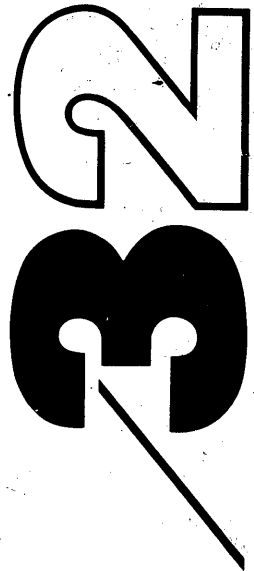


**IBM System/32  
System Control Programming  
Reference Manual  
Word Processing**



*IBM System/32  
Word Processing  
Programming Information*

GC34-0078-3  
File No. S32-36

**Program Number  
5725-SC1**

**IBM System/32  
System Control Programming  
Reference Manual  
Word Processing**

**GC34-0078-3  
File No.  
S32-36  
Program Number  
5725-SC1**

#### **Fourth Edition (November 1978)**

This is a major revision of, and obsoletes, GC34-0078-2. Because the changes and additions are extensive, this publication should be reviewed in its entirety.

This edition applies to version 08, modification 00 of the IBM System/32 system control program (Program Number 5725-SC1) and to all subsequent versions and modifications until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest *IBM System/32 Bibliography*, GC20-0032, for the editions that are applicable and current.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Address your comments about this publication to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

This reference manual provides system programmers with information needed to establish administrative and operating procedures for word processing for an IBM System/32. Information is provided for programmers to run word processing application programs on IBM System/32 and use the system procedures and utilities provided with IBM System/32.

This manual contains:

- An introduction to word processing.
- A general description of IBM System/32 procedures and a detailed description of each word processing procedure. A detailed description of the command statements that evoke the procedures and a summary of command statement formats.
- A description of each word processing utility program provided with IBM System/32 and a description of associated utility control statements.
- A description of the word processing communications utility and a description of the associated command statements and OCL.
- An appendix showing the standard characters for IBM System/32.
- A glossary defining word processing terms used in this manual.

### Prerequisite Publications

*IBM System/32 Introduction*, GC21-7582, provides an overview of the system and its characteristics.

### Related Publications

*IBM System/32 Operator's Guide*, GC21-7591, provides detailed instructions for operating System/32.

*IBM System/32 System Control Programming Reference Manual*, GC21-7593, provides system programmers information needed to establish operating and administrative procedures for the IBM System/32.

Titles and abstracts of related publications are listed in the bibliography:

*IBM System/32 Bibliography*, GC20-0032.



# Contents

<b>LIST OF ABBREVIATIONS AND ACRONYMS</b> . . . . .	vii	WPSET Procedure . . . . .	17
<b>HOW TO USE THIS MANUAL</b> . . . . .	ix	WPSET Command Statement Format . . . . .	17
<b>CHAPTER 1. INTRODUCTION</b> . . . . .	1	WPSET Parameters . . . . .	17
<b>LOADING AND RUNNING WORD PROCESSING</b>		WPSTART Procedure . . . . .	17
PROGRAMS . . . . .	1	WPSTART Command Statement Format . . . . .	18
Setup . . . . .	1	WPSTART Parameters . . . . .	18
Activation . . . . .	1	WPSTOP Procedure . . . . .	18
Termination . . . . .	2	WPSTOP Command Statement Format . . . . .	19
Inquiry Support . . . . .	2		
<b>CHAPTER 2. PROCEDURES</b> . . . . .	3	<b>CHAPTER 3. UTILITIES</b> . . . . .	21
Keyboard Procedure . . . . .	4	\$WPKBD—Keyboard Redefine Utility Program . . . . .	22
KEYBOARD Command Statement Format . . . . .	4	\$WPKBD OCL and Utility Control Statement	
KEYBOARD Parameters . . . . .	4	Format . . . . .	22
WPBELT Procedure . . . . .	5	\$WPKBD Control Statement Parameters . . . . .	22
WPBELT Command Statement Format . . . . .	5	American Keyboards . . . . .	23
WPBELT Parameters . . . . .	5	Austrian/German Keyboards . . . . .	28
WPDELETE Procedure . . . . .	6	French Keyboards . . . . .	30
WPDELETE Command Statement Format . . . . .	6	Italian Keyboards . . . . .	32
WPDELETE Parameter . . . . .	6	Danish Keyboards . . . . .	33
WPDELETE Example . . . . .	6	British Keyboards . . . . .	34
WPFREE Procedure . . . . .	6	\$WPKY—Security File Update Utility Program . . . . .	35
WPFREE Command Statement Format . . . . .	6	\$WPKY OCL and Control Statement Sequence . . . . .	36
WPFREE Parameters . . . . .	7	\$WPLUF—Delete Document Library Member Utility	
WPFREE Example . . . . .	7	Program . . . . .	38
WPIMID Procedure . . . . .	7	WPLUF Utility Control Statement Formats . . . . .	39
WPIMID Command Statement Format . . . . .	7	\$WPLUF Control Statement Parameters . . . . .	39
WPIMID Parameters . . . . .	8	\$WPLUF OCL and Control Statement Sequence . . . . .	40
WPINIT Procedure . . . . .	8	\$WPLUF OCL Parameters . . . . .	41
WPINIT Command Statement Format . . . . .	8	WPLUF Examples . . . . .	41
WPINIT Parameters . . . . .	9	\$WPLUI—Document Library Initialization	
WPINIT Example . . . . .	9	Utility Program . . . . .	42
WPKEY Procedure . . . . .	9	\$WPLUI OCL and Control Statement Sequence . . . . .	42
WPKEY Command Statement Format . . . . .	9	\$WPLUI OCL Parameters . . . . .	42
WPLMAINT Procedure . . . . .	10	\$WPLUI Example . . . . .	43
WPLMAINT Command Statement Format . . . . .	10	\$WPLUL—Load Document Library Members from	
WPLMAINT Parameters . . . . .	10	Diskette Utility Program . . . . .	43
WPMCCARD Procedure . . . . .	10	\$WPLUL Utility Control Statement Formats . . . . .	44
WPMCCARD Command Statement Format . . . . .	11	\$WPLUL Utility Control Statement Parameters . . . . .	44
WPMCCARD Parameter . . . . .	11	\$WPLUL OCL and Control Statement Sequence . . . . .	45
WPPRINT Procedure . . . . .	11	\$WPLUL OCL Parameters . . . . .	46
WPPRINT Command Statement Formats . . . . .	11	\$WPLUL Examples . . . . .	46
WPPRINT Parameters . . . . .	12	\$WPLUM—LDAM Maintenance Utility Program . . . . .	47
WPPRINT Examples . . . . .	12	Rekey . . . . .	47
WPPURGE Procedure . . . . .	13	Rename . . . . .	47
WPPURGE Command Statement Format . . . . .	13	Delete Member . . . . .	47
WPPURGE Parameters . . . . .	13	\$WPLUM Sysin Format . . . . .	47
WPRESTOR Procedure . . . . .	13	\$WPLUM Control Statement Format . . . . .	48
WPRESTOR Command Statement Formats . . . . .	14	\$WPLUM Control Statement Examples . . . . .	49
WPRESTOR Parameters . . . . .	14	\$WPLUP—Document Directory Print Utility	
WPRESTOR Example . . . . .	15	Program . . . . .	50
WPSAVE Procedure . . . . .	15	\$WPLUP Utility Control Statement Formats . . . . .	51
WPSAVE Command Statement Formats . . . . .	15	\$WPLUP Control Statement Parameters . . . . .	51
WPSAVE Parameters . . . . .	16	\$WPLUP OCL and Control Statement Sequence . . . . .	52
WPSAVE Examples . . . . .	16	\$WPLUP OCL Parameters . . . . .	53
		\$WPLUP Examples . . . . .	53

\$WPLUU—Unload Document Library Members to Diskette Utility Program . . . . .	54	Communication Specification Statements . . . . .	67
\$WPLUU Utility Control Statement Format . . . . .	55	LINE Statement . . . . .	67
\$WPLUU Control Statement Parameters . . . . .	55	REMOTE Statement . . . . .	68
\$WPLUU OCL and Control Statement Sequence . . . . .	56	MSG Statement . . . . .	69
\$WPLUU OCL Parameters . . . . .	56	MEMBER Statement . . . . .	69
\$WPLUU Examples . . . . .	57	RECORD Statement . . . . .	70
\$WPMCU—Magnetic Card Utility Program . . . . .	57	END Statement . . . . .	71
\$WPMCU OCL and Control Statement Sequence . . . . .	57	WCU Procedure . . . . .	71
\$WPMCU OCL Parameters . . . . .	58	WCU Command Statement Format . . . . .	71
\$WPPBI—Print Belt Identification Utility . . . . .	58	WCU Command Statement Parameters . . . . .	71
\$WPPBI Utility Control Statement Format . . . . .	58	Examples of Document Library Member Transmission/Reception . . . . .	72
\$WPPBI Control Statement Parameter . . . . .	58	WCUS Procedure . . . . .	73
\$WPPBM—Printer Belt Map Utility Program . . . . .	59	WCUS Command Statement Formats . . . . .	73
\$WPPBM Utility Control Statement Format . . . . .	59	WCUS Command Statement Parameters . . . . .	74
\$WPPBM Control Statement Parameter . . . . .	59	Example of System/32 Sequential File Transmission . . . . .	75
\$WPSET—Word Processing Configuration Utility Program . . . . .	60	\$WCUM OCL and Utility Control Statement Sequence . . . . .	76
\$WPSET OCL and Control Statement Sequence . . . . .	60	\$WCUM OCL Parameters . . . . .	76
\$WPSET OCL Parameters . . . . .	60	\$WCUM Examples . . . . .	77
\$WPURG—Purge Document Directory Utility Program . . . . .	60	Considerations For Communicating With a System/32 . . . . .	79
\$WPURG OCL and Control Statement Sequence . . . . .	61	Considerations For Communicating With an Office Machine . . . . .	80
\$WPURG OCL Parameters . . . . .	61		
<b>CHAPTER 4. COMMUNICATIONS . . . . .</b>	<b>63</b>	<b>APPENDIX A. IBM SYSTEM/32 CHARACTERS . . . . .</b>	<b>81</b>
Communication Utility Functions . . . . .	63	<b>GLOSSARY . . . . .</b>	<b>87</b>
WPCU Operation . . . . .	64	<b>INDEX . . . . .</b>	<b>89</b>
Executing WPCU Using a Point-to-Point Network . . . . .	65		
Nonswitched Network . . . . .	65		
Switched Network . . . . .	66		

## List of Abbreviations and Acronyms

ARR	address recall register	OCL	operation control language
CE	customer engineer	PLCA	program level communication area
CRT	cathode ray tube (display screen)	RIB	request indicator byte
DTF	define the file	SCA	system communication area
ERP	error recovery procedure	SCP	system control program
ERT	error recording table	SIO	start input/output
F1	disk file	SSN	starting sector address
HDR	header	SWA	scheduler work area
I1	diskette drive	UPSI	user program status indicator
IAR	instruction address register	VTOC	volume table of contents
ID	identification	WP	word processing
IOB	input/output block	WPCU	word processing communications utility
IOS	input/output supervisor	WPLCA	word processing library communication area
IPL	initial program load	WPWA	word processing work area
JJT	job control	XR1	index register 1
LDAM	linked direct access method	XR2	index register 2
MCU	mag card unit		
MIC	message identification code		





## How to Use This Manual

This manual has four chapters. Chapter 1 is general operational information. Chapter 2 describes word processing procedures and command statements. Chapter 3 describes word processing utilities. Chapter 4 describes word processing communications.

### Chapter 1

Refer to Chapter 1 if you want general information on operating word processing.

### Chapter 2

Refer to Chapter 2 if you want to know:

- What word processing procedures are supplied with IBM System/32 and the function of each.
- The format and contents of the command statements that evoke the word processing procedures provided with IBM System/32.

### Chapter 3

Refer to Chapter 3 if you want to know:

- What word processing utilities are supplied with IBM System/32 system control programming.
- What the function of each utility program is.
- What OCL and utility control statements are necessary to load and run each utility program.

### Chapter 4

Refer to Chapter 4 if you want to know:

- Word processing communications capabilities.
- The format and content of the command statements that invoke the word processing communications utility.



### Loading and Running Word Processing Programs

The operational sequence of running IBM System/32 word processing is:

1. Setup
2. Activation
3. Termination

#### SETUP

- Run the WPSET procedure (part of CNFIGSCP).
- Before starting word processing, assure that your document library is available on fixed disk.
- If a new document library is required, create it using the WPINIT procedure (see index entry: *WPINIT procedure*).
- If the library is on diskette, load it on fixed disk using the WPRESTOR procedure (see index entry: *WPRESTOR procedure*). If a library has not been created on disk, one must be created before loading the library from diskette (see index entry: *WPINIT procedure*).
- If the printer belt has to be changed, identify the belt using the WPIMID procedure (see index entry: *WPIMID procedure*).

#### ACTIVATION

The WPSTART procedure, keyed at the System/32 keyboard, loads word processing. The word processing system input (sysin) device can be the keyboard, disk, document library, or mag card unit. If the sysin device is not specified, the keyboard becomes the sysin device (see index entry: *WPSTART*).

## TERMINATION

Word processing is terminated by all sysin devices by:

- Pressing the INQ key and replying option 2 to message 4749 when the inquiry request is honored.
- Executing the WPSTOP procedure from the jobstream (see index entry: *WPSTOP procedure*).

Word processing is also terminated when EOF is encountered if the sysin device is a System/32 file, or a document library member.

When word processing is terminated, the keyboard definition is returned to the base System/32 keyboard.

If abnormal termination of word processing occurs (option 2 or 3 is taken from a data management or other nonword processing message), unpredictable results may occur because the word processing work file (WPWRKFLE) and document library files were not closed.

If abnormal termination does occur, the word processing work file (WPWRKFLE) must be deleted before issuing WPSTART again. The document library may also need to be rebuilt if you were adding or deleting members at the time of abnormal termination.

## INQUIRY SUPPORT

Normal System/32 inquiry support may be affected because word processing appears to be active when it is not. Activate word processing via the WPSTART procedure to remedy the problem.

After word processing has been initiated, normal inquiry handling is not in effect. Inquiry is only allowed at the end of tasks and during task processing at certain times when I/O is quiesced. When running word processing in base mode, the system returns the keyboard definition to the standard System/32 keyboard when entering inquiry mode.

**Base/Inquiry Mode:** When the WPSTART command is issued, a test is made for the word processing work file (WPWRKFLE). If it exists, word processing becomes active in inquiry mode. If it does not exist, word processing becomes active in base mode.

## Chapter 2. Procedures

This part of the manual describes all the word processing procedures supplied with IBM System/32 SCP to provide you with an easy method of using word processing functions. The following information is given for each word processing procedure:

- The function of the procedure
- The format of the command statement that evokes the procedure
- A description of the parameters of the command statement used to evoke the procedure

Examples are given for many of the command statements.

In the descriptions of command statement formats and parameters, capitalized words and letters, numbers, special characters, brackets, and braces have special meanings. Capitalized expressions must be entered as they appear in the descriptions. Sometimes numbers or nonalphabetic characters may appear in a capitalized expressions—such numbers and characters must also be entered as they are shown. Words and expressions that are not capitalized must be replaced with a value that is appropriate to your job. The values you can use are listed in the parameter descriptions.

**Note:** Only the WPCKPT, WPLMAINT, WPSTOP, and Word Processing System/32 program product task procedures can be run when word processing is active. In addition, the other word processing SCP procedures should not be run in inquiry mode when word processing is the base mode.

The procedures that run under word processing (WPCKPT, WPLMAINT, WPSTOP, and Word Processing System/32 program product task procedures) must be preceded by two commas (,,) unless the country code specified in the WPSET procedure is 01, in which case they may be preceded by two slashes (/ /).

## KEYBOARD PROCEDURE

The KEYBOARD procedure allows you to change the keyboard/display screen to uppercase/lowercase, as well as changing the EBCDIC representation of certain keys.

The procedure evokes the \$WPKBD utility (see index entry: \$WPKBD).

*Note:* When the keyboard is defined to any of the uppercase/lowercase keyboards, the shift key must be used to enter uppercase alphabetic characters for OCL or procedures. If an invalid parameter is entered, the keyboard remains in the same mode.

## KEYBOARD Command Statement Format

KEYBOARD *nnn*

## KEYBOARD Parameters

<b>nnn</b>	<b>Keyboard</b>
001	IBM Office System 6 96-character keyboard
002	American Standard 101-A keyboard
003	American Standard 101-B keyboard
004	American Standard 101-C keyboard
005	American Standard 101-D keyboard
007	Legal 149 keyboard
008	Legal 177 keyboard
009	Pica 046 keyboard
017	Accounting 172 keyboard
018	Accounting 174 keyboard
026	Austrian/German 88-character keyboard
027	Austrian/German 96-character keyboard
028	Austrian/German 88-character keyboard
029	Austrian/German 96-character keyboard
031	French 96-character keyboard
032	French 88-character keyboard
041	Italian 96-character keyboard
046	Italian 88-character keyboard
056	Danish 88-character keyboard
057	Danish 96-character keyboard
066	British 88-character keyboard
067	British 96-character keyboard
100	Standard System/32 Monocase EBCDIC
202	Danish System/32 Monocase EBCDIC
204	French System/32 Monocase EBCDIC (AZERTY)
205	French System/32 Monocase EBCDIC (QWERTY)
206	Austrian/German System/32 Monocase EBCDIC (QWERTZ)
207	Italian System/32 Monocase EBCDIC
212	British System/32 Monocase EBCDIC
250	Modified System/32 Dual Case EBCDIC

## WPBELT PROCEDURE

The WPBELT procedure allows you to print the lowercase alphabet (a-z) as uppercase (A-Z) on a 64- or 48-character print belt. The lowercase characters that do not exist on the 48- or 64-character belt are mapped into printable uppercase characters. Characters that co-exist on the print belts are unaffected. Characters that do not exist on the 48- or 64-character print belt are printed as a period (.).

The procedure evokes the \$WPPBM utility (see index entry: \$WPPBM).

**Note:** Do not use the WPBELT procedure in the word processing environment. Certain special characters may not be mapped properly.

### WPBELT Command Statement Format

WPBELT *nnn*

### WPBELT Parameters

The WPBELT parameter specifies the two- or three-character belt ID of the 96-character set to be converted to a 48- or a 64-character set. This ID is the same as used on the WPIMID procedure.

<b>nnn</b>	<b>Belt</b>
96	American WP belt supporting keyboards 1, 2, 3, 4, 5, 7, 8, 9, 17, and 18
100	Danish WP belt supporting keyboards 56 and 57
101	French WP belt supporting keyboard 31
102	French WP belt supporting keyboard 32
103	German WP belt supporting keyboards 26 and 27
104	German WP belt supporting keyboard 28
105	German WP belt supporting keyboard 29
106	Italian WP belt supporting keyboards 41 and 46
116	British WP belt supporting keyboards 66 and 67



## WPDELETE PROCEDURE

The WPDELETE procedure deletes an entire document library by deleting the document directory and the document storage file.

This procedure evokes the DELETE procedure.

*Note:* The document security check is *not* performed.

## WPDELETE Command Statement Format

```
WPDELETE libname
```

## WPDELETE Parameter

libname The name of document library to be deleted.

## WPDELETE Example

To delete the document library MYLIB, you would enter:

```
WPDELETE MYLIB
```

## WPFREE PROCEDURE

The WPFREE procedure allows you to delete selected members from a document library. Only those members to which access is allowed are deleted. Access is determined using the key entered via the key parameter. Directory data is printed for each deleted member.

This procedure evokes the \$WPLUF utility (see index entry: \$WPLUF).

## WPFREE Command Statement Format

```
WPFREE [libname] , [key] , [group] , [docname] , [yymmdd-1] [yymmdd-2]  
       [WPL] , [NULL] , , , [000000] [999999]
```

If members in the document library have a *groupname*, you must specify that *groupname*. If you specify *groupname* without a *docname*, all members with that *groupname* are affected. If members in the document library have a NULL *groupname* (blank qualifier with *docname* only), the *docname* must be specified. FROM dates and TO dates can only be used when *groupname* and/or *docname* are specified.

## WPFREE Parameters

libname WPL	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
key NULL	The security key to be used in accessing members of a document directory file that are to be deleted. NULL is the default value.
group	Specifies the qualifier portion (the first 8 bytes) of the member name to be selected.
docname	Specifies that document name portion (last 8 bytes) of the member name to be selected.
yymmdd-1 000000	Specifies that all members with a creation date equal to or greater than this parameter will be selected. The default is 000000.
yymmdd-2 999999	Specifies that all members with a creation date equal to or less than this parameter will be selected. The default is 999999.

## WPFREE Example

To delete the group OURDOCS created between March 15, 1976 and September 9, 1976, you would enter:

```
WPFREE ,,OURDOCS,,760315,760909
```

## WPIMID PROCEDURE

The WPIMID procedure loads the requested printer belt image and places its identification in the system configuration record and program level communication area (PLCA).

This procedure evokes the utility programs \$WPPBI and \$SETCF.

## WPIMID Command Statement Format

```
WPIMID nnn
```

## WPIMID Parameters

nnn      A two- or three-character decimal ID of the mounted printer belt.  
          Leading zeros are not required.

Valid 48-character belt IDs are:

41	Austrian/German (QWERTZ)
42	Danish
47	British
48	American/French/Italian
49	48HN (FORTRAN)

Valid 64-character belt IDs are:

64	American
65	French/Italian
66	Austrian/German (QWERTZ)
67	Danish
72	British

Valid 96-character belt IDs (and their corresponding keyboard IDs) are:

96	American (1/2/3/4/5/7/8/9/17/18)
100	Danish (56/57)
101	French (31)
102	French (32)
103	Austrian/German (26/27)
104	Austrian/German (28)
105	Austrian/German (29)
106	Italian (41/46)
116	British (66/67)

## WPINIT PROCEDURE

The WPINIT procedure creates an initialized document library by allocating a document directory and a document storage file. All document storage records are chained together into the free queue.

This procedure evokes the \$WPLUI utility (see index entry: \$WPLUI).

## WPINIT Command Statement Format

```
WPINIT [libname] [records-1] [records-2]  
      [WPL] [80] [80]
```

## WPINIT Parameters

libname WPL	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
records-1 80	Size of the document directory file to be allocated on disk. The default is 80. The minimum size is also 80.
records-2 80	Size of the document storage file to be allocated on disk. The default is 80. The minimum size is 10.

## WPINIT Example

To allocate the document library PSTDU with 240 document storage records, you would enter:

```
WPINIT PSTDU, , 240
```

## WPKEY PROCEDURE

The WPKEY procedure allows you to update the word processing security file.

The procedure evokes the \$WPKEY utility, which prompts the operator (see index entry: \$WPKEY).

## WPKEY Command Statement Format

```
WPKEY
```

## WPLMAINT PROCEDURE

The WPLMAINT procedure allows the user to evoke the \$WPLUM utility in order to process the user input control statements. This procedure runs only in the word processing job stream and can be entered via valid word processing SYSIN devices (keyboard, mag card unit, and fixed disk) only. The document library is not supported.

If you enter command statements using a mag card typewriter, hyphens must be *required* hyphens.

The WPLMAINT procedure evokes the \$WPLUM utility (see index entry: \$WPLUM).

### WPLMAINT Command Statement Format

{ , , } WPLMAINT  
{ // }

### WPLMAINT Parameters

User control statements are described under \$WPLUM.

*Note:* WPLMAINT can be evoked only in the word processing environment and is an exception to the other word processing procedures. There is no space between the ,, or // and the WPLMAINT when entering the command statement in the job stream. The double slash is only valid when the country code is 01 (specified at installation time via the WPSET procedure).

## WPMCCARD PROCEDURE

The WPMCCARD procedure allows you to:

- write a card eject character in character 1 of track 1 so the card is accepted as blank by the mag card unit access method (MCUAM).
- write all blanks on the card with a card eject character in character 1 of track 1 for security purposes.
- check magnetic cards for bad tracks.

The procedure evokes the \$WPMCU utility (see index entry: \$WPMCU).

## WPMCCARD Command Statement Format

WPMCCARD [ERASE]

### WPMCCARD Parameter

**ERASE** Specifies that blanks are to be written on entire card with a card eject written in character 1 of track 1. If this parameter is omitted, the utility writes only a card eject in character 1 of track 1.

**Note:** If this utility is not terminated and data cards are placed in MCU hopper, the data on the cards may be destroyed. To terminate the utility, respond with option 2 to SCP message 4759 MCU-HOPPER EMPTY.

### WPPRINT PROCEDURE

The WPPRINT procedure lists the document directory file of a document library. Only those records to which access is allowed are printed. Access is determined using the key entered via the key parameter. The procedure evokes the \$WPLUP utility (see index entry: \$WPLUP).

### WPPRINT Command Statement Formats

Use	Format
List all entries of document directory.	WPPRINT [libname WPL] [key NULL]
List selected members of a document directory.	WPPRINT [libname WPL] , [key NULL] ,ACCEPT , [group] , [docname] , [yymmdd-1 000000] [yymmdd-2 999999]
List statistics only.	WPPRINT [libname WPL] ,\$\$\$

If members in the document library have a *groupname*, you must specify that *groupname*. If you specify *groupname* without a *docname*, all members with that *groupname* are affected. If members in the document library have a NULL *groupname* (blank qualifier with *docname* only), the *docname* must be specified. FROM dates and TO dates can only be used when *groupname* and/or *docname* are specified.

## WPPRINT Parameters

libname WPL	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
key NULL	The security key to be used in accessing members of a document directory file. NULL is the default.
ACCEPT	Specifies that selected members of a document directory file are to be listed. If the ACCEPT parameter is not specified, all entries in the document directory are listed.
group	Specifies the qualifier portion (the first 8 bytes) of the member(s) to be selected.
docname	Specifies the document name portion (the last 8 bytes) of the member to be selected.
yymmdd-1 000000	Specifies that all members with a creation date equal to or greater than this parameter are selected. The default is 000000.
yymmdd-2 999999	Specifies that all members with a creation date equal to or less than this parameter are selected. The default is 999999.
\$\$\$\$	Specifies that accumulated member totals and file utilization statistics are printed.

## WPPRINT Examples

To list all members of the document library MYLIB, with a NULL security key, you would enter:

```
WPPRINT MYLIB
```

To print members of the default document library, with a security key LOCK, within group OUTGO, created on or after June 8, 1976, you would enter:

```
WPPRINT ,LOCK,ACCEPT,OUTGO,,760608
```

## WPPURGE PROCEDURE

The WPPURGE procedure frees the document directory of entries left in the freed status. Because the document directory is an indexed file a freed directory entry cannot be reused by any other name but the original. A situation can thus arise where the user has a document directory full of entries and cannot add a new member because of these freed entries. The running of this procedure will make these freed entries available for any new members. The procedure evokes the \$COPY, \$DELET, \$RENAM, and \$WPURG utilities (see index entries: \$COPY, \$DELET, \$RENAM, and \$WPURG).

### WPPURGE Command Statement Format

Use	Format
Purge the specified library of freed directory entries.	WPPURGE [ libname ] [ , records-1 ] [ <u>WPL</u> ] [ , <u>80</u> ]

### WPPURGE Parameters

libname <u>WPL</u>	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If libname is not specified, the default name is WPL.
records-1 80	Size of the newly organized directory file on disk. The default is 80.

## WPRESTOR PROCEDURE

The WPRESTOR procedure adds members to a document library from a diskette file that was created by the WPSAVE procedure.

Directory information is printed for each member that is loaded.

Insert diskette messages for multivolume files are automatically displayed as required, with appropriate label and volume-sequence-number checking.

The procedure evokes the \$WPLUL utility (see index entry: \$WPLUL).



## WPRESTOR Command Statement Formats

Use	Format
Restore all members from a previously saved file.	WPRESTOR [libname] WPL
Restore selected members from a previously saved file.	WPRESTOR [libname] WPL ,ACCEPT , [group] , [docname] , [yymmdd-1] 000000 [yymmdd-2] ,999999

## WPRESTOR Parameters

libname WPL	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
ACCEPT	Specifies that selected members of a previously saved data file are restored to the disk. If the ACCEPT parameter is not specified, all members of the data file are restored.
group	Specifies the qualifier portion (the first 8 bytes) of the member to be selected.
docname	Specifies the document name portion (the last 8 bytes) of the member(s) to be selected.
yymmdd-1 000000	Specifies that all members with a creation date equal to or greater than this parameter are selected. The default is 000000.
yymmdd-2	Specifies that all members with a creation date equal to or less than this parameter are selected. The default is 999999.

## WPRESTOR Example

To restore all members previously saved by the WPSAVE procedure, you would enter:

```
WPRESTOR
```

To restore the document XYZ within group ABC of a file that was saved by WPSAVE, you would enter:

```
WPRESTOR ,ACCEPT,ABC,XYZ
```

To restore all documents within group THEGROUP created between February 1, 1982 and March 31, 1982 you would enter:

```
WPRESTOR ,ACCEPT,THEGROUP , ,820201,820331
```

## WPSAVE PROCEDURE

The WPSAVE procedure causes selected members of a document library or a complete document library to be copied to diskette(s). Appropriate insert diskette messages are given to the operator when a WPSAVE request causes a multivolume diskette file to be created. Directory information is printed for each document library member that is unloaded. This procedure cannot copy to a format 2 (512-byte sectors) diskette.

This procedure evokes the \$WPLUU utility (see index entry: \$WPLUU).

**Note:** If you delete the original file from disk after saving a file by copying it to diskette(s), the diskette(s) file becomes the master copy of the file.

## WPSAVE Command Statement Formats

Use	Format
Save complete document library on diskette.	WPSAVE [libname WPL] , [retention days 999] [,vol-id]
Save selected members of a document library on diskette.	WPSAVE [libname WPL] , [retention days 999] , [vol-id] ,ACCEPT , [group] , [docname] , [yymmdd-1 000000] [yymmdd-2 '999999]

## WPSAVE Parameters

libname WPL	The name of the document library selected. It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
retention days 999	Number of days (0 to 999) the diskette file is to be retained. The default is 999.
vol-id	Volume label of diskette. It must be one to six alphameric characters.
ACCEPT	Specifies that selected members of a document library are to be copied to diskette. If the ACCEPT parameter is not specified, all members of the data file are saved.
group	Specifies the qualifier (the first 8 bytes) of the member(s) to be selected.
docname	Specifies the document name portion (the last 8 bytes) of the member to be selected.
yymmdd-1 000000	Specifies that all members with a creation date equal to or greater than this parameter will be selected. The default is 000000.
yymmdd-2 999999	Specifies that all members with a creation date equal to or less than this parameter will be selected. The default is 999999.

## WPSAVE Examples

To save a document library SALES for 10 days on a diskette labeled 987654, you would enter:

```
WPSAVE SALES,10,987654
```

To save the document library members in group SALESB with a creation date equal to or greater than February 1, 1976 in the document library DAILY, you would enter:

```
WPSAVE DAILY,, ,ACCEPT,SALESB,,760201
```

## WPSET PROCEDURE

The WPSET procedure sets the console keyboard ID, the default mag card translate table ID, and the // allowed indicator in the system configuration record.

### WPSET Command Statement Format

```
WPSET cc
```

### WPSET Parameters

cc            A two-character decimal country code used to determine the console keyboard ID and the default mag card translate table ID to be placed in the configuration record. Any code other than 01 (U.S.) sets the word processing indicator to allow the ,, format only.

cc	Console ID	Mag Card ID	// Allowed	Country
01	100	002	Yes	United States
02	212	066	No	UK
03	206	026	No	Germany/Austria
04	204	032	No	France (Azerty)
05	207	046	No	Italy
06	202	056	No	Denmark
17	205	032	No	France (Qwerty)

## WPSTART PROCEDURE

The WPSTART procedure activates word processing. It defines the printer as SYSLIST and the keyboard/display screen as SYSLOG. A check is made for presence of a Word Processing Work Area on fixed disk. If not found, WPSTART executes \$WPJCT to allocate the work area. If the work area (filename WPWRKFLE) is found on the fixed disk, #WPJCT is not executed and word processing is assumed to be executing in inquiry mode. WPSTART passes control to \$WPJIT to initialize the word processing work area (WPWA). WPSTART then passes control to \$WPJAT which reads a stream of text from SYSIN. The text is a group of control statements that contain procedures to be executed along with data to be processed for word processing. Upon completion of the jobstream in base mode, the work area is deleted and the keyboard reset to standard System/32.

## WPSTART Command Statement Format

WPSTART 

KB
MC
FD
DL

libname
'WPL

### WPSTART Parameters

KB	The keyboard/display screen is the sysin device. This is the default sysin device.
MC	The Mag Card Unit is the sysin device.
FD	The sysin device is a System/32 standard fixed disk file. The record size for this file must be 128.
DL	The document library is the sysin device. Prompts are issued for the member name and the security key. The groupname must be <i>sysin</i> (lowercase) which may be created via the STORE task. The document library must not contain WPLMAINT input control statements.
libname	The name of the document library selected.
WPL	It can be up to five alphameric characters, the first of which must be alphabetic. If <i>libname</i> is not specified, the default name assigned to the document library is WPL. If a <i>libname</i> is specified for inquiry mode, it is ignored and the same library that is active in base mode is used. No diagnostic or error message is issued.

If MC, FD, or DL is specified, additional prompts are issued to determine whether to begin processing with the first job or some other job; and the name of that other job.

### WPSTOP PROCEDURE

The WPSTOP procedure stops word processing and closes all open files. The WPSTOP procedure evokes \$WPJST and \$WPJAT.

WPSTOP can be evoked only in the word processing environment and is an exception to the other word processing procedures.

## WPSTOP Command Statement Format

$\left. \begin{array}{l} // \\ // \end{array} \right\} \text{WPSTOP}$

**Note:** There is no space between the `//` or `//` and the `WPSTOP` when entering the command statement in the job stream.



## Chapter 3. Utilities

This section describes each word processing utility provided with IBM System/32. The following information is given for each utility:

- The function of the utility
- The format of the related utility control statement(s)
- A description of the parameters in the related utility control statement
- The sequence of the OCL and utility control statements required to evoke the utility

### CAUTION

When a program that allows an inquiry request is interrupted, the execution of that program is suspended, permitting the execution of other programs. However, if these other programs alter the status of the system or the status files, the effect may be abnormal termination of the interrupted program or erroneous results when the interrupted program regains control. If you are using inquiry, *do not change* any System/32 files or document library members that were being used by the interrupted (rolled-out) program. System/32 system control programming does not always check for duplicate file labels in the inquiry and interrupted programs.

See *IBM System/32 System Control Programming Reference Manual*, GC21-7593, for a more detailed description of the inquiry mode.

Do not create a document with a blank member name. If you create a document with a blank member name, you will not be able to delete it with a WPFREE command without deleting the entire group of documents.

When using OCL to evoke the utilities, as opposed to the procedures, the user must specify the *libname* specified in the initialization of the document library in the following manner:

1. For the document directory file LABEL, specify the *libname* and add DDO to the end of it.
2. For the document storage file LABEL, specify the same *libname* and add DSO to the end of it.



## \$WPKBD—KEYBOARD REDEFINE UTILITY PROGRAM

Each country may redefine to their corresponding keyboard.

\$WPKBD provides a means of changing the keyboard/display screen character formats.

\$WPKBD is evoked by the KEYBOARD procedure (see index entry: *KEYBOARD procedure*).

Selective keys are redefined to match those on a particular typewriter. Thus by specifying 002 in the KEYBOARD procedure, the keyboard/display screen is defined such that the character format is that of the American Standard 101-A keyboard. (See Figures 1 and 2.)

### \$WPKBD OCL and Utility Control Statement Format

```
// LOAD $WPKBD
// RUN
// KBID nnn
// END
```

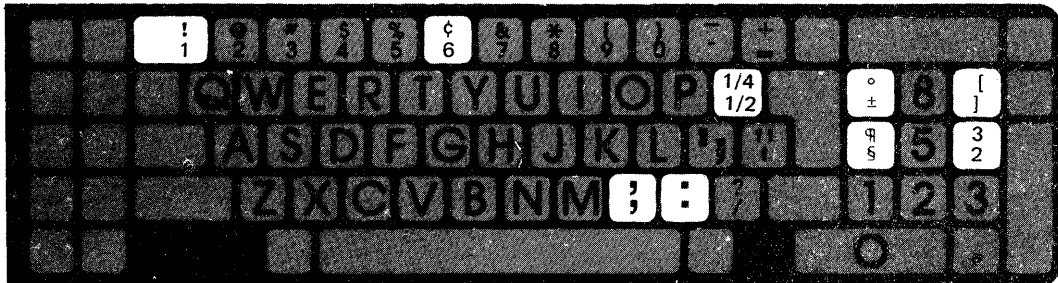
### \$WPKBD Control Statement Parameters

KBID statement:

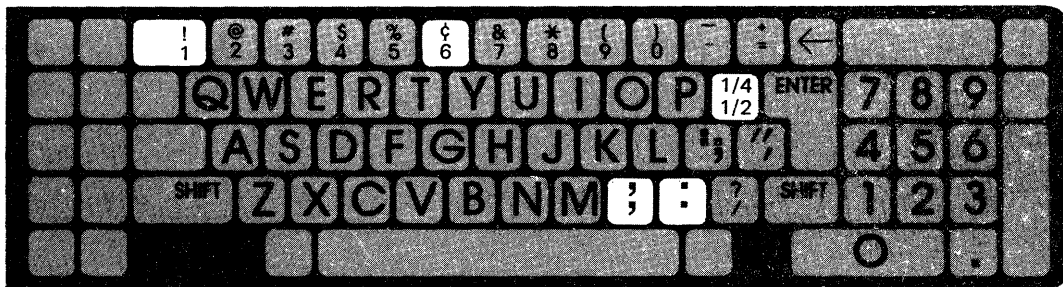
nnn	Keyboard
001	IBM Office System 6 96-character keyboard
002	American Standard 101-A keyboard
003	American Standard 101-B keyboard
004	American Standard 101-C keyboard
005	American Standard 101-D keyboard
007	Legal 149 keyboard
008	Legal 177 keyboard
009	Pica 046 keyboard
017	Accounting 172 keyboard
018	Accounting 174 keyboard
026	Austrian/German 88-character keyboard
027	Austrian/German 96-character keyboard
028	Austrian/German 88-character keyboard
029	Austrian/German 96-character keyboard
031	French 96-character keyboard
032	French 88-character keyboard
041	Italian 96-character keyboard
046	Italian 88-character keyboard
056	Danish 88-character keyboard
057	Danish 96-character keyboard
066	British 88-character keyboard
067	British 96-character keyboard

nnn	Keyboard (continued)
100	American System/32 Monocase EBCDIC
202	Danish System/32 Monocase EBCDIC
204	French System/32 Monocase EBCDIC (AZERTY)
205	French System/32 Monocase EBCDIC (QWERTY)
206	Austrian/German System/32 Monocase EBCDIC (QWERTZ)
207	Italian System/32 Monocase EBCDIC
212	British System/32 Monocase EBCDIC
250	Modified System/32 Dual Case EBCDIC

**American Keyboards**



**Figure 1. Keyboard 001—IBM Office System 6 96-Character**



**Figure 2. Keyboard 002—American Standard 101-A**

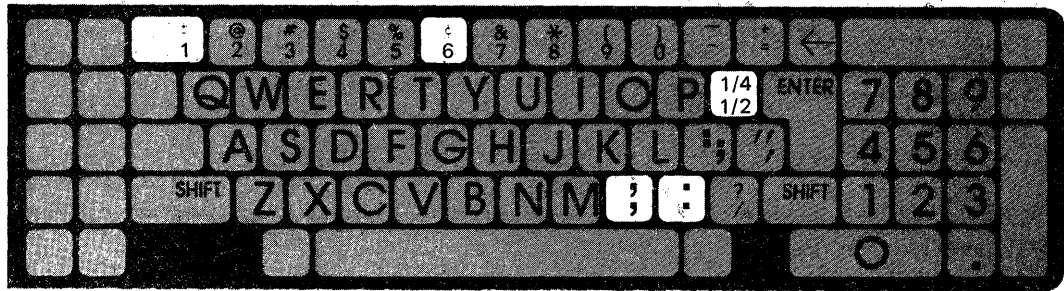


Figure 3. Keyboard 003—American Standard 101-B

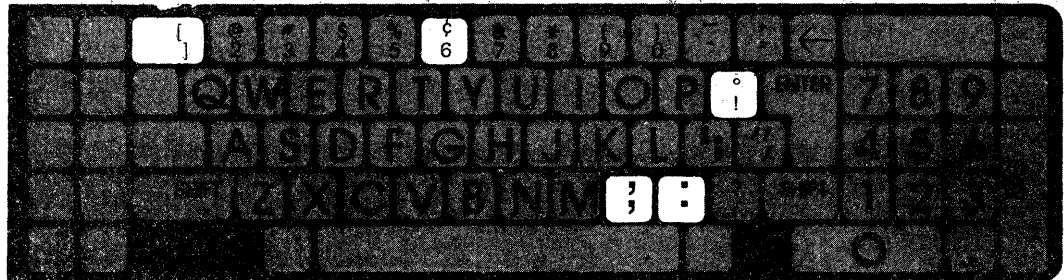


Figure 4. Keyboard 004—American Standard 101-C

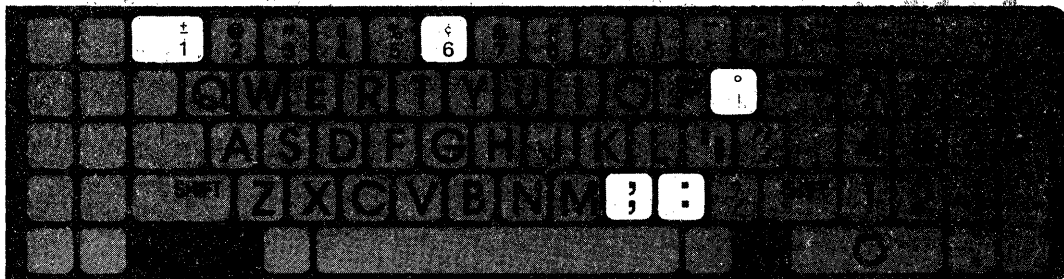


Figure 5. Keyboard 005—American Standard 101-D

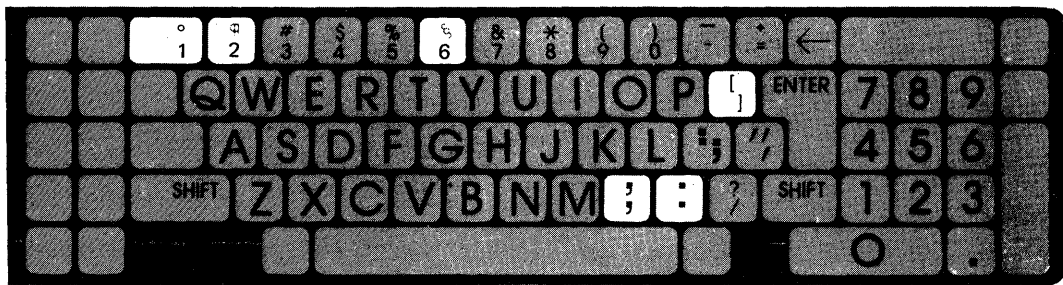


Figure 6. Keyboard 007-Legal 149

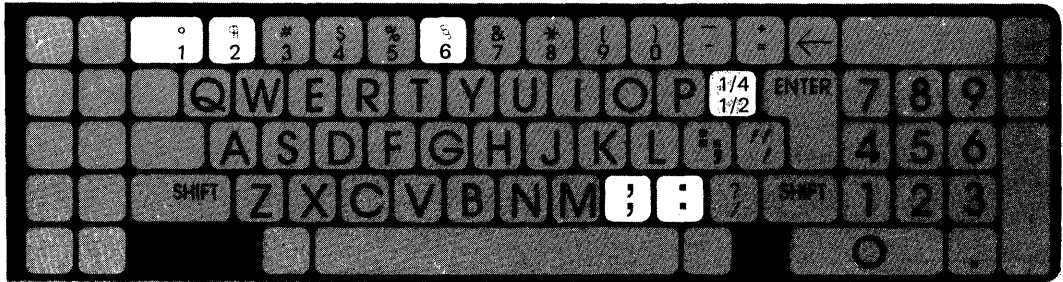


Figure 7. Keyboard 008-Legal 177

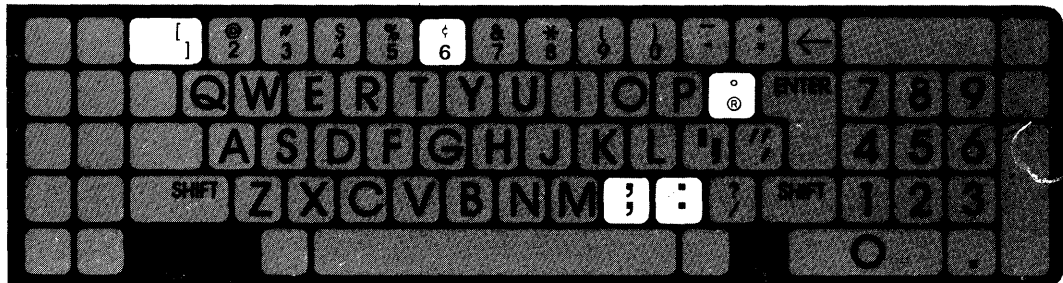
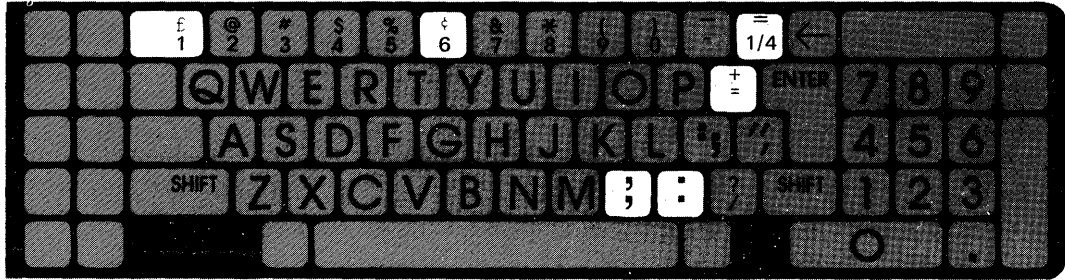
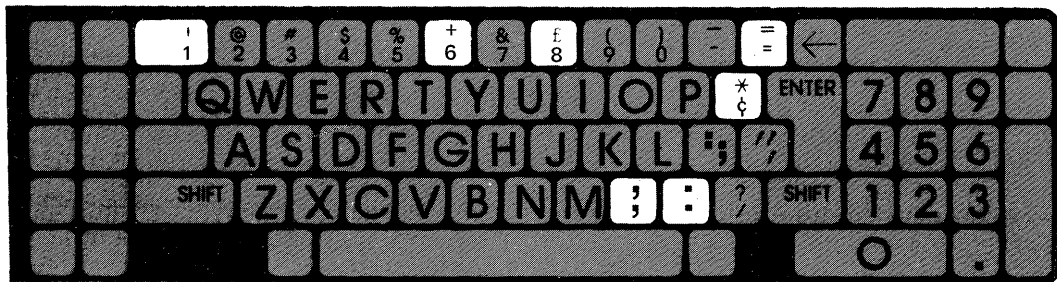


Figure 8. Keyboard 009-Pica 046



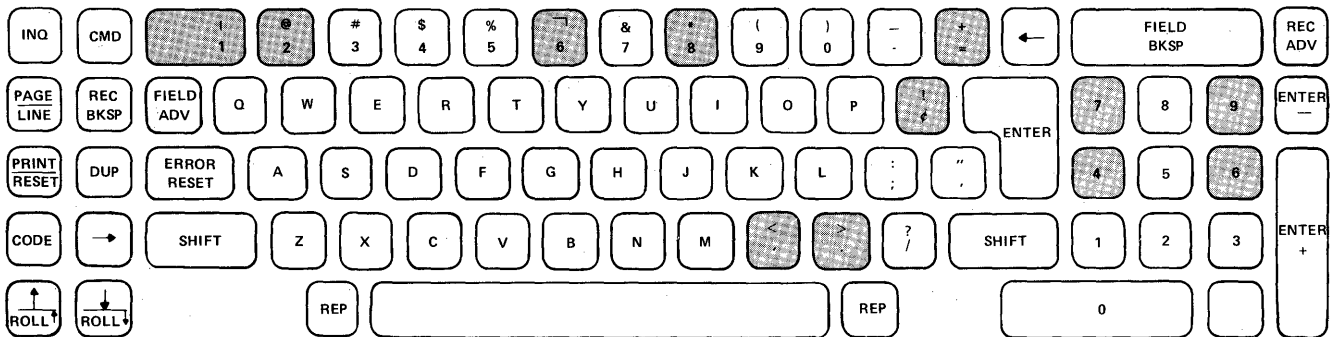
Note: The double underscore is not on any of the line printer belts and therefore prints as blank. The double underscore appears as a single underscore on the display screen.)

Figure 9. Keyboard 017—Accounting 172



Note: The double underscore is not on any of the line printer belts and therefore prints as blank. The double underscore appears as a single underscore on the display screen.

Figure 10. Keyboard 018—Accounting 174



Note: To print <, >, |, and ~, you must have a monospace print belt on the printer. All other characters print using the 96-character dual case print belt.

Figure 11. Keyboard 100—American System/32 Keyboard

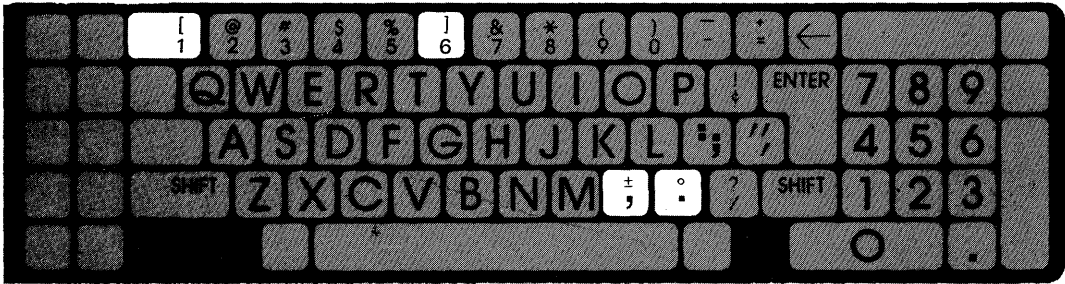


Figure 12. Keyboard 250—Modified American System/32 Dual Case EBCDIC

## Austrian/German Keyboards

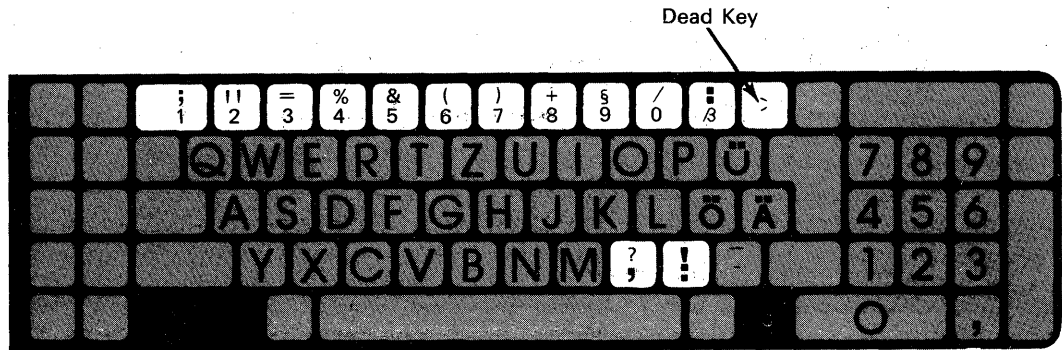


Figure 13. Keyboard 026—Austrian/German 88-character

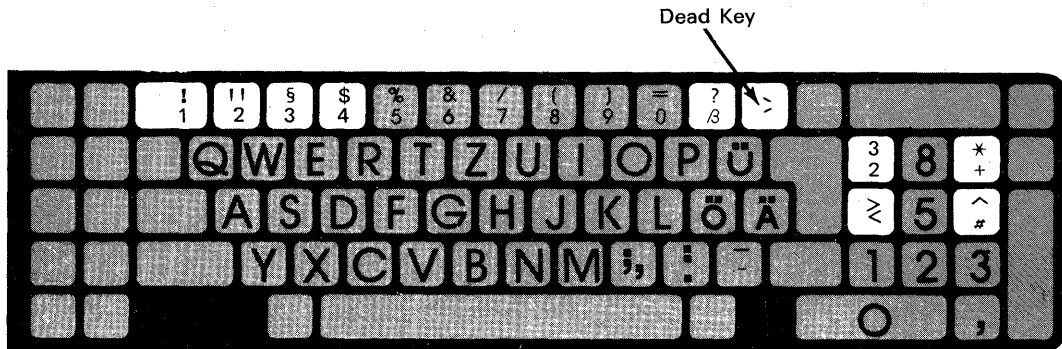


Figure 14. Keyboard 027—Austrian/German 96-character

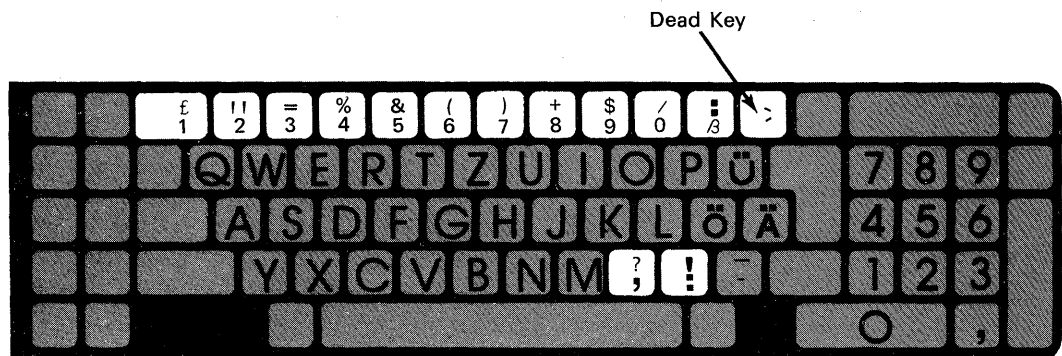
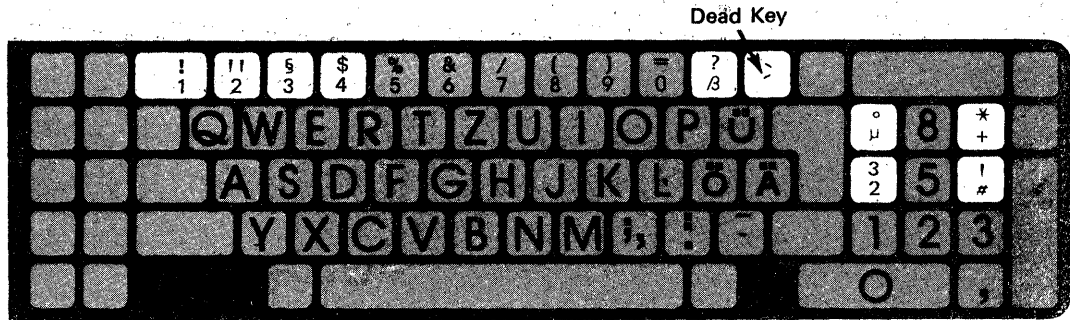


Figure 15. Keyboard 028—Austrian/German 88-character



Note: The ° and μ display as > and <, respectively.

Figure 16. Keyboard 029—Austrian/German 96-character

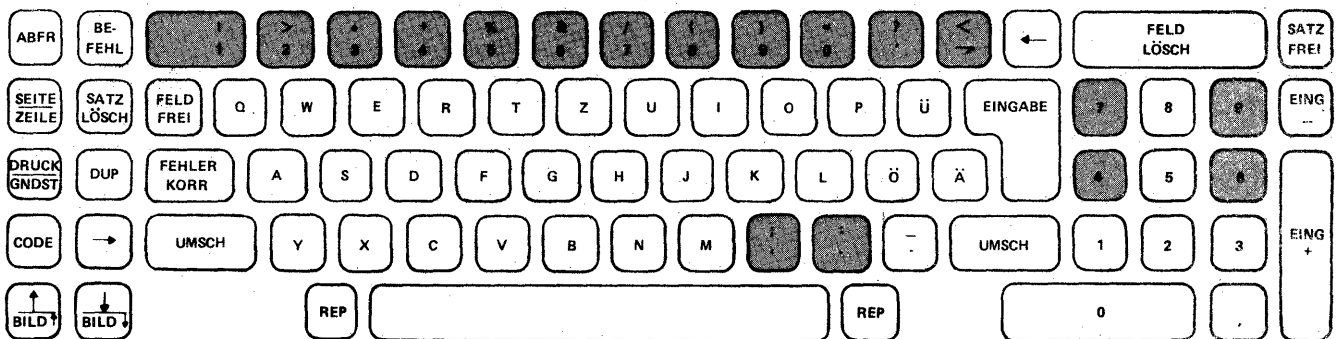


Figure 17. Keyboard 206—Austrian/German System/32 Keyboard



## French Keyboards

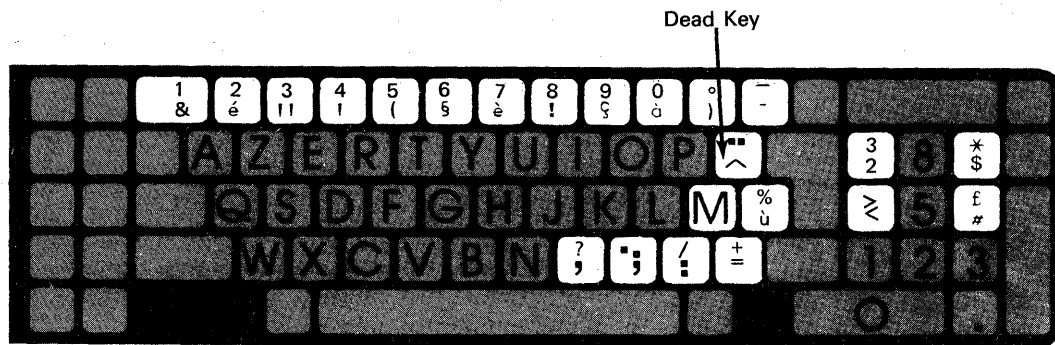


Figure 18. Keyboard 031—French 96-character

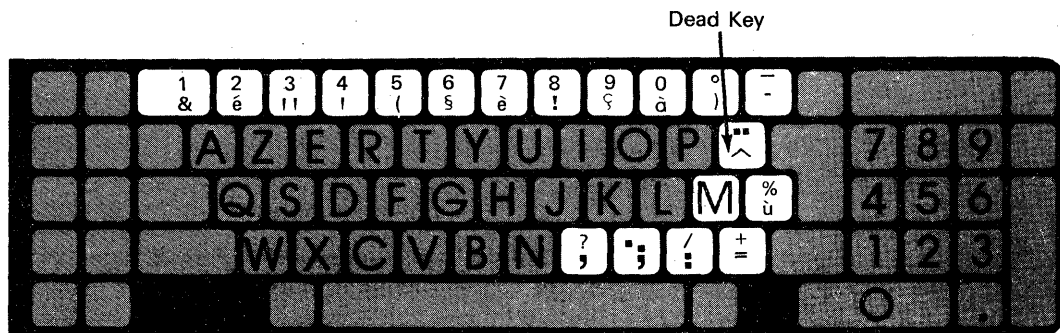


Figure 19. Keyboard 032—French 88-character

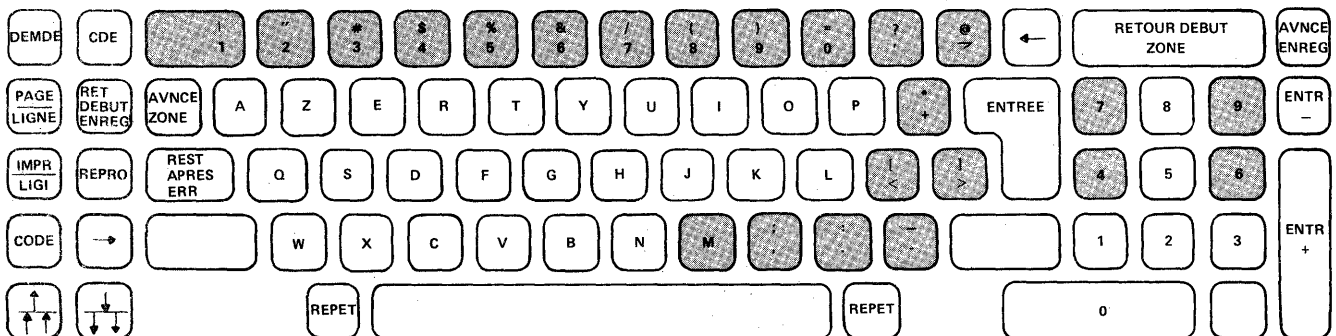


Figure 20. Keyboard 204—French System/32 (AZERTY)

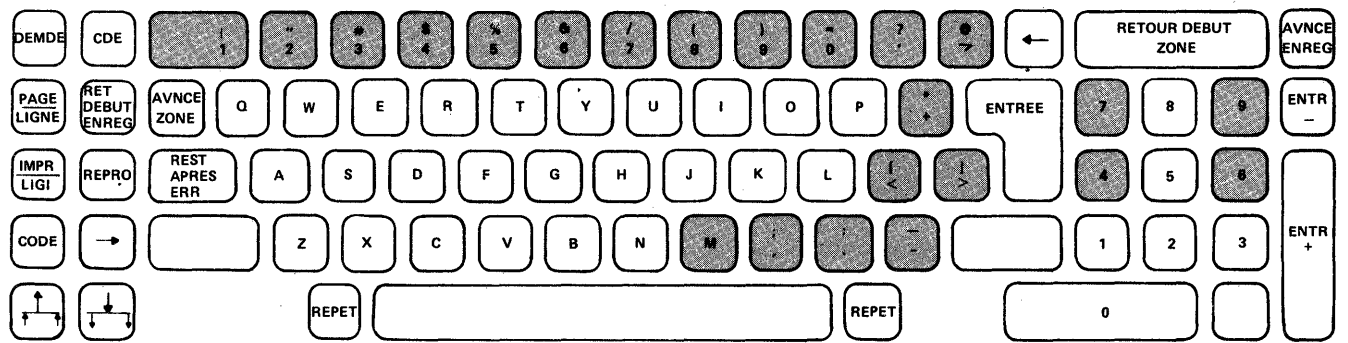


Figure 21. Keyboard 205–French System/32 (QWERTY)

**Italian Keyboards**

Dead Key

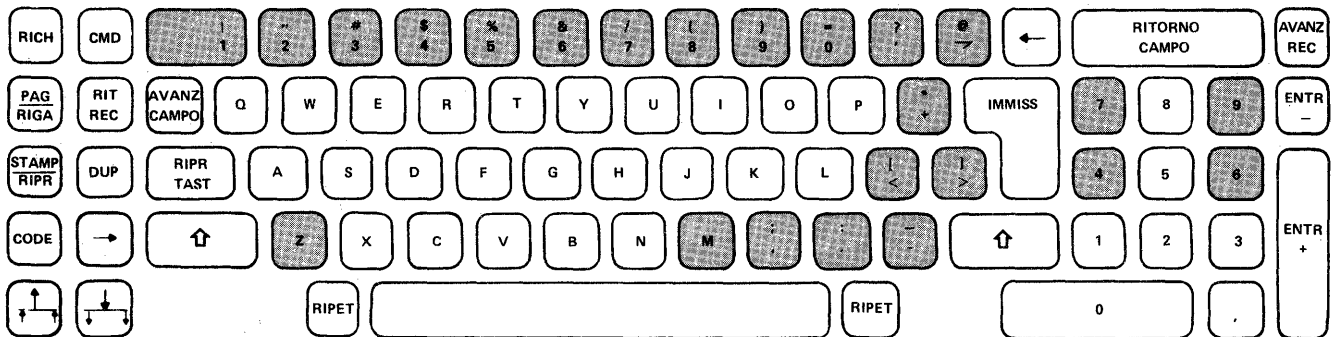


**Figure 22. Keyboard 041-Italian 96-character**

Dead Key



**Figure 23. Keyboard 046-Italian 88-character**



**Figure 24. Keyboard 207-Italian System/32**

## Danish Keyboards

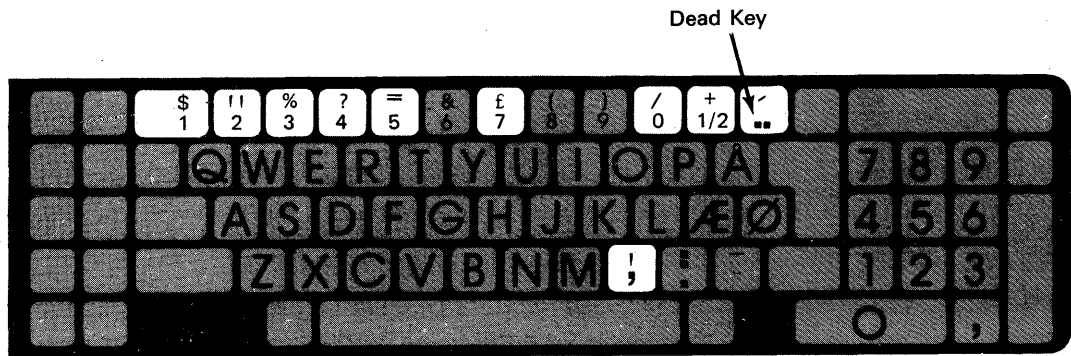


Figure 25. Keyboard 056—Danish 88-character

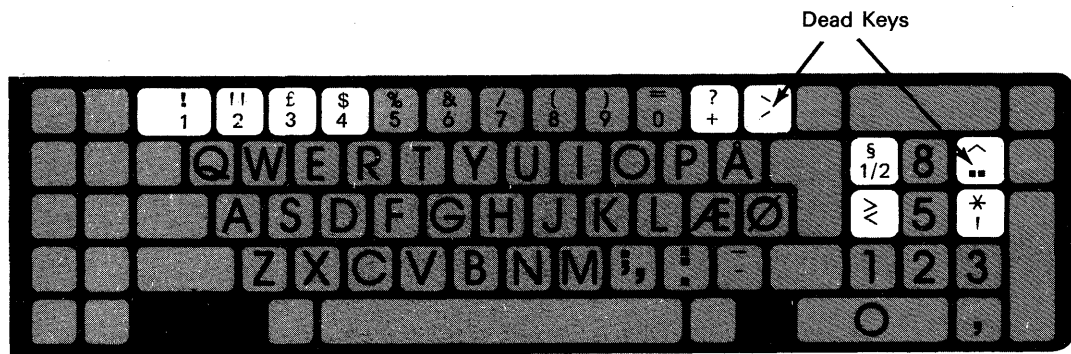


Figure 26. Keyboard 057—Danish 96-character

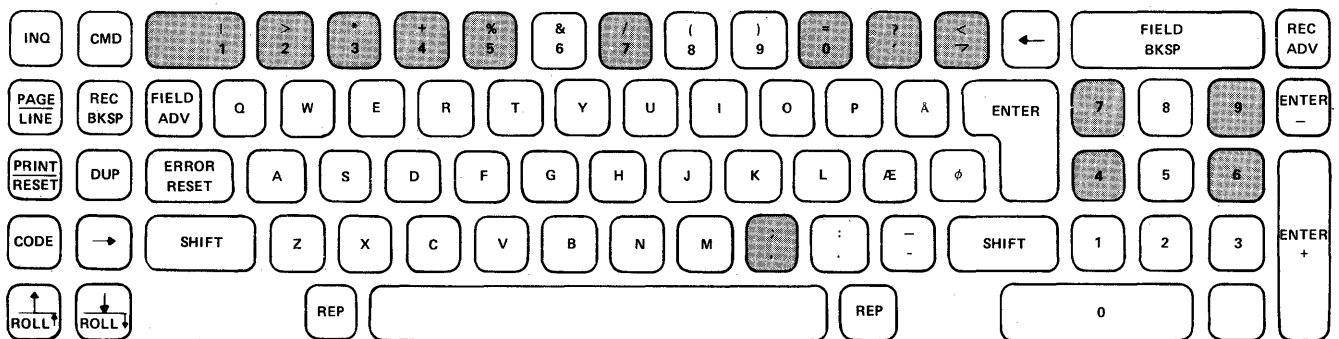


Figure 27. Keyboard 202—Danish System/32

## British Keyboards

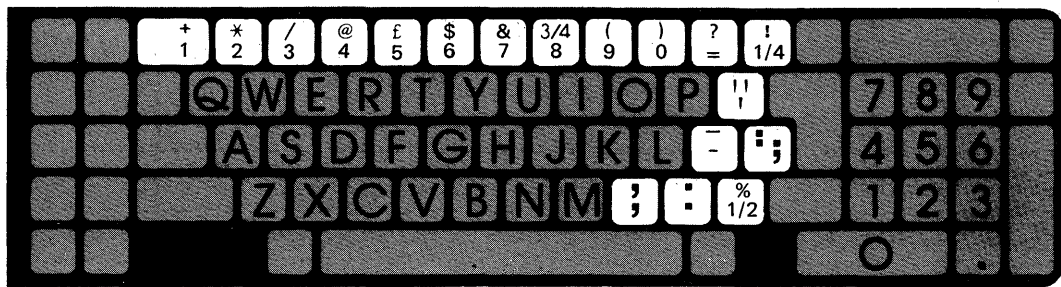


Figure 28. Keyboard 066—British 88-character

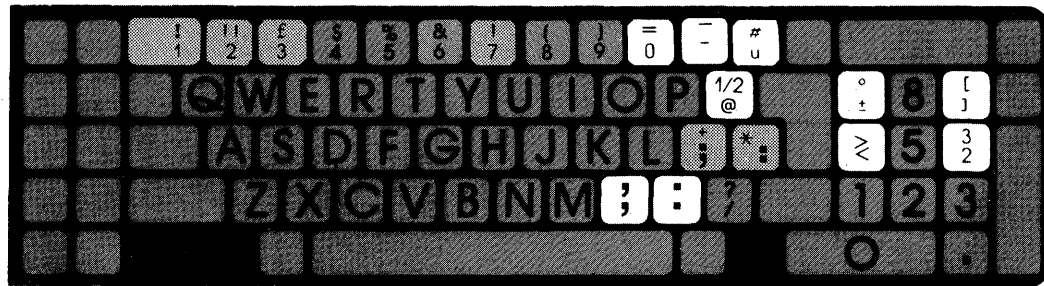


Figure 29. Keyboard 067—British 96-character

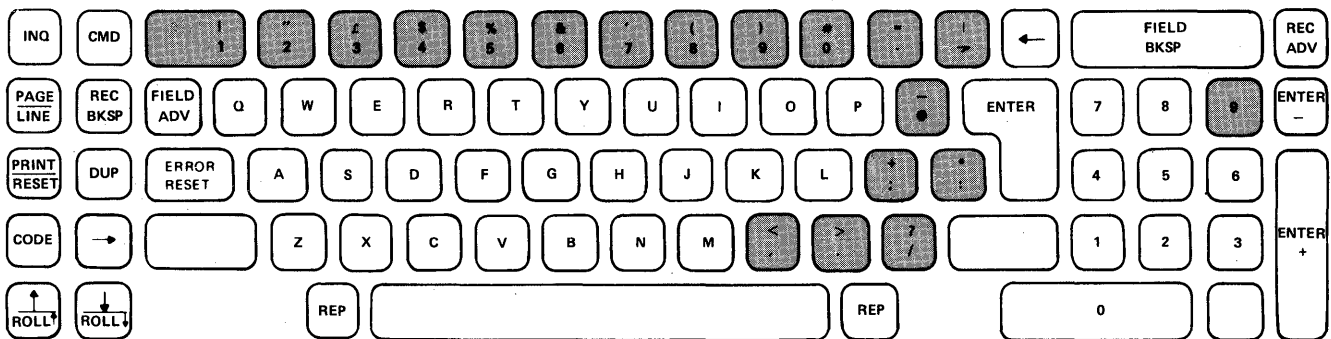


Figure 30. Keyboard 212—British System/32

## **\$WPKEY-SECURITY FILE UPDATE UTILITY PROGRAM**

The \$WPKEY utility allows the user to update the security key file. The utility is secured by an access key which resides in the security file itself. A user may not update or list the security file using this utility without providing the correct access key. (The user is prompted for it). The exception to this is at initial load of word processing. At that time there is no access key. In this case, the access key entered when the utility is first used becomes the access key of the file.

\$WPKEY is evoked by the WPKEY procedure (see index entry: *WPKEY procedure*).

The utility provides the following functions:

*Add a key*—A security key and its corresponding access map are added to the file. The access map specifies the security classes to which jobs with this key belong. A total of 127 keys can be added to a file.

*Change a key*—Changes a security key without changing the access map of the original.

*Delete a key*—Deletes a key from the file.

*Respecify security classes*—Allows the user to replace the access map for a particular key, that is, to respecify the security classes for that key.

*List the file*—The contents of the file in main storage at the time of the list request are printed.

*Delete the file*—All keys in the file are deleted except for the access key.

*Tracking*—A list can be generated that contains for each prompt, the first line of the prompt, the response made to the prompt, and the error message generated, if any. This tracking list may be enabled or disabled by the user of the utility.

*Change the access key*—The access key may be changed after access to the utility is achieved.

The sequence of events within this utility is:

1. Read the security file into main storage.
2. Request and verify the access key.
3. Prompt the user for updates. Make updates to the file in main storage.
4. Write the file to disk.

Since the file is not updated on the disk until end of job, the program will terminate with no disk update if the Inquiry key is pressed and option 2 is taken before end of job.

All user data is entered by responding to prompts on the keyboard/display screen.

Figure 31 is a guide to the use of the prompts.

#### **\$WPKEY OCL and Control Statement Sequence**

```
// LOAD $WPKEY  
// RUN
```

If the prompt is:		And you enter:	You have:	Next prompt:
1	ENTER ACCESS KEY	Access key	Entered a valid access key <sup>1</sup>	2
			Entered an invalid access key <sup>2</sup>	1
		ENTER	Cancelled the job	-
2	SECURITY FILE OPTIONS	A	Selected the change access key option	3
		U	Selected the key update option	5
		D	Selected the file delete option	4
		L	Caused the file to be listed	2
		T	Enabled tracking if off; disabled tracking if on	2
		E	Written the file to diskette and ended the job	-
Other	Entered an invalid option	2		
3	ENTER ACCESS KEY	Access key	Changed the file key	2
		ENTER	cancelled the change	2
4	CONFIRM FILE DELETE	Y	Deleted all keys except the access key <sup>3</sup>	2
		Other	Done nothing	2
5	KEY UPDATE OPTIONS	A	Selected the add option	6a
		C	Selected the change key option	6c
		D	Selected the delete a key option	6d
		R	Selected the respecify classes option	6r
		E	Ended key update	2
		Other	Entered an invalid option	5
6a	ENTER KEY	Security key	Tried to add to a full file (127 keys)	6a
		Security key	Tried to add a duplicate key	6a
		Security key	Added a key with no security classes	7
		ENTER	Cancelled the add	5
6c	ENTER KEY	Security key	Tried to change a key that is not there	6c
		Security key	Specified the key to be changed	8
		ENTER	Cancelled the change	5
6d	ENTER KEY	Security key	Tried to delete a key that is not in the file	6d
		Security key	Deleted the key from the file	5
		ENTER	Cancelled the delete	5
6r	ENTER KEY	Security key	Tried to respecify a key that is not in the file	6r
		Security key	Deleted all security classes for this key	7
		ENTER	Cancelled the respecify	5
7	ENTER A SECURITY CLASS	Security class	Entered an invalid security class	7
		Security class	Associated a security class with a key	7
		ENTER	Completed entering security classes	5
8	ENTER NEW KEY	Security key	Tried to change to a duplicate key	8
		Security key	Changed the key to the specified key	5
		ENTER	Cancelled the change	5

<sup>1</sup> If this is the initial use of the utility, the key entered becomes the access key of the file.  
<sup>2</sup> If an invalid key is entered three times, the program terminates.  
<sup>3</sup> All keys except the access key are deleted. Be sure this is what you want before selecting this option.

Figure 31. Security File Prompts



## \$WPLUF-DELETE DOCUMENT LIBRARY MEMBER UTILITY PROGRAM

\$WPLUF deletes members from the document library based on the utility control statements entered and the utility security check performed. A listing of the members deleted and accumulated totals is produced for user verification of the deleted members. Figure 32 shows a sample listing.

First Page:

```
$WPLUF          DOCUMENT DIRECTORY DATA LIST      06/20/77

      MEMBER NAME      TOTAL      MEMBER      CREATE
      GROUP  DOCC      STOR RCDS  STATUS      DATE
      -----
      GROUP  DOCC          1          F      77/06/20
```

Second Page:

```
ACCUMULATED MEMBER TOTALS

      MEMBERS STATUS F (DELETED)          =          1
      NUMBER STORAGE RECORDS USED        =          1

FREE QUEUE TOTALS

      DIRECTORY RECORDS ALLOCATED         =          80
      DIRECTORY RECORDS REMAINING         =          75
      STORAGE RECORDS ALLOCATED           =          80
      STORAGE RECORDS REMAINING           =          77
```

Figure 32. Sample \$WPLUF Listing

This utility enables the reuse of document library space by changing the document directory record status to deleted and chaining the document storage records back to the free queue. However, the document directory record can only be reused if the same member name is used. If the document directory record is to be reused under a different name, see WPPURGE procedure. The document storage records can be reused by any member.

\$WPLUF is evoked by the WPFREE procedure (see index entry: *WPFREE procedure*).

## \$WPLUF Utility Control Statement Formats

Use	Control Statements
Enter access key	// KEY kkkk
Enter type of selection	// MODE { ALL ACCEPT REJECT }
If MODE is ACCEPT/ REJECT, enter SPECIFY statements	// SPECIFY [GROUP-xxxxxxx] [,DOCNAME-xxxxxxx] [,FROM-yyymmdd] [,TO-yyymmdd]  // END

Note: If the SPECIFY statement is used, at least one of the parameters must be included.

## \$WPLUF Control Statement Parameters

### KEY Statement

kkkk Specifies the security key to be used in accessing members to be freed. NULL indicates that only unsecured documents will be processed. Only those document library members that pass the document security check will be processed.

### MODE Statement

ALL All members are to be processed by the utility. If the MODE statement is not specified, it defaults to ALL.

ACCEPT Only those members which satisfy at least one of the conditions given in SPECIFY statements following the MODE statement are to be processed.

REJECT Only those members which satisfy none of the conditions given in SPECIFY statements following the MODE statement are to be processed.

## SPECIFY Statement

The specify statement is used to give conditions by which individual members are to be accepted or rejected for processing by the utility. A maximum of 20 SPECIFY statements are allowed. During each execution of the utility, a single scan of the file being processed is made with each member being compared to the specified conditions. A member is accepted (or rejected if MODE is REJECT) if it satisfies at least one of the SPECIFY statements. A statement is satisfied only if all of its parts are satisfied. If members in the document library have a *groupname*, you must specify that *groupname*. If you specify *groupname* without a *docname*, all members with that *groupname* are affected. If members in the document library have a NULL *groupname* (blank qualifier with *docname* only), the *docname* must be specified. FROM dates and TO dates can only be used when *groupname* and/or *docname* are specified.

GROUP-xxxxxxx	Specifies the first eight characters of the member name.
DOCNAME- xxxxxxx	Specifies the last eight characters of the member name. This parameter is normally used with the GROUP parameter, but need not be. If either the GROUP or the DOCNAME parameter is less than eight characters, blanks are automatically filled on the right. Blanks are not allowed in the middle of the parameter.
FROM-yymmdd	Specifies that all members with a create date equal to or greater than this parameter are accepted (or rejected if MODE is REJECT). The parameter must be in the form yymmdd.
TO-yymmdd	Specifies that all members with a create date equal to or less than this parameter are to be accepted (or rejected if MODE is REJECT). The parameter must be in the form yymmdd.

## \$WPLUF OCL and Control Statement Sequence

To initiate \$WPLUF through OCL, enter the following:

```
// LOAD $WPLUF
// FILE NAME-WPLDDO[,LABEL-filename-1]
// FILE NAME-WPLDSO[,LABEL-filename-2]
// RUN
// KEY kkkk
// MODE      { ALL
              { ACCEPT
              { REJECT
// SPECIFY   [GROUP-xxxxxxx] [,DOCNAME-xxxxxxx]
              [,FROM-yymmdd] [,TO-yymmdd]
// END
```

**Note:** If the SPECIFY statement is used, at least one of the parameters must be included.

## **\$WPLUF OCL Parameters**

**LABEL-filename-1** Specifies the name assigned to the document directory file of the document library on disk. This must be five or less alphameric characters (the first one must be alphabetic) with DDO added to the end of it.

**LABEL-filename-2** Specifies the name assigned to the document storage file of the document library on disk. This must be the same alphameric characters as were used in *filename-1* but with DSO on the end.

## **\$WPLUF Examples**

To delete the member with GROUP DEPT283 and DOCNAME PROPOSAL from the document library DOCUM within the security classifications specified by the key WEEK, you would enter:

```
// LOAD $WPLUF
// FILE NAME=WPLDDO,LABEL=DOCUMDDO
// FILE NAME=WPLDSO,LABEL=DOCUMDSO
// RUN
// MODE ACCEPT
// KEY WEEK
// SPECIFY GROUP=DEPT283,DOCNAME=PROPOSAL
// END
```

Delete all members that have the GROUP ACCTG and were created in October, 1970. Members are not classified.

```
// LOAD $WPLUF
// FILE NAME=WPLDDO
// FILE NAME=WPLDSO
// RUN
// KEY NULL
// MODE ACCEPT
// SPECIFY GROUP=ACCTG,FROM=701001,TO=701031
// END
```

Delete all members except those created since January 1, 1975. Members are not classified.

```
// LOAD $WPLUF
// FILE NAME=WPLDDO
// FILE NAME=WPLDSO
// RUN
// KEY NULL
// MODE REJECT
// SPECIFY FROM=750101
// END
```

## **\$WPLUI—DOCUMENT LIBRARY INITIALIZATION UTILITY PROGRAM**

The linked direct access method (LDAM) requires that a document library exist. \$WPLUI allows the user to create an initialized document library.

The user specifies the size of the document directory and document storage files. He may also name the files or accept the default values.

In the document library created, all storage space will be chained into the free queue.

\$WPLUI is evoked by the WPINIT procedure (see index entry: *WPINIT*).

### **\$WPLUI OCL and Control Statement Sequence**

```
// LOAD $WPLUI
// FILE NAME=WPLDDO[,LABEL-filename-1],RECORDS-number-1,
  RETAIN-P
// FILE NAME=WPLDSO[,LABEL-filename-2],RECORDS-number-2,
  RETAIN-P
// RUN
```

### **\$WPLUI OCL Parameters**

**LABEL-filename-1** Specifies the name to be assigned the document directory file of the document library on disk. This must be five or less alphameric characters (the first one must be alphabetic) with DDO appended to it.

**LABEL-filename-2** Specifies the name to be assigned to the document storage file of the document library. This must be the same alphameric characters as were used in *filename-1*, with DSO appended to it.

**RECORDS-number-1** Specifies the number of document directory records to allocate. It equals the maximum number of documents that will be stored in the library. Document directory records are allocated in blocks of 80 records. If the number you enter is not a multiple of 80, the system will automatically increase it to the next even multiple of 80.

**RECORDS-number-2** Specifies the number of documents storage records to allocate. It equals the number of 250 character records required to store the text of all documents in the library. Document storage records are allocated in blocks of ten records. If the number you enter is not a multiple of 10, the system increases the number to the next higher multiple of ten.

### \$WPLUI Example

A user wishes to create a document library that can store 200 documents each of which will require an average of five document storage records. The default file names are to be used.

```
// LOAD $WPLUI
// FILE NAME=WPLDDO,RECORDS=200,RETAIN=P
// FILE NAME=WPLDSO,RECORDS=1000,RETAIN=P
// RUN
```

### \$WPLUL-LOAD DOCUMENT LIBRARY MEMBERS FROM DISKETTE UTILITY PROGRAM

The load utility adds selected document library members from a diskette created by the unload utility (\$WPLUU) to an existing document library.

Members are added to a document library if they are not already there or if a member already exists in the library and override is specified. If the member already exists and override is not specified, no action occurs.

If the library to which members are being added is an empty library created by \$WPLUI, the resultant library is organized that is, directory records are in collating sequence by member name and the storage records for each member are in consecutive sectors.

The listing produced from this utility is the same format as that from the unload utility (\$WPLUU) and can be compared with the unload list to verify the load process. Figure 33 shows a sample listing.

```
$WPLUL      DOCUMENT DIRECTORY DATA LIST                               06/20/77

      MEMBER NAME                TOTAL      MEMBER      CREATE
      GROUP   DOCA                STOR RCDS  STATUS      DATE
      GROUP   DOCB                1          1          77/06/20
      GROUP   DOCC                1          1          77/06/20
      GROUP   DOCD                1          1          77/06/20

      MEMBERS STATUS 0 (IN USE)                =          4

      NUMBER STORAGE RECORDS USED            =          4
```

Figure 33. Sample \$WPLUL Listing

\$WPLUL is evoked by the WPRESTOR procedure (see index entry: WPRESTOR).

## \$WPLUL Utility Control Statement Formats

Use	Control Statement	
Enter override	// OVERRIDE	{ YES } { NO }
Enter type of selection	// MODE	{ ALL } { ACCEPT } { REJECT }
If MODE is ACCEPT/REJECT, enter SPECIFY statements	// SPECIFY  // END	[GROUP-xxxxxxx] [,DOCNAME-xxxxxxx] [,FROM-yyymmdd] [,TO-yyymmdd]

Note: If the SPECIFY statement is used, at least one of the parameters must be included.

## \$WPLUL Utility Control Statement Parameters

### OVERRIDE Statement

YES	If the member being loaded from diskette is already in the document library, override it. That is, delete the existing member and add the member from diskette.
NO	Retain the member currently in the document library. If the OVERRIDE statement is not specified, it defaults to NO.

### MODE Statement

ALL	All members are to be processed by the utility. If the MODE statement is not specified, it defaults to ALL.
ACCEPT	Only those members which satisfy at least one of the conditions given in SPECIFY statements following the MODE statement are to be processed.
REJECT	Only those members which satisfy none of the conditions given in SPECIFY statements following the MODE statement are to be processed.

## SPECIFY Statement

The specify statement is used to give conditions by which individual members are to be accepted or rejected for processing by the utility. A maximum of 20 SPECIFY statements are allowed. During each execution of the utility, a single scan of the file being processed is made with each member being compared to the specified conditions. A member is accepted (or rejected if MODE REJECT) if it satisfies at least one of the SPECIFY statements. A statement is satisfied only if all of its parts are satisfied.

- GROUP-xxxxxxx** Specifies the first eight characters of the member name.
- DOCNAME-xxxxxxx** Specifies the last eight characters of the member name. This parameter is normally used with the GROUP parameter, but need not be. If either the GROUP or the DOCNAME parameter is less than eight characters, blanks are automatically filled on the right. Blanks are not allowed in the middle of the parameter list.
- FROM-yymmdd** Specifies that all members with a create date equal to or greater than this parameter are accepted (or rejected if MODE REJECT). The parameter must be in the form yymmdd. The default is 000000.
- TO-yymmdd** Specifies that all members with a create date equal to or less than this parameter are to be accepted (or rejected if MODE REJECT). The parameter must be in the form yymmdd. The default is 999999.

## \$WPLUL OCL and Control Statement Sequence

To initiate the load utility through OCL, enter the following:

```
// LOAD $WPLUL
// FILE NAME-WPLDDO[,LABEL-filename-1]
// FILE NAME-WPLDSO[,LABEL-filename-2]
// FILE NAME-WPLBUO,UNIT-I1[,LABEL-filename-3]
// RUN
// OVERRIDE      { YES }
                 { NO  }
// MODE          { ALL  }
                 { ACCEPT }
                 { REJECT }
// SPECIFY      [GROUP-xxxxxxx] [,DOCNAME-xxxxxxx]
                [,FROM-yymmdd] [,TO-yymmdd]
// END
```

**Note:** If the SPECIFY statement is used, at least one of the parameters must be included.



### **\$WPLUL OCL Parameters**

**LABEL-filename-1** Specifies the name assigned to the document directory file of the document library on disk. This must be the libname specified when the document library was initialized, appended with DDO.

**LABEL-filename-2** Specifies the name assigned to the document storage file of the document library. This must be the libname specified when the document library was initialized, appended with DSO.

**LABEL-filename-3** Specifies the name of the input file on diskette.

### **\$WPLUL Examples**

Load the member with GROUP of FINANCE and DOCNAME PROJECT to the document library SALES from input diskette file STORIT. Also select and load members with GROUP ACCT and created between January 1, 1976 and June 30, 1976. Replace members already existing on library (override is specified).

```
// LOAD $WPLUL
// FILE NAME=WPLDDO,LABEL=SALESDDO
// FILE NAME=WPLDSO,LABEL=SALESDSO
// FILE NAME=WPLBUO,UNIT-I 1,LABEL=STORIT
// RUN
// OVERRIDE YES
// MODE ACCEPT
// SPECIFY GROUP=FINANCE,DOCNAME=PROJECT
// SPECIFY GROUP=ACCT,FROM=760101,TO=760630
// END
```

Load all members on diskette to document library using default options (no override specified, keeping existing members).

```
// LOAD $WPLUL
// FILE NAME=WPLDDO
// FILE NAME=WPLDSO
// FILE NAME=WPLBUO,UNIT-I 1
// RUN
// MODE ALL
// END
```

## **\$WPLUM-LDAM MAINTENANCE UTILITY PROGRAM**

\$WPLUM is evoked as a procedure within the word processing environment under job control.

\$WPLUM provides the following functions to be performed on documents in the document library:

- Change the security classification of a document (REKEY)
- Change a document name (RENAME)
- Delete a document (DELETE)

Note: \$WPLUM can be input via valid word processing SYSIN devices (keyboard, mag card unit, fixed disk) only. The document library cannot be used for \$WPLUM input.

### **Rekey (Change the Security Classification of a Document)**

The access number specified on the utility control statement replaces the access number currently in the document library for the specified member. A document security check is made to assure user access to the specified member. The new access code is checked to see if it is valid for the security key used in the job statement.

### **Rename**

This function allows the user to change a document name. A document security check is made to assure the user is able to access the specified member being renamed.

### **Delete Member**

The specified member is deleted from the library. The sectors used for storage of the data are returned to the free queue. Erase can be specified on a delete operation. Erase causes the sector(s) used for storage of data to be overwritten with blanks before returning them to the free queue. A document security check is made to assure user access to the specified member being deleted.

### **\$WPLUM Sysin Format**

```
,,JOB [ID-xxxx] [,KEY-xxxx]
,,WPLMAINT
// REKEY [GROUP-xxxxxxxx,] DOCNAME-xxxxxxxx,ACCESS-nn
// RENAME [FRGROUP-xxxxxxxx,] FRDOCNM-xxxxxxxx,
  [TOGROUP-xxxxxxxx,] TODOCNM-xxxxxxxx
// DELETE [FRGROUP-xxxxxxxx,] FRDOCNM-xxxxxxxx [,ERASE-Y]
// END
```

Where:

„JOB—is a valid JOB statement in the word processing stream with a valid key field.

„WPLMAINT—is the name of the LDAM maintenance utility procedure that is to be executed in the job stream.

// REKEY, // RENAME, // DELETE—are three valid control statements with their parameters. There can be any number of control statements in any order.

// END—is the control statement that ends the LDAM maintenance utility.

Note: Following this last control statement, you can:

- Issue a request for a new procedure under the present job to execute.
- Issue a new JOB statement to start a new job under job control.

### **\$WPLUM Control Statement Format**

REKEY:

```
// REKEY [GROUP-xxxxxxxx] ,DOCNAME-xxxxxxxx ,ACCESS-nn
```

RENAME:

```
// RENAME [FRGROUP-xxxxxxxx ,]FRDOCNM-xxxxxxxx ,  
[TOGROUP-xxxxxxxx ,]TODOCNM-xxxxxxxx
```

DELETE:

```
// DELETE [FRGROUP-xxxxxxxx] ,FRDOCNM-xxxxxxxx  
[ ,ERASE-Y]
```

Notes:

1. All parameters are keyword (the position in the statement is not important, just the identifying word). Hyphens following the keywords must be a *required* hyphen when entering control statements in a mag card unit job stream. This restriction applies only to \$WPLUM.
2. ACCESS is a valid number between and including 00 and 32. It identifies the security class to which the document is assigned.
3. GROUP and DOCNAME are limited to a maximum of eight characters in length and are padded to the right with blanks. No embedded blanks are allowed.
4. FRGROUP,FRDOCNM,TOGROUP, and TODOCNM are limited to a maximum of eight characters in length and are padded to the right with blanks. No embedded blanks are allowed.
5. The document name portion of all control statements is required if the LDAM document name is not blank. Otherwise a MEMBER NOT FOUND condition occurs.
6. No control statement can end with a comma.
7. No group operations are allowed.

## \$WPLUM Control Statement Examples

- Change access key of library member FIRSTMEMNO1 to 32.

```
// REKEY GROUP-FIRSTMEM, DOCNAME-NO1, ACCESS-32
```

- Change access key of library member MEMB2 to 15.

```
// REKEY GROUP-MEMB2, ACCESS-15
```

- Rename library member FIRSTMEMNO1 to LASTONE.

```
// RENAME FRGROUP-FIRSTMEM, FRDOCNM-NO1, TOGROUP-LAST, TODOCNM-ONE
```

- Rename library member LASTONE to ABCDEF.

```
// RENAME FRGROUP-LAST, FRDOCNM-ONE, TOGROUP-ABC, TODOCNM-DEF
```

- Rename library member ABCDEFXY, to ABCDEFGHIJ.

```
// RENAME FRGROUP-ABC, FRDOCNM-XY, TOGROUP-ABCDEFGH, TODOCNM-IJ
```

- Delete library member ABCDEFGHIJ with no erase.

```
// DELETE FRGROUP-ABCDEFGH, FRDOCNM-IJ
```

- Delete library member ABCDEFGHIJ with erase.

```
// DELETE FRGROUP-ABCDEFGH, FRDOCNM-IJ, ERASE-Y
```

## \$WPLUP—DOCUMENT DIRECTORY PRINT UTILITY PROGRAM

This utility allows the user to obtain a listing of the document directory based on the control statements entered and the utility security check performed. The listing contains the member name, the member status, and create date. Accumulated member totals are printed.

A master list is requested by entering MODE ALL and an access key that allows access to all 32 security classes. (See Figure 34.) A master list prints the access number for each member, the accumulated member totals, and file utilization statistics.

First Page:

\$WPLUP DOCUMENT DIRECTORY DATA LIST 06/20/77

MEMBER NAME	TOTAL STOR RCDS	MEMBER STATUS	CREATE DATE
±±FREEQ	76		
GROUP DOCA	1	0	77/06/20
GROUP DOCB	1	0	77/06/20
GROUP DOCC	1	0	77/06/20
GROUP DOCD	1	0	77/06/20

Second Page:

### ACCUMULATED MEMBER TOTALS

MEMBERS STATUS 0 (IN USE)	=	4
NUMBER STORAGE RECORDS USED	=	4

### FREE QUEUE TOTALS

DIRECTORY RECORDS ALLOCATED	=	80
DIRECTORY RECORDS REMAINING	=	74
STORAGE RECORDS ALLOCATED	=	80
STORAGE RECORDS REMAINING	=	76

Figure 34. Sample \$WPLUP Listing

A statistics only list is requested by entering KEY \$\$\$\$\$. For a statistics only list, the program sets up for a master list but prints only the accumulated member totals and file utilization statistics.

\$WPLUP is evoked by the WPPRINT procedure (see index entry: WPPRINT procedure).

## \$WPLUP Utility Control Statement Formats

Use	Control Statement
Enter access key	// KEY kkkk
Enter type of selection	// MODE { ALL ACCEPT REJECT }
If MODE is ACCEPT/REJECT, enter SPECIFY statements	// SPECIFY [GROUP-xxxxxxx] [,DOCNAME-xxxxxxx] [,FROM-yymmdd] [,TO-yymmdd]  // END

*Note:* If the SPECIFY statement is used, at least one of the parameters must be included.

## \$WPLUP Control Statement Parameters

### KEY Statement

kkkk Specifies the security key to be used in accessing members to be processed. NULL indicates that only unsecured documents will be processed. Only those documents library members that pass the document security check will be processed. Use \$\$\$\$ for a statistics only print.

### MODE Statement

ALL All members are to be processed by the utility. If the MODE statement is not specified, ALL is the default.

ACCEPT Only those members which satisfy at least one of the conditions given in SPECIFY statements following the MODE statement are to be processed.

REJECT Only those members which satisfy none of the conditions given in SPECIFY statements following the MODE statement are to be processed.

## SPECIFY Statement

The specify statement is used to give conditions by which individual members are to be accepted or rejected for processing by the utility. A maximum of 20 SPECIFY statements is allowed. During each execution of the utility, a single scan of the file being processed is made with each member being compared to the specified conditions. A member is accepted (or rejected if MODE REJECT) if it satisfies at least one of the SPECIFY statements. A statement is satisfied only if all of its parts are satisfied. If members in the document library have a *groupname*, you must specify that *groupname*. If you specify *groupname* without a *docname*, all members with that *groupname* are affected. If members in the document library have a NULL *groupname* (blank qualifier with *docname* only), the *docname* must be specified. FROM dates and TO dates can only be used when *groupname* and/or *docname* are specified.

- GROUP-xxxxxxx** Specifies the first eight characters of the member name.
- DOCNAME-xxxxxxx** Specifies the last eight characters of the member name. This parameter is normally used with the GROUP parameter, but need not be. If either the GROUP or the DOCNAME parameter is less than eight characters, blanks are automatically filled on the right. Blanks are not allowed in the middle of the parameter list.
- FROM-yyymmdd** Specifies that all members with a create date equal to or greater than this parameter are accepted (or rejected if MODE REJECT). The parameter must be in the form yyymmdd. The default is 000000.
- TO-yyymmdd** Specifies that all members with a create date equal to or less than this parameter are to be accepted (or rejected if MODE REJECT). The parameter must be in the form yyymmdd. The default is 999999.

## \$WPLUP OCL and Control Statement Sequence

To initiate the print utility, enter the following:

```
// LOAD $WPLUP
// FILE NAME=WPLDDO[, LABEL-filename]
// RUN
// KEY kkkk

// MODE      { ALL
              { ACCEPT
              { REJECT
// SPECIFY   [GROUP-xxxxxxx] [, DOCNAME-xxxxxxx]
              [, FROM-yyymmdd] [, TO-yyymmdd]
// END
```

**Note:** If the SPECIFY statement is used, at least one of the parameters must be included.

## **\$WPLUP OCL Parameters**

**LABEL-filename** Specifies the name assigned to the document directory file of the document library on disk.

## **\$WPLUP Examples**

**Process all members that have the GROUP SALES01 and with a create date in January, 1975. Members are not classified (no key).**

```
// LOAD $WPLUP
// FILE NAME=WPLDDO
// RUN
// MODE ACCEPT
// SPECIFY GROUP=SALES01, FROM=750101, TO=750131
// END
```

**Process all members except those created since March 1, 1975 and exclude those with the GROUP CREDIT1. Members are not classified (no key).**

```
// LOAD $WPLUP
// FILE NAME=WPLDDO
// RUN
// MODE REJECT
// SPECIFY FROM=750301
// SPECIFY GROUP=CREDIT1
// END
```

**Process a master list when the KEY ABCD allows access to all members.**

```
// LOAD $WPLUP
// FILE NAME=WPLDDO
// RUN
// KEY ABCD
// MODE ALL
// END
```



**\$WPLUU--UNLOAD DOCUMENT LIBRARY MEMBERS TO DISKETTE UTILITY PROGRAM**

\$WPLUU unloads selected document library members to diskette in member name sequence based on the control statements entered. For each member on diskette, data from the document directory will be followed by data from each of the document storage records of the member.

Only members with a status of in use (0) are unloaded. The free queue and those members with a status of deleted (F) or being created (1) are not unloaded. The load utility (\$WPLUL) can be used to add the unloaded members to an existing document library.

More than one diskette may be needed to store the selected members. If this condition occurs, the operator is prompted to insert another diskette and continue.

A listing produced from this utility containing the names of the members unloaded, the member storage records, and its create date. Accumulated totals are printed. Figure 35 shows a sample listing.

*Note:* The document library members cannot be unloaded to a format 2 (512-byte sectors) diskette.

```

$WPLUU      DOCUMENT DIRECTORY DATA LIST                      06/20/77

      MEMBER NAME                TOTAL      MEMBER      CREATE
      GROUP   DOCA                STOR RCDS  STATUS      DATE
      GROUP   DOCB                1          0          77/06/20
      GROUP   DOCC                1          0          77/06/20
      GROUP   DOCD                1          0          77/06/20

      MEMBERS STATUS 0 (IN USE)                      =          4

      NUMBER STORAGE RECORDS USED                    =          4
  
```

**Figure 35. Sample \$WPLUU Listing**

\$WPLUU is evoked by the WPSAVE procedure (see index entry: WPSAVE).

## \$WPLUU Utility Control Statement Format

Use	Control Statement
Enter type of selection	*// MODE { ALL ACCEPT REJECT }
If MODE is ACCEPT/REJECT, enter SPECIFY statements	// SPECIFY [GROUP-xxxxxxx] [,DOCNAME-xxxxxxx] [,FROM-yyymmdd] [,TO-yyymmdd]  // END

## \$WPLUU Control Statement Parameters

### MODE Statement

ALL	All members are to be processed by the utility. If the MODE statement is not specified, ALL is the default.
ACCEPT	Only those members which satisfy at least one of the conditions given in SPECIFY statements following the MODE statement are to be processed.
REJECT	Only those members which satisfy none of the conditions given in SPECIFY statements following the MODE statement are to be processed.

### SPECIFY Statement

The specify statement is used to give conditions by which individual members are to be accepted or rejected for processing by the utility. A maximum of 20 SPECIFY statements are allowed. During each execution of the utility, a single scan of the file being processed is made with each member being compared to the specified conditions. A member is accepted (or rejected if MODE REJECT) if it satisfies at least one of the SPECIFY statements. A statement is satisfied only if all of its parts are satisfied.

GROUP-xxxxxxx	Specifies the group name.
DOCNAME-xxxxxxx	Specifies the member name. This parameter is normally used with the GROUP parameter, but need not be.

**Note:** If either the GROUP or the DOCNAME parameter is less than eight characters, blanks are automatically filled on the right. Blanks are not allowed in the middle of the parameter list.

- FROM-yymmdd Specifies that all members with a create date equal to or greater than this parameter are accepted (or rejected if MODE REJECT). The parameter must be in the form yymmdd. The default is 000000.
- TO-yymmdd Specifies that members with a create date equal to or less than this parameter are to be accepted (or rejected if MODE REJECT). The parameter must be in the form yymmdd. The default is 999999.

### \$WPLUU OCL and Control Statement Sequence

To initiate the unload utility through OCL, enter the following:

```
// LOAD $WPLUU
// FILE NAME-WPLDDO [, LABEL-filename-1]
// FILE NAME-WPLDSO [, LABEL-filename-2]
// FILE NAME-WPLBUO, UNIT-I1 [, LABEL-filename-3]
  [, RETAIN-retention-days], PACK-vol-id
// RUN
// MODE      { ALL
              { ACCEPT }
              { REJECT }
// SPECIFY   [GROUP-xxxxxxx] [, DOCNAME-xxxxxxx]
              [, FROM-yymmdd] [, TO-yymmdd]
// END
```

Note: If the SPECIFY statement is used, at least one of the parameters must be included.

### \$WPLUU OCL Parameters

- LABEL-filename-1 Specifies the name assigned to the document directory file of the document library on disk. This must be the *libname* specified when the document library was initialized, appended with DSO.
- LABEL-filename-3 Specifies the name of the output file on diskette.
- RETAIN-retention-days Number of days the document library backup on diskette is to be retained. The default is 1.
- PACK-vol-id The volume identification of the document library backup on diskette.

## **\$WPLUU Examples**

Unload members with GROUP FINANCE and DOCNAME PROJECT from the document library INPUT to output file name on diskette STORIT with retention days of 10, and volume id of KEEPIT. Also, select members with GROUP ACCT and created between January 1, 1976 and June 30, 1976.

```
// LOAD $WPLUU
// FILE NAME=WPLDDO,LABEL=INPUTDDO
// FILE NAME=WPLDSO,LABEL=INPUTDSO
// FILE NAME=WPLBUO,UNIT=I1,LABEL=STORIT,RETAIN=10,
// PACK=KEEPIT
// RUN
// MODE ACCEPT
// SPECIFY GROUP=FINANCE,DOCNAME=PROJECT
// SPECIFY GROUP=ACCT,FROM=760101,TO=760630
// END
```

Unload all members of the document library using the default options.

```
// LOAD $WPLUU
// FILE NAME=WPLDDO
// FILE NAME=WPLDSO
// FILE NAME=WPLBUO,UNIT=I1,PACK=KEEPIT
// RUN
// MODE ALL
// END
```

## **\$WPMCU—MAGNETIC CARD UTILITY PROGRAM**

\$WPMCU allows the user to erase the data on magnetic cards, write a card eject character in character 1 of track 1. Also, the utility allows the user to check the card for usability.

If the user specifies ERASE in the WPMCCARD procedure, the entire card has blanks written on it with a card eject character in character 1 of track 1.

\$WPMCU is evoked by the WPMCCARD procedure (see index entry: WPMCCARD).

## **\$WPMCU OCL and Control Statement Sequence**

```
// SWITCH indicator settings
// LOAD $WPMCU
// RUN
```

## **\$WPMCU OCL Parameters**

### **SWITCH Statement**

<b>0xxxxxx</b>	Specifies that only a card eject character is written in character 1 of track 1.
<b>1xxxxxx</b>	Specifies that blanks are written on the entire card with a card eject character written on character 1 of track 1.

## **\$WPPBI-PRINT BELT IDENTIFICATION UTILITY**

This program sets the print belt ID in the system configuration record and the program level communication area (PLCA). It also reads the requested belt number from disk and writes it into the member **##BELT**. The procedure WPIMID loads **##BELT** member as the print image.

### **\$WPPBI Utility Control Statement Format**

```
// LOAD $WPPBI
// RUN
// BELTID nnn
// END
```

### **\$WPPBI Control Statement Parameter**

<b>nnn</b>	Specifies the print belt as follows:
<b>96</b>	American WP belt supporting keyboards 1, 2, 3, 4, 5, 7, 8, 9, 17, and 18
<b>100</b>	Danish WP belt supporting keyboards 56 and 57
<b>101</b>	French WP belt supporting keyboard 31
<b>102</b>	French WP belt supporting keyboard 32
<b>103</b>	German WP belt supporting keyboards 26 and 27
<b>104</b>	German WP belt supporting keyboard 28
<b>105</b>	German WP belt supporting keyboard 29
<b>106</b>	Italian WP belt supporting keyboards 41 and 46
<b>116</b>	British WP belt supporting keyboards 66 and 67

## **\$WPPBM-PRINTER BELT MAP UTILITY PROGRAM**

\$WPPBM changes the Printer Belt Table for the 48- or 64-character belt so that 96-character set data may be printed. The character set parameter is passed in the UPSI switch.

\$WPPBM allows the user to print the lowercase alphabet a through z as uppercase A through Z on a 64- or 48-character set printer belt. This is done by mapping the lowercase characters that do not exist on the 64/48 character belt into printable uppercase characters in the print belt image table. All other characters that do not exist on the 64/48 belt are printed as a period (.). The characters that co-exist are unaffected. The effect of this utility is temporary and is removed by the next image load.

\$WPPBM is evoked by the WPBELT procedure (see index entry: *WPBELT*). Do not run WPPBM in a word processing environment.

### **\$WPPBM Utility Control Statement Format**

\$WPPBM can be evoked with the following:

```
// SWITCH indicator settings
// LOAD $WPPBM
// RUN
```

### **\$WPPBM Control Statement Parameter**

<b>Indicator Setting</b>	<b>Function</b>
00000010	Maps the American WP print belt characters (belt ID 96)
00000100	Maps the Danish WP print belt characters (belt ID 100)
00000110	Maps the French and Italian 96-character WP print belt characters (belt ID's 101 and 106)
00001000	Maps the French 88-character WP print belt characters (belt ID 102)
00001010	Maps the German (belt ID 103)
00001100	Maps the German 88-character WP print belt characters (belt ID 104)
00001110	Maps the German 96-character WP print belt characters (belt ID 105)
00010000	Maps the British WP print belt characters (belt ID 116)

## **\$WPSET—WORD PROCESSING CONFIGURATIONS UTILITY PROGRAM**

\$WPSET sets the console keyboard ID, the default mag card translate table ID, and the // allowed indicator in the system configuration record by entering a two-digit country code.

\$WPSET is evoked by the WPSET procedure (see index entry: *WPSET*).

### **\$WPSET OCL and Control Statement Sequence**

```
// LOAD $WPSET
// RUN
// CC nn
// END
```

### **\$WPSET OCL PARAMETERS**

nn Specifies the two-digit country code. The code is used to identify the console keyboard ID and the default mag card translate table that is to be placed in the configuration record. Any code other than 01 (U.S.) sets the word processing indicator to allow the ,, format only.

nn	Console ID	Mag Card ID	// Allowed	Country
01	100	002	Yes	United States
02	212	066	No	UK
03	206	026	No	Germany/Austria
04	204	032	No	France (Azerty)
05	207	046	No	Italy
06	202	056	No	Denmark
17	205	032	No	France (Qwerty)

## **\$WPURG—PURGE DOCUMENT DIRECTORY UTILITY PROGRAM**

\$WPURG updates the document directory freed queue field which indicates the number of directory entries remaining.

\$WPURG is evoked by the WPPURGE procedure (see index entry: *WPPURGE*).

*Note:* This utility cannot be invoked by itself. It must be used in conjunction with utilities \$COPY, \$DELET, and \$RENAM and they must be used in the sequence listed in the WPPURGE procedure.

## **\$WPURG OCL and Control Statement Sequence**

To initiate the purge function through OCL, enter the following:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-filename-1
// FILE NAME-COPYO,LABEL-filename-2,RETAIN-P,RECORDS-nn
// RUN
// COPYFILE OUTPUT-DISK,DELETE-'23,XC6',REORG-YES
// END
// LOAD $DELET
// RUN
// REMOVE UNIT-F1,LABEL-filename-1
// END
// LOAD $RENAM
// RUN
// RENAME LABEL-filename-2,NEWLABEL-filename-1
// END
// LOAD $WPURG
// FILE NAME-WPLDDO,LABEL-filename-1
// RUN
```

## **\$WPURG OCL Parameters**

- LABEL-filename-1** Specifies the name assigned to the document directory file of the document library on disk. This must be five or less alphameric characters (the first one must be alphabetic) with DDO added to the end.
- LABEL-filename-2** Specifies the name assigned to the temporary document directory file of the document library on disk. The WPPURGE procedure uses SCRATCH for this name.
- NEWLABEL-filename-1** Specifies the name assigned to the document directory file of the document library on disk. This is the resultant directory after the freed entries have been removed. This must be five or less alphameric characters (the first one must be alphabetic) with DDO added to the end.
- RECORDS-nn** Specifies the size of the newly purged document directory.





## Chapter 4. Communications

This section describes the communications capabilities provided with the IBM System/32 system control program word processing feature. The following information is included:

- The function of the word processing communication utility (WPCU).
- A description of how to prepare for communications and execute the utility.
- The communication specification statements used to define your communication job requirements.
- The format of the procedure command statements used to evoke the utility.
- The OCL and utility control statements for WPCU.

### COMMUNICATION UTILITY FUNCTIONS

The word processing communications utility (WPCU) provides the capability of transmitting and receiving document libraries and data files to/from the System/32 fixed disk. WPCU supports communication between two System/32's with word processing support, or between a System/32 and one of the following office machines:

- IBM Office System 6/430 Information Processor
- IBM Office System 6/440 Information Processor
- IBM Office System 6/450 Information Processor
- IBM 6640 Document Printer
- IBM Mag Card II Typewriter-Communicating

Throughout this chapter, office machine refers to these devices.

WPCU uses the System/32 binary synchronous communications adapter (BSCA) programming support. Refer to the *IBM System/32 Functions Reference Manual*, GA21-0176 for detailed information about BSCA. Before using WPCU, ensure that your system is configured for BSCA communications, that all required equipment is installed and that a communication line is available.

## WPCU OPERATION

Two procedures provided with the WPCU enable transmission/reception of document library members and System/32 files (see WCU and WCUS procedures). You may also write your own procedures or operator control language statements to transmit or receive these files.

Along with these procedures, you will need to provide communication specification statements to define the type of operation (transmit or receive), the remote device characteristics, and other information used by WPCU (see *communication specification statements*). You can put these communication specification statements on the disk so they are accessible each time a communication job is required, or you can enter statements from the keyboard each time a job is transmitted or received. To put your communication specification statements in the source library on the disk, you can:

- Use the source entry utility program of System/32 utilities program product and enter the communication specification statements from the keyboard. See the *IBM System/32 Utilities Program Product Reference Manual—Source Entry Utility*, SC21-7605, for information about the source entry utility.
- Place the communication specification statements on a diskette. (The communication specification statements on the diskette must be in basic exchange format.) The communication specification statements can be transferred to the disk library by copying them to a file using the TRANSFER command and from the file to the source library using the TOLIBR command. See the *IBM System/32 System Control Programming Reference Manual*, GC21-7593, for more information.
- Use the reader-to-library copy function of the utility program \$MAINT and enter the source program from the keyboard. See the *IBM System/32 System Control Programming Reference Manual*, GC21-7593, for more information.

You should ensure the System/32 BSCA is configured properly. If changes are required, use the ALTERBSC or OVERRIDE procedures. Refer to the *IBM System/32 Data Communications Reference Manual*, GC21-7691, for more information about these procedures.

You can configure your system for RPG (respond YES to *DATA COMMUNICATIONS SUPPORT FOR RPG?* prompt during installation) so WPCU automatically determines the line type and mode (transmit or receive). Otherwise, if you are using a switched line, you need to use the OVERRIDE procedure each time you change from transmit mode to receive mode. The SWTYPE parameter in the OVERRIDE procedure must be changed from MC (manual call) to AA (auto answer). Refer to the *Procedures Used for System Configuration and Installation* section of the *IBM System/32 System Control Programming Reference Manual*, GC21-7593.

## **EXECUTING WPCU USING A POINT-TO-POINT NETWORK**

Once the BSCA is configured properly, the steps to be performed depend upon the type of BSCA network: switched or nonswitched. For nonswitched networks, a direct communications line between stations exists. For switched networks, a direct communication line is not always established; the operator must establish the communication line.

When executing WPCU, a schedule should be made between both the transmitting and receiving stations to ensure that the System/32 and remote station are both initialized at the proper time. Data format should be checked to ensure it is compatible. Execution steps must be followed exactly or a disconnect can occur.

### **Nonswitched Network**

1. Prepare the printer, inserting the proper forms.
2. If WPCU is to transmit:
  - a. The receiving station should be made ready to receive data. If the receiving station is a System/32, WPCU must be used to receive the data.
  - b. Load and initialize the WPCU with either the WCU procedure (to transmit document library members) or the WCUS procedure (to transmit System/32 files).
  - c. WPCU begins executing and data is transmitted.
3. If WPCU is to receive:
  - a. A suitable document library must be made available to store the input. See WPINIT procedure.
  - b. Load and initialize the WPCU with either the WCU procedure (to receive document library members) or the WCUS procedure (to receive System/32 files).
  - c. WPCU begins executing and data is received.

## Switched Network

If you are initiating the call:

1. Prepare the printer, inserting the proper forms.
2. Load and initialize the WPCU with either the WCU procedure (for document library members), or the WCUS procedure (for System/32 files).
3. When information message 3295 appears:
  - a. Put your modem in talk mode.
  - b. Pick up the receiver and dial the remote station.
  - c. Either the operator of the remote terminal answers your call or you hear a high-pitched tone indicating that the operator has placed his/her modem in auto-answer mode. If you talk to the operator, request that his/her modem be placed in data mode.
  - d. Enter option 0.
  - e. Place your modem in data mode.
  - f. Hang up the receiver.
  - g. WPCU begins executing and data is transmitted.

If you are receiving the call and talk to the calling operator:

1. Prepare the printer, inserting the proper forms.
2. Prepare the document library using WPINIT.
3. Load and initialize WPCU.
4. When information message 3290 appears:
  - a. After you are called, lift the receiver and talk to the operator of the other system.
  - b. Enter option 0 to continue.
  - c. Put your modem in data mode *before* the caller puts the calling modem in data mode and hang up the receiver.
  - d. WPCU begins executing and data is received.

If you are receiving the call and answer automatically:

1. Prepare the printer, inserting the proper forms.
2. Prepare the document library using WPINIT.
3. Load and initialize WPCU.
4. Put your modem in auto-answer mode.
5. When your phone rings, WPCU begins executing and data is received. (Do not lift the receiver on your modem when the phone rings.)

If you encounter problems while attempting to perform any of the preceding steps, refer to the Problem Determination section of the *IBM System/32 Operator's Guide*, GC21-7591.

## COMMUNICATION SPECIFICATION STATEMENTS

The communication specification statements define the characteristics of the entire transmission. Valid statement identifiers are LINE, REMOTE, MSG, MEMBER, RECORD, and END.

*Note:* WPCU expects the valid statement identifiers to start in position 1 of the statement.

### LINE Statement

The LINE statement defines the type of operation, transparency option, and terminal ID character. You must include one LINE statement and it must be the first communication specification statement in the source library member. The format of the LINE statement is:

$$\text{LINE MODE-} \left\{ \begin{array}{l} \text{XMIT} \\ \text{RCV} \end{array} \right\}, \left[ \text{XPCY-} \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \right] \left[ ,\text{RCVID-c,XMITID-c} \right]$$

where:

- |        |                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MODE   | specifies the type of operation (transmit or receive). Valid MODE operands are XMIT for transmit and RCV for receive. The MODE keyword must be specified.                                                                                                                                                                                                                             |
| XPCY   | specifies the use of the EBCDIC transparency feature. Valid XPCY operands are N for nontransparent operation and Y for transparent operation. This keyword defaults to N, nontransparent operation. If transparency is specified, it must be specified at both the transmitting and receiving systems. Transparent operation is valid only for System/32 to System/32 communications. |
| RCVID  | specifies the terminal identification to be verified on a switched network. The operand specifies one character (c) which is expanded to the two character terminal identification sequence for verification. The RCVID keyword is ignored for nonswitched (leased) networks and is optional for switched (dial) networks. If RCVID is specified, XMITID must also be specified.      |
| XMITID | specifies the terminal identification to be transmitted on a switched network. The operand specifies one character (c) which is expanded to the two character terminal identification sequence for verification. The XMITID keyword is ignored for nonswitched (leased) networks and is optional for switched (dial) networks. If XMITID is specified, RCVID must also be specified.  |

## REMOTE Statement

The REMOTE statement describes the characteristics of the remote terminal. You must include one REMOTE statement and it must follow the LINE statement. The format of the REMOTE statement is:

$$\text{REMOTE TYPE-} \left\{ \begin{array}{l} \text{S32} \\ \text{OM1} \end{array} \right\} \left[ ,\text{ID-}n\text{nn} \right] \left[ ,\text{BSIZE-} \left\{ \begin{array}{l} 128 \\ 256 \\ 512 \end{array} \right\} \right] \left[ ,\text{DEVICE-} \left\{ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} \right\} \right]$$

where:

- TYPE** specifies the type of remote station; S32 for a System/32, OM1 for office machines. The TYPE keyword must be specified.
- ID-*nnn*** specifies the keyboard ID for communications with an office machine. This defines the EBCDIC/WP transmission code and indicates the translation to be performed by WPCU. This parameter is ignored when a System/32 is the remote station. The WPCU supports character set translation for the following keyboards:

ID	Name
001	Office System 6 96-character keyboard
002	American Standard 101-A keyboard <sup>1</sup>
003	American Standard 101-B keyboard <sup>1</sup>
004	American Standard 101-C keyboard <sup>1</sup>
005	American Standard 101-D keyboard <sup>1</sup>
007	Legal 149 keyboard <sup>1</sup>
008	Legal 177 keyboard <sup>1</sup>
009	Pica 046 keyboard
017	Accounting 172 keyboard
018	Accounting 174 keyboard
026	Austrian/German 88-character keyboard
027	Austrian/German 96-character keyboard
028	Austrian/German 88-character keyboard
029	Austrian/German 96-character keyboard
031	French 96-character keyboard
032	French 88-character keyboard
041	Italian 96-character keyboard
046	Italian 88-character keyboard
056	Danish 88-character keyboard
057	Danish 96-character keyboard
066	British 88-character keyboard
067	British 96-character keyboard

<sup>1</sup>The character set for this keyboard is a subset of keyboard 1. You may specify ID-001 for this keyboard.

- BSIZE** specifies the BSC buffer size. Valid operands are 128, 256, and 512. The default is 512 bytes.
- DEVICE** selects the component at the remote office machine that will accept transmitted data. Valid operands are: 1, 2, 3, and 4. The receiving operator should tell you which code to specify because the device code assigned depends on the type of office machine and the particular installation. If **DEVICE** is not specified, the receiving station determines the output device selected. For System/32 to System/32 transmission, this parameter is ignored if coded.

### MSG Statement

The **MSG** statement allows you to transmit operator messages to another System/32 or an office machine. The **MSG** statement is optional and if coded, must follow the **REMOTE** statement. Informational messages received at a System/32 by WPCU (from another System/32 via **MSG** statements or from an office machine via the **INF** command) are listed on the System/32 printer. You may include up to four **MSG** statements. This statement is invalid if **MODE-RCV** is coded on the **LINE** statement. The format of the **MSG** statement is:

```
MSG ' text '
```

where:

text is from 1 to 80 characters enclosed by single quotes (').

### MEMBER Statement

The **MEMBER** statement is used to specify the document library member name. The **MEMBER** statement is used only for document library member transmission or reception. The **MEMBER** statements follow the **MSG** statement if the **MSG** statement is coded; otherwise the **MEMBER** statement(s) follows the **REMOTE** statement.

If **MODE-RCV** is coded in the **LINE** statement, code only one **MEMBER** statement. If you receive more than one document library member, the two-digit number appended to the document library member name is incremented by one after each member is received.

If you coded **MODE-XMIT** on the **LINE** statement, you can code from one to eight **MEMBER** statements. The order of the **MEMBER** statements determines the order in which the document library members are transmitted to the receiving station. You can transmit multiple members as separate documents or chain them together as a single document. The format of the **MEMBER** statement is:

```
MEMBER [groupname.] docname[( nn)] [,KEY-cccc] [ ,CHAIN- { Y } { N } ]
```



where:

- groupname** specifies the document library member qualifier (optional).
- docname** specifies the name of the document library member. For receive operations, this parameter is limited to a maximum of six characters. WPCU appends a two-character sequence number to this name as each member is stored. For transmit operations, the name can be up to eight characters and must be the name of an existing document library member.
- (nn)** specifies the document security class (0-32) associated with a document to be stored (receive operations only). If omitted, class 0 is assumed.
- cccc** specifies the user security key for document library access.
- Y**  
**N** specifies whether the document library member specified in the next MEMBER statement is to be concatenated to this document library member as one contiguous document. The default is N, which means treat the following member as a separate document. If Y is selected, the member specified on the following MEMBER statement is to be concatenated on transmission with this member. The CHAIN keyword is valid only for transmit operations (MODE-XMIT on LINE specification statement).

#### **RECORD Statement**

The RECORD statement is optional. It specifies the record size for creation of sequential files. If the RECORD statement is not coded, the record size of the sequential System/32 file created is 128 bytes. The RECORD and MEMBER statements are mutually exclusive. The RECORD statement follows the REMOTE statement and is valid only if MODE-RCV is coded on the LINE statement. The format of the RECORD statement is:

RECORD SIZE-*nnn*

where:

- SIZE-*nnn*** specifies the record size for creation of the sequential System/32 file. The size specified must be less than or equal to the BSIZE keyword of the REMOTE statement.

## END Statement

The END statement must be the last statement. The END statement has no parameters.

## WCU PROCEDURE

To transmit up to eight or receive up to 99 document library members, use the WCU command statement.

### WCU Command Statement Format

WCU [libname  
WPL] , [specname  
KEYBOARD] [ ,Y  
 ,N]

### WCU Command Statement Parameters

libname	specifies the name of the document library selected.
WPL	It can be up to five uppercase alphameric characters, the first of which must be alphabetic (A-Z, #, \$, or @). If <i>libname</i> is not specified, the default name assigned to the document library is WPL.
specname	specifies the name of the source library member on disk containing the communications specification statements. If <i>specname</i> is not specified, you are prompted to enter the specification statements from the System/32 keyboard.
KEYBOARD	
Y	Y specifies the operator may change the specification statements contained in the source library member named in the <i>specname</i> parameter (the source library member itself is not changed). This parameter applies only if a source library member name is specified. If Y is specified, each of the specification statements stored in the source library member is displayed one at a time and the operator may change or delete them as necessary. If Y is not specified, no prompting is done.
N	

To change a specification statement after Y has been specified, make corrections to the line and press the RECORD ADVANCE key. If you do not wish to correct or change the displayed statement, press the RECORD ADVANCE key. To delete the statement, press the ENTER key.

## Examples of Document Library Member Transmission/Reception

### System/32 to System/32

Two document library members, LETTER and ENCLOSE, are to be transmitted from one System/32 to another System/32. In the sending System/32, the two members are in document library ABC and have a groupname XYZ. The source library member SPECS contains the communication specification statements for sending this job:

In the source library of the sending System/32, the member SPECS contains:

```
LINE MODE-XMIT
REMOTE TYPE-S32
MSG 'LETTER AND ENCLOSURE FOR SALES
CAMPAIGN'
MEMBER XYZ.LETTER,KEY-ABCD
MEMBER XYZ.ENCLOSE,KEY-ABCD
END
```

To send the two document library members, the operator enters:

```
WCU ABC,SPECS
```

In the receiving System/32, the source library member RCVCOMM contains the communication specification statements for receiving this job:

```
LINE MODE-RCV
REMOTE TYPE-S32
MEMBER A123.LETTER(10),KEY-MJCC
END
```

The receiving System/32 operator enters:

```
WCU ,RCVCOMM
```

The two documents received are stored in the WPL library (default library name) with the names A123.LETTER01 and A123.LETTER02. The document security class is 10.

**Note:** A library must have been prepared using WPINIT.

## Office Machine to System/32

A series of documents are to be received from an office machine and written to the document library SALES, using the communication specification statements from the System/32 source library member named SYS6. In addition, the statements in SYS6 are to be changed. For this job, the receiving System/32 operator enters:

```
WCU SALES,SYS6,Y
```

The communication specification statements in the System/32 source member named SYS6 are:

```
LINE MODE-RCV  
REMOTE TYPE-OM1, ID-1  
MEMBER SALES.JUNE( 30 ),KEY-ABCD  
END
```

After the communications line is established, the documents are received from the office machine and stored in the document library named SALES. The first document received is named SALES.JUNE01, the second document SALES.JUNE02, etc., until end of transmission is received.

## WCUS PROCEDURE

To transmit or receive a System/32 file, use the WCUS command statement. Two formats of the WCUS command statement are shown. The first is used for transmit; the second is used for receive.

### WCUS Command Statement Formats

WCUS filename, [specname  
KEYBOARD] , [ Y ]  
[ N ]

or

WCUS filename, [specname  
KEYBOARD] , [ Y ] , { RECORDS,value-1 }  
[ N ] , { BLOCKS,value-2 }

## WCUS Command Statement Parameters

filename	specifies the disk file to be transmitted or received. This parameter must be specified and can be from one to eight uppercase alphameric characters. If this is a transmit operation, the file must already exist on disk. If this is a receive operation, specify a unique name to be given to the new file created for the received records and specify the number of records or blocks to be allocated.
specname	specifies the name of the source library member on disk containing the communications specification statements.
KEYBOARD	If specname is not specified, the operator is prompted to enter the specification statements from the System/32 keyboard.
Y	Y specifies the operator may change the specification statement contained in the source library member named in the specname parameter (the source library member itself is not changed). This parameter applies only if a source library member name is specified. If Y is specified, each of the specification statements stored in the source library member is displayed one at a time and the user may change or delete them as necessary. If Y is not specified, no prompting is done.
N	
	To change a specification statement after Y has been specified, make corrections to the line and press the RECORD ADVANCE key. If you do not wish to correct or change the displayed statement, press the RECORD ADVANCE key. To delete the statement, press the ENTER key.
RECORDS	requests that the disk file be made large enough to contain the number of records indicated by value-1.
value-1	specifies the number of records that the disk file is to accommodate.
BLOCKS	requests that the disk file be made large enough to contain the number of blocks indicated by value-2.
value-2	specifies the number of blocks that the disk file is to accommodate.

## Example of System/32 Sequential File Transmission

### System/32 to System/32

A System/32 data file is to be transmitted to another System/32. The data file, named DATAFILE, contains 100 96-byte records with both alphameric and packed fields, requiring the transparency feature to be used. The sending operator enters:

```
WCUS DATAFILE
```

The operator enters the communications specification statements through the keyboard:

```
LINE MODE-XMIT,XPCY-Y  
REMOTE TYPE-S32,BSIZE-128  
END
```

The receiving operator enters:

```
WCUS MYFILE,,,RECORDS,100
```

This specifies the receiving System/32 is to create a file named MYFILE and the file is to contain 100 records. The receiving operator is then prompted to enter the communication specification statements. The operator enters:

```
LINE MODE-RCV,XPCY-Y  
REMOTE TYPE-S32  
RECORD SIZE-96  
END
```

## \$WCUM OCL and Utility Control Statement Sequence

If you want to code your own OCL and utility control statements instead of using the IBM-supplied procedures, you must enter LOAD, FILE, RUN, SOURCE, and END statements.

```
// LOAD $WCUM
// FILE NAME-libnameDDO,UNIT-F1[,LABEL-filename-1]
// FILE NAME-libnameDSO,UNIT-F1[,LABEL-filename-2]
// or
// FILE NAME-filename-3,UNIT-F1[,LABEL-filename-4]
//      [,RETAIN-retention      ] {,RECORDS,value-1}
//                                     {,BLOCKS,value-2}
// RUN
[ // SOURCE NAME-specname[,PROMPT- $\begin{cases} Y \\ N \end{cases}$ ]]
// END
```

For document library files two FILE statements are required, one describing the document library directory file and one describing the document library storage file. For sequential disk files, only one FILE statement is required to describe the file to be transmitted or created.

**Note:** The SOURCE statement is optional but if you omit the SOURCE statement, you must include the communication specification statements after the RUN statement.

## \$WCUM OCL Parameters

- |                  |                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME-libnameDDO  | Specifies the name assigned to the document directory file of the document library on disk. This must be the <i>libname</i> specified when the document library was initialized, appended with DDO. |
| NAME-libnameDSO  | Specifies the name assigned to the document storage file of the document library. This must be the <i>libname</i> specified when the document library was initialized, appended with DSO.           |
| LABEL-filename-1 | Specifies the name to be assigned to the document directory that is being transmitted.                                                                                                              |
| LABEL-filename-2 | Specifies the name to be assigned to the document storage file that is being transmitted.                                                                                                           |
| LABEL-filename-3 | Specifies the name of the System/32 sequential file on disk that is to be transmitted.                                                                                                              |
| LABEL-filename-4 | Specifies the name that is to be assigned to the System/32 sequential file being transmitted.                                                                                                       |

RETAIN-P T S	Specifies the retention designation of the disk output file being created: T (temporary), P (permanent), or S (scratch).
RECORDS,value-1	Specifies the number of records to allocate for the System/32 sequential file. This parameter is specified only when receiving documents.
BLOCKS,value-2	Specifies the number of blocks to allocate for the System/32 sequential file. This parameter is specified only when receiving documents.
NAME-specname	Specifies the name of the member within the system source member library which contains the communication specification statements.
PROMPT- Y N	Specifies that operator is prompted with each of the communication specification statements in the requested source member. Changes made to the specification statements are not permanent.

### **\$WCUM Examples**

#### *Transmit a Document Library Member*

To transmit a document library member from the library named MYLIB, you could use the following OCL and utility control statements. In this example, the SOURCE statement specifies the communications specification statements in the source library member named DOCLIBT are to be used for a transmit operation:

```
// LOAD $WCUM
// FILE NAME=WPLDDO,UNIT=F1,LABEL-MYLIBDDO
// FILE NAME=WPLDSO,UNIT=F1,LABEL-MYLIBDSO
// RUN
// SOURCE NAME=DOCLIBT,PROMPT=Y
// END
```



### ***Transmit and Receive System/32 Sequential Files***

In the following examples, the file being transmitted and received is named FILE1 and the source library members containing the communication specification statements are named SEQFILET (for the transmit operation) and SEQFILER (for the receive operation).

To transmit a System/32 sequential file named FILE1, use the following OCL and utility control statements:

```
// LOAD $WCUM  
// FILE NAME=WPLSEQ,UNIT-F1,LABEL-FILE1  
// RUN  
// SOURCE NAME=SEQFILET,PROMPT-N  
// END
```

To receive a System/32 sequential file named FILE1, you would use the following OCL and utility control statements (100 records are to be allocated):

```
// LOAD $WCUM  
// FILE NAME=WPLSEQ,UNIT-F1,LABEL-FILE1,RETAIN-P,RECORDS,  
// SOURCE NAME=SEQFILER,PROMPT-Y  
// END
```

## CONSIDERATIONS FOR COMMUNICATING WITH A SYSTEM/32

If you are transmitting to another System/32, you can send documents or data files in any format that is acceptable to programs in the receiving system. The receiving system must have word processing support (SCP feature 6002) and the word processing communications utility installed.

Documents transmitted to another System/32 may be unformatted, and may contain formatting instructions and OCL that can be interpreted by the receiving system. Documents may also contain *format lines* (used to define automatic tab settings, right margin setting, line spacing, and adjust mode on Mag Card II typewriters or other office machines that have the format feature). WPCU does not interpret embedded commands or translate data during communications with another System/32. If you want to send final copy to a System/32, ensure that the document(s) to be transmitted are formatted in the document library before evoking WPCU.

You can also transmit or receive System/32 sequential, indexed, or direct data files using WPCU (see the WCUS command statement). A data file is always transmitted or received consecutively, by record, regardless of the actual file organization. Only one data file can be transmitted or received in a communications session. Ensure that the operator at the other System/32 also uses the WCUS command statement to transmit or receive a data file.

When transmitting System/32 data files, use transparent mode (XPCY-Y) if the data file contains any binary or packed data fields. Ensure that the receiving System/32 specified the transparency option also.

## **CONSIDERATIONS FOR COMMUNICATING WITH AN OFFICE MACHINE**

If you are transmitting to an office machine (IBM Office System 6), ensure that documents are in a format acceptable to the receiving system or device. In general, text that can be played out on a mag card typewriter is compatible with an office machine.

If a document contains any formatting instructions, they must be removed or executed to produce final copy prior to using WPCU. Do not send documents containing formatting instructions or System/32 OCL to an office machine.

If a document is to be revised or formatted at the receiving office machine, you will probably want to send unformatted text (with no embedded instructions). Margin text (headings, footings, page numbers) can be added when you are ready to produce final copy.

IBM Office System 6 operator control language and format lines may be included in documents, if final copy is to be produced at the office machine. You may want to receive standard sets of Office System 6 instructions and maintain them in the document library for retransmission with various text documents.

Documents transmitted or received from an office machine are translated to ensure character set compatibility between the two systems. The keyboard ID parameter on the REMOTE specification statement defines the appropriate translation. If you are transmitting to an office machine, specify the number of the mag card keyboard used to create the document(s), and inform the receiving operator of the keyboard ID. If you are receiving a document(s) from an office machine, ask the other operator for the appropriate keyboard ID.

## Appendix A. IBM System/32 Characters

The following characters are from the 96-character, 64-character, and 48-character print belts. The print belts are as follows:

- 41 Austrian/German 48-character belt supporting keyboard 206
- 42 Danish 48-character belt supporting keyboard 202
- 47 British 48-character belt supporting keyboard 212
- 48 American/French/Italian 48-character belt supporting keyboards 100, 204, 205, and 207
- 49 48HN FORTRAN 48-character belt
- 64 American 64-character belt supporting keyboard 100
- 65 French/Italian 64-character belt supporting keyboards 204, 205, and 207
- 66 Austrian/German 64-character belt supporting keyboard 206
- 67 Danish 64-character belt supporting keyboard 202
- 72 British 64-character belt supporting keyboard 212
- 96 American 96-character WP belt supporting keyboards 1, 2, 3, 4, 5, 7, 8, 9, 17, and 18.
- 100 Danish 96-character WP belt supporting keyboards 56 and 57
- 101 French 96-character WP belt supporting keyboard 31.
- 102 French 96-character WP belt supporting keyboard 32.
- 103 Austrian/German 96-character WP belt supporting keyboards 26 and 27.
- 104 Austrian/German 96-character WP belt supporting keyboard 28
- 105 Austrian/German 96-character WP belt supporting keyboard 29
- 106 Italian 96-character WP belt supporting keyboards 41 and 46.
- 116 British 96-character WP belt supporting keyboards 66 and 67.

Character	Hexadecimal Equivalent	Printer Belt ID																		
		41	42	47	48	49	64	65	66	67	72	96	100	101	102	103	104	105	106	116
Blank	40																			
\$ (dollar)	4A			.						.										.
¢ (cent)	4A							.				.								
[ (left bracket)	4A							.												
# (number)	4A								.	.						.	.			
◌́ (acute accent)	4A											.								
◌° (degree)	4A												.	.					.	
. (period)	4B	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
< (less than)	4C						.	.	.	.	.	.	.	.	.	.	.	.	.	.
± (plus / minus)	4C						.	.	.	.	.	.	.	.	.	.	.	.	.	.
◌̂ (circumflex u)	4C												.							
μ (mu)	4C																.			
( (left paren)	4D						.	.	.	.	.	.	.	.	.	.	.	.	.	.
+ (plus)	4E	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
(OR)	4F						.	.	.	.	.	.	.	.	.	.	.	.	.	.
! (exclamation)	4F											.	.	.	.	.	.	.	.	.
[ (left bracket)	4F											.					.			.
& (ampersand)	50	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
§ (section)	5A												.	.						
! (exclamation)	5A						.			.	.									.
] (rt bracket)	5A						.										.	.		
\$ (dollar)	5A							.							.		.	.		
▣ (lozenge)	5A								.											
£ (pound)	5A											.								
é (acute e)	5A																		.	
Ü (umlaut U)	5B	.						.							.		.	.		
Å (angstrom)	5B	.	.							.			.							
£ (pound)	5B			.						.										.
\$ (dollar)	5B				.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
ê (circumflex e)	5B													.						
* (asterisk)	5C	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
î (circumflex i)	5C													.						
) (right paren)	5D						.	.	.	.	.	.	.	.	.	.	.	.	.	.
; (semicolon)	5E						.	.	.	.	.	.	.	.	.	.	.	.	.	.
¬ (not)	5F						.	.	.	.	.	.	.	.	.	.	.	.	.	.
] (rt bracket)	5F											.								
ˆ (circumflex)	5F											.	.	.				.		
◌́ (acute accent)	5F														.		.	.		
◌° (degree)	5F																			.
- (minus)	60	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Character	Hexadecimal Equivalent	Printer Belt ID																		
		41	42	47	48	49	64	65	66	67	72	96	100	101	102	103	104	105	106	116
/ (slash)	61	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
§ (section)	6A								.			.								
ø (slash oh)	6A											.								
û (grave u)	6A												.	.						
ö (umlaut o)	6A														.		.	.		
ò (grave o)	6A																		.	
± (plus / minus)	6A																			.
, (comma)	6B	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
% (percent)	6C	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
_ (underscore)	6D						.	.	.	.	.	.	.	.	.	.	.	.	.	.
> (greater than)	6E						.	.	.	.	.		.	.			.	.	.	.
° (degree)	6E											.						.		
ô (circumflex o)	6E														.					
? (question)	6F						.	.	.	.	.	.	.	.	.	.	.	.	.	.
® (registered)	75											.								
£ (pound)	76											.								
' (prime)	77																	.		
¼	77																			.
£ (pound)	78																.			
¾	78																			.
˘ (grave accent)	79						.	.	.	.	.	.	.		.		.	.		
¶ (paragraph)	79											.								
ë (umlaut e)	79														.					
# (number)	79												.							
û (grave u)	79																		.	
] (rt bracket)	79																			.
: (colon)	7A						.	.	.	.	.	.	.	.	.	.	.	.	.	.
# (number)	7B				.		.	.		.		.					.			.
£ (pound)	7B												.						.	
Ä (umlaut A)	7B	.							.						.		.	.		
Æ (ligature AE)	7B		.							.		.								
â (circumflex a)	7B														.					
@ (at)	7C			.	.		.	.		.		.								.
Ø (slash O)	7C		.							.		.								
Ö (umlaut O)	7C	.							.						.		.	.		
â (grave a)	7C													.	.					
§ (section)	7C																		.	
' (single quote)	7D	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
ˆ (circumflex)	7D															.				
= (equal)	7E					.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
“ (double quote)	7F						.	.	.	.	.	.	.	.	.	.	.	.	.	.

Character	Hexadecimal Equivalent	Printer Belt ID																		
		41	42	47	48	49	64	65	66	67	72	96	100	101	102	103	104	105	106	116
a	81											.	.	.	.	.	.	.	.	.
b	82											.	.	.	.	.	.	.	.	.
c	83											.	.	.	.	.	.	.	.	.
d	84											.	.	.	.	.	.	.	.	.
e	85											.	.	.	.	.	.	.	.	.
f	86											.	.	.	.	.	.	.	.	.
g	87											.	.	.	.	.	.	.	.	.
h	88											.	.	.	.	.	.	.	.	.
i	89											.	.	.	.	.	.	.	.	.
j	91											.	.	.	.	.	.	.	.	.
k	92											.	.	.	.	.	.	.	.	.
l	93											.	.	.	.	.	.	.	.	.
m	94											.	.	.	.	.	.	.	.	.
n	95											.	.	.	.	.	.	.	.	.
o	96											.	.	.	.	.	.	.	.	.
p	97											.	.	.	.	.	.	.	.	.
q	98											.	.	.	.	.	.	.	.	.
r	99											.	.	.	.	.	.	.	.	.
<sup>1</sup> / <sub>4</sub>	A1											.								
¨ (umlaut)	A1												.	.	.					
ß (sharp s)	A1															.	.			
ï (grave i)	A1																	.	.	
μ (mu)	A1																		.	.
s	A2											.	.	.	.	.	.	.	.	.
t	A3											.	.	.	.	.	.	.	.	.
u	A4											.	.	.	.	.	.	.	.	.
v	A5											.	.	.	.	.	.	.	.	.
w	A6											.	.	.	.	.	.	.	.	.
x	A7											.	.	.	.	.	.	.	.	.
y	A8											.	.	.	.	.	.	.	.	.
z	A9											.	.	.	.	.	.	.	.	.
<sup>2</sup> (superscript)	C0											.								.
æ (ligature ae)	C0												.							
é (acute e)	C0													.	.					
ä (umlaut a)	C0															.	.			
à (grave a)	C0																	.	.	
A	C1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
B	C2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
C	C3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
D	C4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
E	C5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
F	C6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
G	C7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
H	C8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
I	C9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Character	Hexadecimal Equivalent	Printer Belt ID																		
		41	42	47	48	49	64	65	66	67	72	96	100	101	102	103	104	105	106	116
<sup>3</sup> (superscript)	D0											.								.
å (angstrom)	D0												.							
è (grave e)	D0													.	.					.
ü (umlaut u)	D0															.	.			.
J	D1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
K	D2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
L	D3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
M	D4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
N	D5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
O	D6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
P	D7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Q	D8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
R	D9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
<sup>1/2</sup>	DA											.							.	.
<sup>3</sup> (superscript)	DA												.		.		.	.		.
ï (umlaut i)	DA													.						
\	E0								.		.	.								
§ (section)	E0														.		.	.		
<sup>1/2</sup>	E0											.								.
\$ (dollar)	E0											.								.
ç (cedilla c)	E0												.	.						.
S	E2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
T	E3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
U	E4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
V	E5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
W	E6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
X	E7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Y	E8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
Z	E9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
§ (section)	EA												.							
<sup>2</sup> (superscript)	EA												.		.		.	.		.
ü (umlaut u)	EA													.						.
¨ (umlaut)	EA																		.	.
0	F0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	F1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	F2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	F3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
4	F4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	F5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
6	F6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	F7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
8	F8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
9	F9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.





**command statement:** A statement that is used to request the performance of a particular function. It always contains the command name and may include parameters.

**disk file:** A collection of records (data) stored on a disk.

**document directory:** The document library component that contains information about each member in the document library.

**document library:** An area on disk that contains word processing members.

**document storage:** The document library component that contains data portion of the text.

**format 1:** A record containing system information that describes a file. There is one format 1 record for each file on the disk.

**job control (JJT):** That portion of the word processing control programming which initiates, processes, and terminates jobs.

**linked direct access method (LDAM):** That portion of the word processing control programming which interfaces with the standard System/32 access methods.

**member:** A named collection of records, text, or statements in the document library.

**parameter:** A variable that is given a constant value for a specific purpose or process.

**procedure:** A named collection of related OCL statements, and possibly, utility control statements, that describe a specific function or group of functions. A procedure is evoked by a command statement or included OCL statements.

**system utility programs:** A set of programs provided with the system that are used to perform the everyday routine tasks required by a data processing system.

**sysin device:** The device that is specified by the WPSTART command. May be keyboard, fixed disk, mag card unit, or document library. All subsequent task commands must be input via the specified device.

**work processing:** The processing of business correspondence by use of automated methods.

**word processing job stream:** The input from word processing sysin device. The job stream can contain word processing procedures and their data.



- \$MAINT utility program 64
- \$WCUM
  - examples 77
  - OCL and utility control statement sequence 76
  - OCL parameters 76
- \$WPKBR
  - control statement parameters 22
  - OCL utility control statement format 22
  - utility program (keyboard redefine utility) 22
- \$WPKEY
  - OCL and control statement sequence 36
  - utility program (security file update option) 35
- \$WPLUF
  - control statement parameters 39
  - examples 41
  - OCL and control statement sequence 40
  - OCL parameters 41
  - utility control statement formats 39
  - utility program (delete document library member) 38
- \$WPLUI
  - examples 43
  - OCL and control statement sequence 42
  - OCL parameters 42
  - utility program (library initialization) 42
- \$WPLUL
  - examples 46
  - OCL and control statement sequence 45
  - OCL parameters 46
  - utility control statement formats 44
  - utility control statement parameters 44
  - utility program (load members from diskette) 43
- \$WPLUM
  - control statement examples 49
  - control statement format 48
  - sysin format 47
  - utility program (maintenance utility) 47
- \$WPLUP
  - control statement parameters 51
  - document directory print utility program 50
  - examples 53
  - OCL and control statement sequence 52
  - OCL parameters 53
  - utility control statement formats 51
- \$WPLUU
  - control statement parameters 56
  - examples 57
  - OCL parameters 56
  - OCL and control statement sequence 56
  - utility control statement formats 55
  - utility program (unload members to diskette) 54
- \$WPMCU
  - OCL and control statement sequence 57
  - OCL parameters 58
  - utility program (magnetic card utility) 57
- \$WPPBI
  - control statement parameters 58
  - utility control statement format 58
  - utility program (print belt identification) 58
- \$WPPBM
  - control statement parameters 59
  - utility control statement format 59
  - utility program (printer belt map) 59
- \$WPSET
  - OCL and control statement sequence 60
  - OCL parameters 60
  - utility program (word processing configuration) 60
- \$WPURG
  - OCL and control statement sequence 61
  - OCL parameters 61
  - utility program (purge document directory) 60
- ##BELT 58
- abbreviations and acronyms vii
- access key 35
- activation of word processing 1
- add a key 35
- ALTERBSC procedure 64
- American keyboards 23
- Appendix A: IBM System/32 characters 81
- Austrian/German keyboards 28
- British keyboards 34
- BSCA 64, 65

change a key 35  
characters, System/32 81  
command statement format  
    KEYBOARD 4  
    WCU 71  
    WCUS 73  
    WPBELT 5  
    WPDELETE 6  
    WPFREE 6  
    WPIMID 7  
    WPINIT 8  
    WPKEY 9  
    WPLMAINT 10  
    WPMCCARD 11  
    WPPRINT 11  
    WPPURGE 13  
    WPRESTOR 14  
    WPSAVE 15  
    WPSET 17  
    WPSTART 18  
    WPSTOP 19  
command statement parameters, WCU 71  
command statement parameters, WCUS 74  
communications 63  
    specification statement 67  
        END statement 71  
        LINE statement 67  
        MEMBER statement 69  
        MSG statement 69  
        RECORD statement 70  
        REMOTE statement 68  
    utility functions 63  
considerations for communications, with a  
    System/32 79  
considerations for communications, with an  
    office machine 80  
control statement formats  
    \$WPLUF utility 39  
    \$WPLUM utility 48  
control statement parameters  
    \$WPKBD 22  
    \$WPLUF 39  
    \$WPLUP 51  
    \$WPLUU 55  
    \$WPPBI 58  
    \$WPPBM 59

Danish keyboards 33  
delete  
    a key 35  
    a member (\$WPLUM) 47  
    the security file 35  
        document library member utility program, \$WPLUF 38  
document directory print utility program, \$WPLUP 50  
document library initialization utility  
    program, \$WPLUI 42

END communications specification statement 71  
examples  
    \$WCUM utility program 77  
    \$WPLUF utility program 41  
    \$WPLUI utility program 44  
    \$WPLUL utility program 46  
    \$WPLUM utility program 49  
    \$WPLUP utility program 53  
    \$WPLUU utility program 57  
    document library member transmission/reception 72  
    System/32 sequential file transmission 75  
    WCU command statement 72  
    WPDELETE command statement 06  
    WPFREE command statement 7  
    WPINIT command statement 9  
    WPPRINT command statement 12  
    WPRESTOR command statement 15  
    WPSAVE command statement 16

French keyboards 30

glossary 87

how to use this manual ix

IBM Mag card II Typewriter-Communicating 63  
IBM System 6/430 Information Processor 63  
IBM System 6/440 Information Processor 63  
IBM System 6/450 Information Processor 63  
IBM 6640 Document Printer 63

introduction 1  
  activation 1  
  setup 1  
  termination 2  
inquiry support 2  
Italian keyboards 32

KBID statement parameters 22

key, security 35

keyboards

  American 23  
  British 34  
  Danish 33  
  French 30  
  Austrian/German 28  
  Italian 32

KEYBOARD

  command statement format 4  
  parameters 4  
  procedure 4

keyboard redefine utility program, \$WPKBD 22

LDAM maintenance utility (see \$WPLUM)

LINE statement 67

list the file (security) 35

load document library members from diskette utility  
  program, \$WPLUL 43

load members from diskette (see \$WPLUL)

loading and running word processing programs 1

magnetic card utility program, \$WPMCU 57

maintenance utility program, \$WPLUM 47

MEMBER statement 69

MODE statement in \$WPLUF utility 39

MSG statement 69

network nonswitched 65

network switched 66

OCL and control statement sequence

\$WPKEY 36

\$WPLUF 40

\$WPLUI 42

\$WPLUL 45

\$WPLUP 52

\$WPLUU 56

\$WPMCU 57

\$WPSET 60

\$WPURG 61

\$WPKBD 22

\$WCUM 76

OCL parameters

\$WPLUF 41

\$WPLUI 42

\$WPLUL 46

\$WPLUP 53

\$WCUM 76

\$WPLUU 56

\$WPMCU 58

\$WPSET 60

\$WPURG 61

OVERRIDE procedure 64

parameters

  KEYBOARDS 4

  OCL (see LABEL, LOAD, and RUN statements)

  WPBELT 5

  WPDELETE 6

  WPFREE 7

  WPIMID 8

  WPINIT 9

  WPLMAINT 10

  WPMCCARD 11

  WPPRINT 12

  WPPURGE 13

  WPRESTOR 14

  WPSAVE 16

  WPSET 17

  WPSTART 18

point-to-point networks 65

print belt identification utility, \$WPPBI 58

print belt map (see \$WPPBM)

printer belt map utility program, \$WPPBM 59  
print utility program, \$WPLUP 50  
procedures 3  
    KEYBOARD 4  
    WCU 71  
    WCUS 73  
    WPBELT 5  
    WPDELETE 6  
    WPFREE 6  
    WPIMID 7  
    WPINIT 8  
    WPKEY 9  
    WPMMAINT 10  
    WPMCCARD 10  
    WPPRINT 11  
    WPPURGE 13  
    WPRESTOR 13  
    WPSAVE 15  
    WPSET 17  
    WPSTART 17  
    WPSTOP 18  
prompts, security file 37  
purge document directory utility program, \$WPURG 60

RECORD statement 70  
rekey (security classification) 47  
REMOTE statement 68  
rename a document 47  
respecify security classes 35  
RPG support 64

security file prompts 37  
security file update utility program, \$WPKEY 35  
setup of word processing 1  
source entry utility 64  
SPECIFY statement in \$WPLUR utility 40  
sysin format, \$WPLUM 47

termination of word processing 2  
TOLIBR command 64  
tracking (security file) 35  
TRANSFER command 64  
transparency feature (XPCY) 67

unload document library members to diskette utility  
program, \$WPLUU 54  
unload members to diskette (\$WPLUU)  
utilities 21  
utility control statements format  
    \$WPLUL 44  
    \$WPLUP 51  
    \$WPLUU 55  
    \$WPPBM 59

WCU  
    command statement format 71  
    command statement parameters 71  
    procedure 71

WCUS  
    command statement formats 73  
    command statement parameters 74  
    procedure 73

word processing communications (see WPCU)  
word processing configuration utility  
program, \$WPSET 60

WPBELT  
    command statement format 05  
    parameters 5  
    procedure 5

WPCU  
    operation 64  
    program (communications utility) 63  
    program (communications utility), communications  
    specification statement 67  
    program (communications utility), description 64  
    program (communications utility), operation 64  
    program (communications utility), point-to-point  
    nonswitched 65  
    program (communications utility), point-to-point  
    switched 66

WPDELETE  
    command statement format 6  
    examples 6  
    parameters 6  
    procedure 6

WPFREE  
    command statement format 06  
    examples 7  
    parameters 7  
    procedure 6

WPIMID  
  command statement format 7  
  parameters 8  
  procedure 7  
WPINIT  
  command statement format 8  
  example 9  
  parameters 9  
  procedure 8  
WPKEY  
  command statement format 9  
  procedure 9  
WPL (default libname)  
  WPFREE procedure 7  
  WPINIT procedure 9  
  WPPRINT procedure 11  
  WPPURGE procedure 13  
  WPRESTOR procedure 14  
  WPSAVE procedure 16  
  WPSTART procedure 18  
WPLMAINT  
  command statement format 10  
  parameters 10  
  procedure 10  
WPMCCARD  
  command statement format 11  
  parameters 11  
  procedure 10  
WPPRINT  
  command statement format 11  
  examples 12  
  parameters 12  
  procedure 11

WPPURGE  
  command statement format 13  
  parameters 13  
  procedure 13  
WPRESTOR  
  command statement formats 14  
  examples 15  
  parameters 14  
  procedure 13  
WPSAVE  
  command statement format 15  
  examples 16  
  parameters 16  
  procedure 15  
WPSET  
  command statement formats 17  
  parameters 17  
  procedure 17  
WPSTART  
  command statement format 18  
  parameters 18  
  procedure 17  
WPSTOP  
  command statement format 19  
  parameters 19  
  procedure 18

XPCY (transparency feature) 67







**International Business Machines Corporation**

**General Systems Division  
4111 Northside Parkway N.W.  
P.O. Box 2150  
Atlanta, Georgia 30301  
(U.S.A. only)**

**General Business Group/International  
44 South Broadway  
White Plains, New York 10601  
U.S.A.  
(International)**

IBM System/32 System Control Programming Reference Manual Word Processing (File No. S32-36) Printed in U.S.A. GC34-0078-3