITY command 276

### MESSAGE

command description 234

message

in MESSAGE command 235

sending a 234

MESSAGE (OMNIS command) 300

message answering 111, 300

message congestion 62

message exchange

between terminals 18

message files 43

message flow control 106

OMNIS-MENU 156

message length

TIAM/OMNIS 60

message number 112

message suppression 23, 111

message table 300

message tables 112

assigning partners 116

assignment to partners 115

creating 112

defining 229

deleting 116

generating entries 112

messages

responses 112

metasyntax 171

MID 92, 304

for CANCEL command 179

in INF MUX command 220

module library 43

### MODULES

in EXIT command 199



- MSG 304
  - in INF MTAB command 220
  - in MDEF command 230
- MTAB 304
  - in INF command 220
  - in OPNCON command 247
  - in OPTION command 262
  - in PARAM command 270
  - in SET command 288
- multiplex connection
  - HIPLEX AF 84
- multiplex connections 40
- multiplex identifier 92
- multiplex transport connection
  - system failure 84
- MUX 304
- N**
- NAME
  - in CHANGELOG command 182
- network 159
- NEWPAC
  - in OCCUPY command 238
- NL 304
- NP 304
- NPASS
  - in SET command 288
- O**
- OAC 90, 304
  - in CANCEL command 179
- OCCUPY 77
  - command description 237
- OI
  - in CANCEL command 179
- OID 92, 304
  - in INF command 221
- OMMFEND 145
- OMMFMAI 133
- OMMFSEL 135
- OMNIS
  - in INF command 221
  - in RESTART command 277
  - with HIPLEX AF 84
- OMNIS address code 90
- OMNIS command mode 101, 128, 138
- OMNIS files
  - overview of 42
- OMNIS files, overview 42
- OMNIS identifier 92
- OMNIS key 301
- OMNIS-MENU 25
  - access verification 161
  - allocation table 142
  - Application Menu 135
  - application switching 141
  - declaration of the principal 163
  - input checks by 144
  - logging in 132
  - logging out 145
  - login mask 133
  - logout mask 145
  - session 133
- OMNIS-OMNIS connection 39, 80
  - restarting 277
- OMNIS-PROP 30, 111
- OMNISFC 304
- OMNISKD 304
- OMNISLC 304
- OMNISTC 304
- OMY
  - message prefix 298
- OPASS
  - in HALT command 207
  - in OCCUPY command 239
  - in OPNCON command 247
  - in OPTION command 262
  - in PARAM command 270
- OPEN SESSION 151
- operation
  - in formatted mode 96
  - in line mode 94
- operator logon 61
- OPNCON
  - command description 240
  - in SET command 288
- OPNSTART
  - in SET command 289

### OPTION

command description 252

output control 105

### OUTPUT-LOGGING

in OPTION command 262

in SET command 289

## P

### P keys

lock 154

management 154

switching application using 139

### PAC 304

in CANCEL command 177

in OPNCON command 247

in OPTION command 263

in PARAM command 271

in SET command 289

### pac 89

in CLSCON command 189

in OPNCON command 243

in PARAM command 268

in TURN command 295

### PACMAX

in SET command 289

paging file 43, 301

paging mode

log 124

### PAM 304

### PARAM

command description 266

### PARTNER

in INF command 217

partner 15, 301

\$CONSOLE 61

\$DIALOG 59

\$VMCONS 63

CMX application 66

DCAM application 66

declared/undeclared 53

defining attributes 266

partner address code 89

partner connections 38

partner group 18, 37, 301

defining 204

partner identifier 58, 91

partner key 301

partner switching

in formatted mode 18

partner types 37

partner UTM application with multiplex

connection 69

password 134

changing 153

### PCB 304

PID 58, 91, 304

in CANCEL command 177

in INF command 218

in OCCUPY command 238

### PKEY

in OPNCON command 248

in OPTION command 263

in PARAM command 271

in SET command 290

### PLUS 304

principal 160

### PRIORITY

command description 274

### PRONAME

in HCOPY command 208

in OPNCON command 247

### PROP 304

PROP (OMNIS-PROP) 301

### PROTOCOL

in OPNCON command 248

### PTN 304

### PTNMAX

in SET command 290

### PTNNAME

in HCOPY command 208

in OPNCON command 248

**R**

Readme file 10  
realm 160  
REPEAT  
    in CHANGELOG command 183  
REPLY  
    in MDEF command 231  
    in OPTION command 263, 290  
reply suppression 23, 61  
RESTART 81  
    command description 277  
restart  
    OMNIS-OMNIS connection 81, 277  
restart message  
    OMNIS-MENU 156  
RFC standard 159  
ROUTE  
    in OPNCON command 248

**S**

SAM 304  
SAVE  
    in OPNCON command 248  
    in OPTION command 263  
    in PARAM command 271  
    in SET command 290  
save  
    P keys 156  
    screen contents 156  
save P keys 20, 117  
save screen 20, 117  
scope 301  
scope of commands 168  
SCREEN  
    in OPNCON command 248  
    in OPTION command 263  
    in PARAM command 271  
    in SET command 290  
screen overflow control 20, 108  
SDF 304  
SECOS 161  
SECURITY  
    with HOLD 49, 75, 78, 206, 259  
security system 159  
SECURITY, in HOLD 259  
session 40, 302  
    OMNIS-MENU 133  
SET  
    command description 278  
short messages 297  
SIGNON data 152  
SIZELIMIT  
    in CHANGELOG command 183  
slave 38, 300, 302  
SNMP 304  
    in OPNCON command 248  
    in PARAM command 271  
SoftBooks 297  
SSO system 159  
standard messages 297  
standard partner groups 90, 302  
standard restart message 157  
start sequence 73  
START-SEQUENCE  
    in OPNCON command 249  
startup file 21, 42, 302  
station name  
    SHOW-STATUS command 59  
switching  
    between partners in formatted mode 98  
syntax description  
    OMNIS commands 171  
SYSTEM  
    in INF command 221, 222  
system operating  
    automatic 111

### T

- table-name
  - in MDEF command 230
- TAC 304
  - application switching 141
- TAC allocation table 142
  - configuration update 143
- TERMINAL
  - in INF command 218
  - in MESSAGE command 234
- terminal
  - as a BS2000 console 22
  - changing during operation 18
  - defining attributes 252
  - signing off from OMNIS 206
- terminal identifier 48, 91
- terminal types 37
- terminate
  - a connection 175
- TESTMODE
  - in OPTION command 264
- text file 43
- TIAM 304
- TID 48, 91, 304
  - in INF command 217
  - in INF GROUP command 219
  - in INF TERMINAL command 219
  - in MESSAGE command 234
  - in OCCUPY command 238
- TRACE
  - command description 293
- trace file 43
- transport connection 40
- TSN 304
- TSOS 304
- TURN
  - command description 295
- TYP
  - in OPNCON command 249
- TYPE
  - in INF SYSTEM command 221

### U

- UCON 304
- UCON partner 61
- UTM 304

### V

- virtual host name 84
- VTSUCB 304

### X

- XHCS 24, 158, 304
- XS 304