

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK
GARCHING BEI MÜNCHEN

R E S Y

A R e p o r t E d i t o r S y s t e m

F.R. Hertweck, U.J. Schneider

IPP R-28

August 1978

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

R E S Y
A R E P O R T E D I T O R S Y S T E M

F. R. Hertweck
U. J. Schneider

Program description, command description, and
use of RESY.

0. Introduction	1
1. General Philosophy of RESY	2
1.1 Structure of Text	2
1.2 Standard Format of Report	2
1.3 Character Set	4
1.4 Right-Justification and Hyphenation	5
2. RESY Commands	6
2.1 Syntactic Description	6
2.2 Parameter Specification Commands	7
2.3 Report Title Commands	8
2.4 Paragraph Descriptor Commands	10
2.5 Item List Commands	11
2.6 Display Commands	12
2.7 Layout Controlling Commands	13
3. How to operate RESY	13
3.1 Using RESY with a Batch Job	13
3.2 Printing a Report with the DIABLO Terminal	14
3.3 Using a DIABLO 1620 as RESY Terminal	15
Appendix 1: Summary of RESY Commands	16
Appendix 2: Error Indicators	17
Appendix 3: Sample Text	18

Copyright © 1978 by
Max-Planck-Institut fuer Plasmaphysik
8046 Garching, GERMANY
Alle Rechte,
auch die des photomechanischen Nachdrucks, vorbehalten
All Rights Reserved

0. Introduction

This note describes the program RESY, a system to generate reports in a standard format, from input stored in an AMOS file segment. The edited report, together with references to input lines, is printed on an IBM line printer (with TN chain) and at the same time it may be stored in another AMOS file segment as 133 byte print lines, ready for printing in final form (i.e. without line numbers, comments, etc., cf 3.2).

Though in many ways resembling ROFF and similar systems, the main emphasis has been laid on simplicity of the input, at the expense of generality. It is supposed that input is being prepared on a typewriter like terminal, for instance a DIABLO 1620 with 96 ASCII character set.

As its main characteristic the system may be described as "paragraph-oriented", i.e. all editing necessary for a paragraph in most cases can be described by one single command. Paragraphs may start with heading of chapters, sections, or subsections, and within those there may be "titled" and "normal" paragraphs. Chapters, sections, and subsections are numbered automatically and optionally all output lines may be right-justified by inserting additional blanks between words.

There are a few commands that permit the generation of (as an option, automatically) numbered lists of items, either with integers or with the letters of the alphabet. In addition, lists of items may be "numbered" by words (called "verbal lists"). For these lists, additional indentation is automatically generated. While those lists are considered to be substructures of paragraphs, they cannot themselves contain paragraphs.

A set of other commands permits the generation of "centered" output (i.e. centered on a line), copying input as it is to the output file(s), the printing of a headline and/or page numbers on the top of each page, printing in one or two columns per page, adjustment of page size, line length, etc., generation of a sequence of free lines or free pages (for later insertion of pictures and graphs), etc.

As an option the user may restrict the number of blanks to be inserted in a line during justification. If a conflict arises during justification (i.e. the next word to be put into the line is too long, and the number of blanks to be distributed is too large), a hyphenation routine is called in interactive mode of operation (yet to be implemented) which permits the insertion of hyphens, represented by the symbol tilde (cf. 1.3) During generation of output a tilde is removed if it appears within a line, and converted to a hyphen if it appears at the end of a line.

1. General Philosophy of RESY

In this chapter we describe the general philosophy of RESY, the structure of its input text, the standard format of output, the character set used, etc.

1.1 Structure of Text

The input to RESY consists of commands and text. The text is the information the user wants to print, the commands are used for structuring it.

Commands

Commands always start with a \$ sign in column one of an input line. They usually have some parameters to specify the action of the command in more detail, and, in other cases, they may also contain text (for details, cf. chapter 2).

Text

Though the commands structure the report mainly in paragraphs, the text which builds the paragraphs is a sequence of words, separated by blanks (or a line end). In other words, the first non-blank character of a line defines the beginning of a new word, which ends with the last non-blank character following it (always scanning from left to right); the next word starts with the first non-blank character following the (sequence of) blank(s) and ends with the last non-blank character in this sequence or with the last character of the line.

Pseudo-blanks

When right-justification of a text is performed (before it is output), the blanks that otherwise would remain at the end of a line are distributed over the line by increasing the normal inter-word gap of one blank to several. In some cases this is not desirable (for instance where a special character, not existent in the ASCII-96 character set, is later to be introduced by hand); then the user may use the "pseudo-blank" circumflex (cf.1.3) which will be printed as a blank by RESY (but may be used as a normal character during AMOS editing).

1.2 Standard Format of Report

RESY produces a report in a standardized format. This implies that there are certain restrictions as to the format of the report. Main emphasis has been laid on easiness of description rather than generality of the layout. For instance, generation of chapter, section or subsection headlines including preceding or following blank lines is controlled by one command for each type of heading. As an example, the command

\$. Chapter Heading

produces three blank lines, a line of the form

3. Chapter Heading

(assuming it is chapter 3) and two more blank lines; the digit 3 is left justified at the left margin.

The whole of this paper has been prepared with RESY, and most of its features have been used. Appendix 3 shows the input to produce the report.

The format of the report can be described best by the concept of paragraphs. For instance, taking the example of the present section: It starts with a headline followed by a paragraph. Then there is a special line displaying the format of the chapter heading command, continued with the text of the first paragraph, then the display of the chapter heading as produced by RESY, followed with the remainder of the paragraph. Then the second paragraph, this paragraph, starts. Apart from the display of lines mentioned, we have been talking about two paragraphs: a section header paragraph, and a "normal" paragraph. In all, RESY distinguishes between five types of paragraphs:

- (1) chapter header paragraphs (with headlines numbered 0., 1., 2., etc., or optionally starting with n)
- (2) section header paragraphs (with headlines numbered 1.1, 1.2, 1.3, etc., in each chapter)
- (3) subsection header paragraphs (with headlines numbered 2.3.1, 2.3.2, 2.3.3, etc. in each section)
- (4) titled paragraphs (with an unnumbered headline)
- (5) (normal) paragraphs (no headline)

If nothing else has been specified, all paragraphs start with the first line indented by a certain number of spaces.

In addition to paragraphs the user may produce "lists of items", like the one in the preceding paragraphs. There are several possibilities to produce those lists:

- . indicate numbering explicitly by typing the numbers between parentheses
- . use automatic lettering (starting with a, b, c,...)
- . use any of the symbols . - or o to "number" the items (as done in the present list)
- . use words to produce a "verbal list"

In all of these cases, the numbering will stick out in front of a line, and all following lines, describing the item listed, will be automatically indented by 5 places (except for verbal lists, where indentation is controlled by the length of the word in the "short" form of the command), as

shown in the two lists produced so far. The verbal list in "long" form produces a left-justified headline and the following text is indented by 5 places.

Another standard formatting concept is the "displaying" of (input) lines. This facility caters for those cases where a sequence of lines has to be reproduced in the report irrespective of the formatting constraints of RESY. A sequence of input lines, preceded by a DISPLAY command and followed by an (arbitrary) escape character is copied exactly as it is, with the possibility of deleting the first n characters and taking the following m characters of each input line.

In order to facilitate the construction of the report title page, a CENTER command is available and has the form

```
$C Text of the Headline of Report
|<-----text part----->|
```

RESY takes the text part of this command (defined as the string of characters starting with the first non-blank character following the C and ending with the last non-blank character of the line) and prints it centered on a line.

1.3 Character Set

The RESY character set is built around the 7-bit ASCII 96 character set (shown in the arrangement of a DIABLO 1620 keyboard):

```
! " # $ % & ' ( ) _ = ~ |
1 2 3 4 5 6 7 8 9 0 - ^ \

Q W E R T Y U I O P {
q w e r t y u i o p @ [

A S D F G H J K L + * }
a s d f g h j k l ; : ]

Z X C V B N M < > ?
z x c v b n m , . /
```

Because this character set needs 7 bits for its representation the eighth (leftmost) bit of each byte is used to augment the character set with a second set of underscored characters. This removes the need for any "underline" commands. The user at the typewriter types in the text, performs as many backspaces as desired and presses the underscore key to add the underscores. The DIABLO support routines will then construct the appropriate character encodings (yet to be implemented).

The line as stored in AMOS is a sequence of bytes where each byte corresponds to one DIABLO printing position (in "update mode", i.e. when the line is printed for update purposes). Obviously, a RESY output line cannot coincide with an input line, except for display lines.

There are characters that have a special meaning:

(a) the tilde

is used as the hyphen symbol for word hyphenation at the end of lines; within an output line it is suppressed, at the end of an output line it is printed as -

(b) the circumflex

is treated as a non-blank character as far as determination of string lengths and automatic blank insertion during right justification is concerned, but it is printed as blank in an output line.

Therefore, these two characters cannot be used as normal text characters (except a tilde in a title line and both in display lines).

1.4 Right-Justification and Hyphenation

In order to produce a neatly printed report text, RESY permits right-justification of an output text line by insertion of additional blanks between words (normally, words are separated by one blank).

Justification is under the control of the user. A special command sets a justification control parameter J to either -1 (no justification) or a value >0, indicating how many additional blanks may be inserted in a line.

For smaller positive values of J a conflict may arise: the number LFREE of remaining free positions in an output line, may be greater than J, but LW + 1 (where LW is the length of the next word) may be too large:

$$J < \text{LFREE} < \text{LW} + 1$$

Then in interactive mode of operation of RESY the user at the terminal will be asked for a "hyphenation breakpoint": The word which should be hyphenated will be printed and by positioning the printing device of the DIABLO by a suitable number of backspaces and then entering a termination command, the user may indicate where the hyphen should be inserted (cf. section 3.3).

When this option is selected, a new version of the input segment is produced, with the hyphens (encoded as tilde) inserted into the lines of the AMOS segment which can later be used as RESY input again. To the hyphens inserted either in this way or through normal AMOS File Editor updating apply the rules described in 1.3.

2. RESY Commands

In this chapter we present a detailed description of the RESY commands.

In the first section the syntactic description of the commands is explained. In the following sections, commands are described according to the following classification:

- o parameter specification commands
- o report title commands
- o paragraph descriptor commands
- o item list commands
- o display commands
- o layout control commands

2.1 Syntactic Description

In this section the general rules for the syntactic description of commands are outlined.

A command starts with a \$ in position 1 of the input line, immediately followed by the command verb, if any. As usual, syntactic entities will be described by words enclosed in <...>. Square brackets, [...] will be used to indicate optional parameters. If a parameter can be omitted, there is a note with each command indicating the default settings.

In addition, the following general rules hold:

- (a) Capital letters and special symbols stand for themselves
- (b) n, m, and k denote non-negative integers
- (c) <string> is a sequence of characters on one input line, starting and ending with a non-blank character (i.e. the string starts with the first non-blank character to the right of the previous syntactic entity encountered, and ends with the last non-blank character of the line)
- (d) <text> is a sequence of words (the words not containing any blanks), each pair of words separated by one or more blanks or the end of the input line; wherever <text> is used, the parameter is optional, i.e. it may be omitted.
- (e) In a command line syntactic entities and parameters are separated by blanks.

Any further rules will be described when used.

2.2 Parameter Specification Commands

The commands of this group are used to specify the parameters for RESY operation. They describe the page layout, (one or two columns of text per page, page length, maximum line length, current line length, headline text, page numbering, line numbering, paragraph indentation, justification control, etc.) and whether a table of contents should be constructed.

\$CO n [m n] COLUMNS

The maximum page width is set and it is specified, whether one or two columns are to be printed on each page.

If there is only one integer n, single column layout is specified: n is the width of that column (i.e. the maximum possible line length). For single column layout, n must be in the range $20 \leq n \leq 120$.

If there are three integers, double column layout is specified: n is the width for each of the columns and m is the number of blanks between the two columns. For double column layout, m must be in the range $1 \leq m \leq 20$. As there are not more than 120 print positions for one line, n must be chosen that $20 \leq n \leq 59$ and $41 \leq 2*n+m \leq 120$ is valid.

Initially single column layout with column width 60 is specified.

The "COLUMNS" command performs a major reset:

- the headline is cleared to blank
- page numbering is switched off
- if there is an unfinished page, it is ejected and a new page is initialized
- the line length is set to n
- in case of double columns layout, line numbering is suppressed

\$PL n PAGE LENGTH

Specifies the page length by n, which must be in the range $20 \leq n \leq 100$. Initial (default) value is n=60.

\$HD [<string>] HEADER DEFINITION

Sets a new value for the headline. Maximum length of the string is 6 less the maximum line width. If the string is omitted, the headline text is reset to blank. If a headline text and/or page numbering have been specified, a headline with the string (left adjusted) and/or the page number (right adjusted) is printed as the topmost line of each page, followed by two blank lines (single column) or four blank lines (double column). These three or five lines are not included in the page length.

\$PN [n] PAGE NUMBERING
 Resets the current value of page number. If parameter n is omitted, no page numbering is requested. If page numbering has been requested and/or a headline text has been defined, a headline is printed as explained above for \$HD.

\$LL n LINE LENGTH
 Sets the current text line length LL to value n which must be in the range $20 \leq n \leq LLMAX$. The new setting becomes valid after the current output line being assembled in the buffer has been printed and a new line is started. Initial (default) value is $LL=60$ ($LLMAX=72$).

\$LN [n] LINE NUMBERING
 Specifies whether or not the latest input line number (columns 73-80 of the input line) is printed on the left-hand margin of the output line. If $n=0$, no line numbering is requested; $n>0$ or n omitted requests line numbering. Line numbering is not effective, if double column layout is specified.

\$PI n PARAGRAPH INDENTATION
 Sets the parameter INDENTP for paragraph indentation; n must be in the range $0 \leq n \leq 10$. A new value is effective when a new paragraph starts. Default value is 3.

\$JC [n] JUSTIFICATION CONTROL
 Sets the parameter J for line justification control to n. If n is omitted, J is set to -1 which means: no justification. If $n>0$, it specifies the maximum number of blanks that may be inserted into a line during justification. If the length LW of the next word to be put into the output line buffer is such that

$$J < LFREE < LW,$$

then hyphenation (cf. 1.4) will be requested. If $J > LLMAX$, hyphenation obviously never will be requested. Default value is $J=100$.

\$CN n CHAPTER NUMBER
 Set initial value for first chapter to be printed. Default value is 0.

2.3 Report Title Commands

This group of commands is used to create a title page of the report and, optionally, a table of contents. Though these commands should appear in the beginning of the input of a report to RESY, printing of the title page and the table of contents is delayed until the report has been completed.

The report title page has a format as shown on the title page of this report. The top line of the title page may optionally contain the report number and the current date (of the day of printing) or a specified date. There are

three lines reserved for the title of the report (i.e. unspecified lines are printed as blank lines) and up to 10 lines for the authors. An additional maximum of 10 lines may be used to print a summary of the report. Finally, the automatic production of a table of contents (consisting of chapter, section, subsection, and appendix headings, and containing up to 60 lines) may be requested. It is printed either on the title page (if there is still room left) or on a new page. The title page has the format of 60 lines with 60 columns each.

\$RN <string> REPORT NUMBER
Defines the report number, consisting of a string of characters of maximum length 40. Will be printed left-justified on top line of title page.

\$RD [<string>] REPORT DATE
Defines report date as string of maximum length of 12 characters. If input string is blank, the date is obtained from the system and printed in the form

Feb 12, 1978

\$RT <string> REPORT TITLE
Defines the next line of the title. The string may have a maximum length of 60 characters. Up to three title lines may be defined which will be printed centered and separated by one blank line.

\$RA [<string>] REPORT AUTHOR
Defines the name (and affiliation) of the next author. The string may have a maximum length of 60 characters. It is printed centered. Up to 10 author lines may be defined.

\$RS <text> REPORT SUMMARY
Defines the summary of the report. It is printed following the author list as an unindented paragraph of up to 10 lines, 50 columns wide and indented 5 places. The text of the summary may start on the command line and may be supplemented by more text input lines. The text of the summary is terminated by any other command. The layout of the summary does not disturb the current definitions of line length and indentation.

\$RC REPORT TABLE OF CONTENTS
Generates a table of contents consisting of chapter, section, and subsection headings. The latter are indented 3 and 6 places, respectively. Blank lines are inserted where appropriate. The table of contents is printed on lines 60 columns wide with the page numbers right-justified. If appendices have been specified, the headings of those are also printed in the table of contents.

2.4 Paragraph Descriptor Commands

In this section the paragraph heading commands are described. They are used to generate paragraphs with and without headings, at the same time concluding any previous paragraph. As a rule, a paragraph ends with a (possibly not completely filled) line, which is left non-justified, and a blank line.

If a paragraph with a heading is generated, care is taken by RESY that the heading does not appear without at least part of the paragraph at the bottom of a page. Otherwise an early skip to a new page is performed.

The structure of the input for a paragraph is always the same: a paragraph heading command is followed by a sequence of text lines up to (but not including) the next command line.

`$. <string>` CHAPTER HEADING

The last line of the previous paragraph is printed. If there are less than 10 lines left on the current page, start new page. Print 3 blank lines, increase the chapter number by 1, reset the section and subsection counts to 0, and print (left-justified) the chapter number and the `<string>` as chapter heading, followed by 2 more blank lines. Any `<text>` input lines that follow are printed as the body of a paragraph, the first line of which starts indented INDENTP spaces.

`$.. <string>` SECTION HEADING

The last line of the previous paragraph is printed. If there are less than 8 lines left on the current page, start a new page. Print 2 blank lines, increase the section count by 1, reset the subsection count to 0, and print (left-justified) the section number and `<string>` as section heading, followed by a blank line. Any `<text>` input lines that follow are printed as the body of a paragraph, the first line of which starts indented INDENTP spaces.

`$... <string>` SUBSECTION HEADING

The last line of the previous paragraph is printed. If there are less than 6 lines left on the current page, start a new page. Print a blank line, increase the subsection count by 1, and print (left-justified) the subsection number and `<string>` as subsection heading, followed by a blank line. Any `<text>` input lines that follow are printed as the body of a paragraph, the first line of which is indented INDENTP spaces.

`$T [<string>]` TITLED PARAGRAPH

The last line of the previous paragraph is printed, followed by a blank line. If there are less than 4 lines left on the current page, start a new page. Print the `<string>` left-justified as the title of a paragraph, followed by a blank line. Any `<text>` input lines that follow are printed as the body of a paragraph, the first line of which is indented INDENTP spaces.

\$ [n] <text> PARAGRAPH
 The last line of the previous paragraph is printed, followed by a blank line. Then a new paragraph is started, taking <text> and any following <text> lines as the text of the paragraph. The first line of the new paragraph is indented n spaces (n must be in the range 0<=n<LL). If n is omitted, indentation is INDENTP spaces. Because n may be omitted, <text> must not start with an integer. (If a paragraph with standard indenting does start with an integer, either omit n and <text> and put the text on the next line, or set n=INDENTP and use <text> in any form.)

\$AP APPENDIX
 The last line of the previous paragraph is printed. Chapter numbering is reset to zero and appendix processing is initiated.

\$. [<string>] APPENDIX HEADING
 This command produces an appendix heading if the \$AP command has been issued. The heading is printed in the form:

Appendix 1: [<string>]

etc. The commands \$.. and \$... are not permitted in appendices.

2.5 Item List Commands

The five commands of this group are used to create "item lists" to enumerate items. Two of the commands may be used to generate automatically numbered lists. An "item list" normally is ended when the next command is encountered; exceptions are the break command \$B, and the display commands \$D and \$C. The latter use the indentation position as left-hand margin.

\$(x) <text> ITEM, EXPLICITLY NUMBERED
 x is a string of one or two characters, not containing a right parenthesis. Print last line of previous paragraph and a blank line. Print (x) or (xx) followed by <text> and indent all following text lines (up to next command) 5 additional spaces. Treat the item text like a paragraph (i.e. conclude with an extra blank line when next command is encountered).

\$(z) <text> ITEM, MARKED BY SPECIAL SYMBOL
 z is either of the symbols . or - or o enclosed in parentheses. Print last line of previous paragraph followed by a blank line. Print z followed by text lines (up to next command) printed indented by 5 additional spaces. Treat item text like paragraph (i.e. conclude with an extra blank line when next command is encountered).

\$V <string> VERBAL LIST (SHORT)
 Create list where items are "numbered" by words. Print last line of previous paragraph, followed by a blank

line. Print <string> followed by any text from ensuing <text> input lines. Indent all following text lines by k+1 spaces, where k is the length of <string>, up to the next command. Conclude with an extra blank line when next command is encountered.

\$W <string> VERBAL LIST (LONG)
 Create a list where items are defined by a title line consisting of <string>. Print any text lines indented 5 places. <text> input lines are treated like a paragraph. Conclude with an extra blank line when next command is encountered.

\$N [n] <text> LIST NUMERICALLY
 cf. the description of \$A following in the next paragraph.

\$A [n] <text> LIST ALPHABETICALLY
 Create a list of items where items are numbered automatically, either by numbers increasing by 1 (in the range 0,1,...,20) or by the consecutive letters a, b, c,... of the alphabet. Print last line of previous paragraph, followed by a blank line. If n is present, reset item counter to indicated value n if a numeric list is desired; in case of an alphabetic list n must be 1 for initialization. If n is omitted, increase the list parameter by 1. Print (x) or (yy) where x is a digit or a letter and yy are numbers in the range between 10 and 20. Indent all following text lines additional 5 spaces; treat item <text> and following text lines like a paragraph. Conclude with an extra blank line when next command is encountered.

2.6 Display Commands

There are two types of display commands. They serve to copy information from input lines to output lines, one by one. They should be used whenever information is to be displayed in fixed format. Both commands cause a previous line to be terminated.

\$D [n [m]] [x] DISPLAY
 Copy information from the input lines that follow into the output lines using the current indentation as left margin; the parameters, if present, have the following meaning:

- n - delete first n characters from each input line; if omitted default is n=0
- m - take next m characters from each input line; if omitted default is m=remainder of line
- x - is the escape character that ends the display sequence, if found as first character of an input line; x may not be a digit unless n and m are specified, the default is x="\$".

\$C [**<string>**] **CENTER**
Any previous line is terminated and printed. Then the **<string>** is printed centered on a new line using the current indentation as left margin.

2.7 Layout Controlling Commands

The commands of this group are used to control the layout of the output e.g. left margin control, and empty lines or pages.

\$I [**n**] **INDENTATION**
The indentation constant in use is reset to **n**. All following lines - starting with the next output line - are printed with the left margin indented **n** positions with respect to the leftmost printing position. With **n** not defined or **n=0** the leftmost print position on the line is addressed.

\$B [**n**] **BREAK and INSERT BLANK LINES**
The last line is ended without right justification; then **n** empty lines are inserted. If **n** is omitted, it is assumed to be zero.

\$L [**n**] **INSERT BLANK LINES**
After the next line has been printed, **n** empty lines are inserted. The default value for **n** is 1.

\$FP [**n**] **INSERT FREE PAGES**
When a page has been completed, **n** free pages are inserted. The default value for **n** is 1.

\$P **TERMINATE PAGE**
After the current line has been completed, start a new page. If requested before, the headline and page number are printed.

3. How to operate RESY

At the IPP Computer Installation RESY may be run either as a job under OS/MVT without terminal control or interactively with a DIABLO 1620 Terminal connected to an INTERDATA.

It is assumed that the user is familiar with the AMOS system, as RESY input usually will be prepared with the AMOS File Editor and job submission also will be carried out via AMOS.

3.1 Using RESY with a Batch Job

RESY may run as a batch job under OS MVT on the IBM 360/91 (MVT/91) or under OS MVT on the AMDAHL 470 V/6 (MVT/470). The necessary JCL statements to execute the RESY load module are stored in the AMOS segment SYS:JOBS.RESY.

As output device for the RESY job either a line printer with upper and lower case character printing or - in addition - an AMOS segment may be used. The only high speed line printer with upper and lower case characters available at IPP computer center is usually attached to the IBM 360/91. Therefore, a RESY job should be submitted to MVT/91 (AMOS command S91 before job submission).

There are two ways to let the RESY job handle input and output:

- (1) the text input is read from the OS SYSIN data set; the edited report is printed onto the line printer.

To MVT/91 XSUBMIT a segment containing:

```
$$ SYS:JOBS.RESY
```

followed by the text of the report, or a SUBSTITUTE statement referencing the segment which contains the report text.

- (2) the text input is not submitted with the control statements, but is read directly from an AMOS segment while the RESY job is running. The edited report is printed on the line printer and also written into an AMOS segment (cf. 3.2).

XSUBMIT a segment containing:

```
$$ SYS:JOBS.RESY
// EXEC RESY,SOURCE='<segment specification>',
// REPORT='<segment specification>'
```

where

```
<segment specification> ::= <userid>:<filename>.<segname>
```

If the parameter REPORT is omitted, no AMOS segment is written.

In order to obtain reproducible output for offset printing etc., the DIABLO printer of Remote Station 8 can be used. Remote Station 8 is located at building D2/1st floor and is attached to MVT/470.

The RESY job should be submitted to MVT/470 with the job setup preceded by two additional control commands:

```
/*ROUTE PRINT REMOTE8
/*CONTROL FORM=1
```

3.2 Printing a Report with the DIABLO Terminal

The segment generated by RESY can only be printed with a special job using a DIABLO terminal connected to VM/370. As an AMOS user, submit this job to MVT/470 with the following AMOS commands:

```
?S470
?SUBMIT SYS:JOBS.DIAPRINT
PROBLEM?
TIME?
REGION?
FURTHER PARAMETERS?CLASS=L
```

When the job has reached the go-step, press the "BREAK" key on a VM/370 DIABLO terminal and type in the command:

```
DIAL MVT 3D
```

The segment name and a few print control parameters have to be specified at the terminal. Before the last question (start printing?) is answered, the DIABLO should be positioned to print directly above the next horizontal paper perforation. After all printing has been completed, empty input will end the print job and switch the terminal back to VM/370.

3.3 Using a DIABLO 1620 as RESY Terminal

Implementation and description are forthcoming.

Acknowledgements

The authors wish to thank Mrs. Inge Precht for useful discussions. We also appreciate the work of Mrs. Marianne Glandien, who typed this report in "RESY".

Appendix 1: Summary of RESY Commands

Parameter Specification Commands

\$CO n [m n]	Column Setting
\$PL n	Page Length
\$HD [<string>]	Header Definition
\$PN [n]	Page Numbering
\$LL n	Line Length
\$LN [n]	Line Numbering
\$PI n	Paragraph Indentation
\$JC [n]	Justification Control
\$CN n	Chapter Numbering

Report Title Page and Table of Contents

\$RN <string>	Report Number
\$RD [<string>]	Report Date
\$RT <string>	Report Title
\$RA [<string>]	Report Author
\$RS <text>	Report Summary
\$RC	Generate Table of Contents

Paragraph Descriptor Commands

\$. <string>	Chapter Heading
\$.. <string>	Section Heading
\$... <string>	Subsection Heading
\$T <string>	Titled Paragraph
\$ [n] <text>	Paragraph, n Indentation
\$AP	Appendix
\$. [<string>]	Appendix Heading

Item List Commands

\$(n) <text>	List Item, Explicit Number
\$(.) <text>	List Item, Symbol (. - o)
\$V <string>	List Item, Verbal (short)
\$W <string>	List Item, Verbal (long)
\$N [n] <text>	List Item, Numerical
\$A [n] <text>	List Item, Alphabetical

Display Commands

\$D [n [m]] [x]	Display Begin
x	Display End, default x="\$"
\$C [<string>]	Center

Layout Controlling Commands

\$I [n]	Indentation
\$B [n]	Break, Insert n Blank Lines
\$L [n]	Insert n Blank Lines
\$P	Terminate Page
\$FP [n]	Insert n Free Pages

Appendix 2: Error Indicators

In case RESY detects an error in an input line, this line is printed with an error indicator in the gap between line number and the first position of the line.

Note that error indicators are not stored in a RESY generated AMOS segment.

List of error indicators and their meaning:

(TR) line truncated;

(--) line discarded i.e. too many lines in case of Report Summary;

(**) parameter error in command line;

(??) command unknown.

Appendix 3: Sample Text

The text on the following pages is the input to RESY of this report. A comparison of this input with the output as produced by RESY should help the user to understand how RESY commands are used and what their effect is.

\$LN 0
 \$C
 \$L 52
 \$D

Copyright c 1978 by
 Max-Planck-Institut fuer Plasmaphysik
 8046 Garching, GERMANY
 Alle Rechte,
 auch die des photomechanischen Nachdrucks, vorbehalten
 All Rights Reserved

\$
 \$P
 \$PN 1
 \$LN 0
 \$RN IPP R/28
 \$HD R E S Y - A Report Editor System
 \$RT R E S Y
 \$RT A R E P O R T E D I T O R S Y S T E M
 \$RA F. R. Hertweck
 \$RA U. J. Schneider
 \$RS Program description, command description, and use of RESY.
 \$RD August 1978
 \$RC

\$. Introduction
 This note describes the program RESY, a system to generate reports in a standard format, from input stored in an AMOS file segment. The edited report, together with references to input lines, is printed on an IBM line printer (with TN chain) and at the same time it may be stored in another AMOS file segment as 133 byte print lines, ready for printing in final form (i.e. without line numbers, comments, etc., cf 3.2).

\$.
 Though in many ways resembling ROFF and similar systems, the main emphasis has been laid on simplicity of the input, at the expense of generality. It is supposed that input is being prepared on a typewriter like terminal, for instance a DIABLO 1620 with 96 ASCII character set.

\$.
 As its main characteristic the system may be described as "paragraph-oriented", i.e. all editing necessary for a paragraph in most cases can be described by one single command. Paragraphs may start with heading of chapters, sections, or subsections, and within those there may be "titled" and "normal" paragraphs. Chapters, sections, and subsections are numbered automatically and optionally all output lines may be right-justified by inserting additional blanks between words.

\$.
 There are a few commands that permit the generation of (as an option, automatically) numbered lists of items, either with integers or with the letters of the alphabet. In addition, lists of items may be "numbered" by words (called "verbal lists"). For these lists, additional indentation is automatically generated. While those lists are considered to be substructures of paragraphs, they cannot themselves contain paragraphs.

\$.
 A set of other commands permits the generation of "centered" output (i.e. centered on a line), copying input as it is to the output file(s), the printing of a headline and/or page numbers on the top of each page, printing in one or two columns per page, adjustment

of page size, line length, etc., generation of a sequence of free lines or free pages (for later insertion of pictures and graphs), etc.

\$

As an option the user may restrict the number of blanks to be inserted in a line during justification. If a conflict arises during justification (i.e. the next word to be put into the line is too long, and the number of blanks to be distributed is too large), a hyphenation routine is called in interactive mode of operation (yet to be implemented) which permits the insertion of hyphens, represented by the symbol tilde (cf. 1.3) During generation of output a tilde is removed if it appears within a line, and converted to a hyphen if it appears at the end of a line.

\$. General Philosophy of RESY

In this chapter we describe the general philosophy of RESY, the structure of its input text, the standard format of output, the character set used, etc.

\$.. Structure of Text

The input to RESY consists of commands and text. The text is the information the user wants to print, the commands are used for structuring it.

\$T Commands

Commands always start with a \$ sign in column one of an input line. They usually have some parameters to specify the action of the command in more detail, and, in other cases, they may also contain text (for details, cf. chapter 2).

\$T Text

Though the commands structure the report mainly in paragraphs, the text which builds the paragraphs is a sequence of words, separated by blanks (or a line end). In other words, the first non-blank character of a line defines the beginning of a new word, which ends with the last non-blank character following it (always scanning from left to right); the next word starts with the first non-blank character following the (sequence of) blank(s) and ends with the last non-blank character in this sequence or with the last character of the line.

\$T Pseudo-blanks

When right-justification of a text is performed (before it is output), the blanks that otherwise would remain at the end of a line are distributed over the line by increasing the normal inter-word gap of one blank to several. In some cases this is not desirable (for instance where a special character, not existent in the ASCII-96 character set, is later to be introduced by hand); then the user may use the "pseudo-blank" circumflex (cf.1.3) which will be printed as a blank by RESY (but may be used as a normal character during AMOS editing).

\$.. Standard Format of Report

RESY produces a report in a standardized format. This implies that there are certain restrictions as to the format of the report. Main emphasis has been laid on easiness of description rather than generality of the layout. For instance, generation of chapter, section or subsection headlines including preceding or following blank lines is controlled by one command for each type of heading. As an example, the command

\$C

\$C \$. Chapter Heading

\$C

produces three blank lines, a line of the form

\$C

\$C 3. Chapter Heading

\$C

(assuming it is chapter 3) and two more blank lines; the digit 3 is left justified at the left margin.

\$ The whole of this paper has been prepared with RESY, and most of its features have been used. Appendix 3 shows the input to produce the report.

\$ The format of the report can be described best by the concept of paragraphs. For instance, taking the example of the present section: It starts with a headline followed by a paragraph. Then there is a special line displaying the format of the chapter heading command, continued with the text of the first paragraph, then the display of the chapter heading as produced by RESY, followed with the remainder of the paragraph. Then the second paragraph, this paragraph, starts. Apart from the display of lines mentioned, we have been talking about two paragraphs: a section header paragraph, and a "normal" paragraph. In all, RESY distinguishes between five types of paragraphs:

\$N 1 chapter header paragraphs (with headlines numbered 0., 1., 2., etc., or optionally starting with n)

\$N section header paragraphs (with headlines numbered 1.1, 1.2, 1.3, etc., in each chapter)

\$N subsection header paragraphs (with headlines numbered 2.3.1, 2.3.2, 2.3.3, etc. in each section)

\$N titled paragraphs (with an unnumbered headline)

\$N (normal) paragraphs (no headline)

\$ 0 If nothing else has been specified, all paragraphs start with the first line indented by a certain number of spaces.

\$ In addition to paragraphs the user may produce "lists of items", like the one in the preceding paragraphs. There are several possibilities to produce those lists:

\$(.) indicate numbering explicitly by typing the numbers between parentheses

\$(.) use automatic lettering (starting with a, b, c,...)

\$(.) use any of the symbols . - or o to "number" the items (as done in the present list)

\$(.) use words to produce a "verbal list"

\$ 0

In all of these cases, the numbering will stick out in front of a line, and all following lines, describing the item listed, will be automatically indented by 5 places (except for verbal lists, where indentation is controlled by the length of the word in the "short" form of the command), as shown in the two lists produced so far. The verbal list in "long" form produces a left-justified headline and the following text is indented by 5 places.

\$ Another standard formatting concept is the "displaying" of (input) lines. This facility caters for those cases where a sequence of lines has to be reproduced in the report irrespective of the formatting constraints of RESY. A sequence of input lines, preceded by a DISPLAY command and followed by an (arbitrary) escape character is copied exactly as it is, with the possibility of deleting the first n characters and taking the following m characters of each input line.

\$ In order to facilitate the construction of the report title page, a CENTER command is available and has the form

\$D

```
$C Text of the Headline of Report
|<-----text part----->|
```

\$ 0

RESY takes the text part of this command (defined as the string of characters starting with the first non-blank character

following the C and ending with the last non-blank character of the line) and prints it centered on a line.

\$.. Character Set

The RESY character set is built around the 7-bit ASCII 96 character set (shown in the arrangement of a DIABLO 1620 keyboard):

\$D

```
! " # $ % & ' ( ) _ = ~ |
1 2 3 4 5 6 7 8 9 0 - ^ \
```

```
Q W E R T Y U I O P {
q w e r t y u i o p @ [
```

```
A S D F G H J K L + * }
a s d f g h j k l ; : ]
```

```
Z X C V B N M < > ?
z x c v b n m , . /
```

\$ 0

Because this character set needs 7 bits for its representation the eighth (leftmost) bit of each byte is used to augment the character set with a second set of underscored characters. This removes the need for any "underline" commands. The user at the typewriter types in the text, performs as many backspaces as desired and presses the underscore key to add the underscores.

The DIABLO support routines will then construct the appropriate character encodings (yet to be implemented).

\$ The line as stored in AMOS is a sequence of bytes where each byte corresponds to one DIABLO printing position (in "update mode", i.e. when the line is printed for update purposes).

Obviously, a RESY output line cannot coincide with an input line, except for display lines.

\$ There are characters that have a special meaning:

\$A 1 the tilde

\$B is used as the hyphen symbol for word hyphenation at the end of lines; within an output line it is suppressed, at the end of an output line it is printed as -

\$A the circumflex

\$B is treated as a non-blank character as far as determination of string lengths and automatic blank insertion during right justification is concerned, but it is printed as blank in an output line.

\$ 0

Therefore, these two characters cannot be used as normal text characters (except a tilde in a title line and both in display lines).

\$.. Right-Justification and Hyphenation

In order to produce a neatly printed report text, RESY permits right-justification of an output text line by insertion of additional blanks between words (normally, words are separated by one blank).

\$ Justification is under the control of the user. A special command sets a justification control parameter J to either -1 (no justification) or a value >0, indicating how many additional blanks may be inserted in a line.

\$ For smaller positive values of J a conflict may arise: the number LFREE of remaining free positions in an output line, may be greater than J, but LW + 1 (where LW is the length of the next word) may be too large:

```
$C
$C      J < LFREE < LW+1
$C
```

Then in interactive mode of operation of RESY the user at the terminal will be asked for a "hyphenation breakpoint": The word which should be hyphenated will be printed and by positioning the printing device of the DIABLO by a suitable number of backspaces and then entering a termination command, the user may indicate where the hyphen should be inserted (cf. section 3.3).

\$ When this option is selected, a new version of the input segment is produced, with the hyphens (encoded as tilde) inserted into the lines of the AMOS segment which can later be used as RESY input again. To the hyphens inserted either in this way or through normal AMOS File Editor updating apply the rules described in 1.3.

\$. RESY Commands

In this chapter we present a detailed description of the RESY commands.

\$ In the first section the syntactic description of the commands is explained. In the following sections, commands are described according to the following classification:

```
$(o)  parameter specification commands
$(o)  report title commands
$(o)  paragraph descriptor commands
$(o)  item list commands
$(o)  display commands
$(o)  layout control commands
```

\$.. Syntactic Description

In this section the general rules for the syntactic description of commands are outlined.

\$ A command starts with a \$ in position 1 of the input line, immediately followed by the command verb, if any. As usual, syntactic entities will be described by words enclosed in <...>. Square brackets, [...] will be used to indicate optional parameters. If a parameter can be omitted, there is a note with each command indicating the default settings.

\$ In addition, the following general rules hold:

\$A 1 Capital letters and special symbols stand for themselves

\$A n, m, and k denote non-negative integers

\$A <string> is a sequence of characters on one input line, starting and ending with a non-blank character (i.e. the string starts with the first non-blank character to the right of the previous syntactic entity encountered, and ends with the last non-blank character of the line)

\$A <text> is a sequence of words (the words not containing any blanks), each pair of words separated by one or more blanks or the end of the input line; wherever <text> is used, the parameter is optional, i.e. it may be omitted.

\$A In a command line syntactic entities and parameters are separated by blanks.

\$ 0

Any further rules will be described when used.

\$.. Parameter Specification Commands

The commands of this group are used to specify the parameters for RESY operation. They describe the page layout, (one or two

columns of text per page, page length, maximum line length, current line length, headline text, page numbering, line numbering, paragraph indentation, justification control, etc.) and whether a table of contents should be constructed.

\$B 1

\$W \$CO n [m n] COLUMNS

The maximum page width is set and it is specified, whether one or two columns are to be printed on each page.

\$B

If there is only one integer n, single column layout is specified: n is the width of that column (i.e. the maximum possible line length). For single column layout, n must be in the range $20 \leq n \leq 120$.

\$B

If there are three integers, double column layout is specified: n is the width for each of the columns and m is the number of blanks between the two columns. For double column layout, m must be in the range $1 \leq m \leq 20$. As there are not more than 120 print positions for one line, n must be chosen that $20 \leq n \leq 59$ and $41 \leq 2*n+m \leq 120$ is valid.

\$B

Initially single column layout with column width 60 is specified.

\$B 1

The "COLUMNS" command performs a major reset:

\$(-) the headline is cleared to blank

\$(-) page numbering is switched off

\$(-) if there is an unfinished page, it is ejected and a new page is initialized

\$(-) the line length is set to n

\$(-) in case of double columns layout, line numbering is suppressed

\$W \$PL n PAGE LENGTH

Specifies the page length by n, which must be in the range $20 \leq n \leq 100$. Initial (default) value is $n=60$.

\$W \$HD [<string>] HEADER DEFINITION

Sets a new value for the headline. Maximum length of the string is 6 less the maximum line width. If the string is omitted, the headline text is reset to blank. If a headline text and/or page numbering have been specified, a headline with the string (left adjusted) and/or the page number (right adjusted) is printed as the topmost line of each page, followed by two blank lines (single column) or four blank lines (double column). These three or five lines are not included in the page length.

\$P

\$W \$PN [n] PAGE NUMBERING

Resets the current value of page number. If parameter n is omitted, no page numbering is requested. If page numbering has been requested and/or a headline text has been defined, a headline is printed as explained above for \$HD.

\$W \$LL n LINE LENGTH

Sets the current text line length LL to value n which must be in the range $20 \leq n \leq LLMAX$. The new setting becomes valid after the current output line being assembled in the buffer has been printed and a new line is started. Initial (default) value is $LL=60$ ($LLMAX=72$).

\$W \$LN [n] LINE NUMBERING

Specifies whether or not the latest input line number (columns 73-80 of the input line) is printed on the left-hand margin

of the output line. If $n=0$, no line numbering is requested; $n>0$ or n omitted requests line numbering. Line numbering is not effective, if double column layout is specified.

`$W $PI n` PARAGRAPH INDENTATION
Sets the parameter INDENTP for paragraph indentation; n must be in the range $0 \leq n \leq 10$. A new value is effective when a new paragraph starts. Default value is 3.

`$W $JC [n]` JUSTIFICATION CONTROL
Sets the parameter J for line justification control to n . If n is omitted, J is set to -1 which means: no justification. If $n>0$, it specifies the maximum number of blanks that may be inserted into a line during justification. If the length LW of the next word to be put into the output line buffer is such that

`$C`
`$C J < LFREE < LW,`
`$C`

then hyphenation (cf. 1.4) will be requested. If $J > LLMAX$, hyphenation obviously never will be requested. Default value is $J=100$.

`$W $CN n` CHAPTER NUMBER
Set initial value for first chapter to be printed. Default value is 0.

`$..` Report Title Commands
This group of commands is used to create a title page of the report and, optionally, a table of contents. Though these commands should appear in the beginning of the input of a report to RESY, printing of the title page and the table of contents is delayed until the report has been completed.

`$`
The report title page has a format as shown on the title page of this report. The top line of the title page may optionally contain the report number and the current date (of the day of printing) or a specified date. There are three lines reserved for the title of the report (i.e. unspecified lines are printed as blank lines) and up to 10 lines for the authors. An additional maximum of 10 lines may be used to print a summary of the report. Finally, the automatic production of a table of contents (consisting of chapter, section, subsection, and appendix headings, and containing up to 60 lines) may be requested. It is printed either on the title page (if there is still room left) or on a new page. The title page has the format of 60 lines with 60 columns each.

`$W $RN <string>` REPORT NUMBER
Defines the report number, consisting of a string of characters of maximum length 40. Will be printed left-justified on top line of title page.

`$W $RD [<string>]` REPORT DATE
Defines report date as string of maximum length of 12 characters. If input string is blank, the date is obtained from the system and printed in the form

`$C`
`$C Feb 12, 1978`
`$C`

`$W $RT <string>` REPORT TITLE
Defines the next line of the title. The string may have a maximum length of 60 characters. Up to three title lines may be defined which will be printed centered and separated by one blank line.

`$W $RA [<string>]` REPORT AUTHOR

Defines the name (and affiliation) of the next author. The string may have a maximum length of 60 characters. It is printed centered. Up to 10 author lines may be defined.

\$W \$RS <text> REPORT SUMMARY

Defines the summary of the report. It is printed following the author list as an unindented paragraph of up to 10 lines, 50 columns wide and indented 5 places. The text of the summary may start on the command line and may be supplemented by more text input lines. The text of the summary is terminated by any other command. The layout of the summary does not disturb the current definitions of line length and indentation.

\$W \$RC REPORT TABLE OF CONTENTS

Generates a table of contents consisting of chapter, section, and subsection headings. The latter are indented 3 and 6 places, respectively. Blank lines are inserted where appropriate. The table of contents is printed on lines 60 columns wide with the page numbers right-justified.

If appendices have been specified, the headings of those are also printed in the table of contents.

\$.. Paragraph Descriptor Commands

In this section the paragraph heading commands are described. They are used to generate paragraphs with and without headings, at the same time concluding any previous paragraph. As a rule, a paragraph ends with a (possibly not completely filled) line, which is left non-justified, and a blank line.

\$ If a paragraph with a heading is generated, care is taken by RESY that the heading does not appear without at least part of the paragraph at the bottom of a page. Otherwise an early skip to a new page is performed.

\$ The structure of the input for a paragraph is always the same: a paragraph heading command is followed by a sequence of text lines up to (but not including) the next command line.

\$W \$. <string> CHAPTER HEADING

The last line of the previous paragraph is printed. If there are less than 10 lines left on the current page, start new page. Print 3 blank lines, increase the chapter number by 1, reset the section and subsection counts to 0, and print (left-justified) the chapter number and the <string> as chapter heading, followed by 2 more blank lines. Any <text> input lines that follow are printed as the body of a paragraph, the first line of which starts indented INDENTP spaces.

\$W \$.. <string> SECTION HEADING

The last line of the previous paragraph is printed. If there are less than 8 lines left on the current page, start a new page. Print 2 blank lines, increase the section count by 1, reset the subsection count to 0, and print (left-justified) the section number and <string> as section heading, followed by a blank line. Any <text> input lines that follow are printed as the body of a paragraph, the first line of which starts indented INDENTP spaces.

\$W \$... <string> SUBSECTION HEADING

The last line of the previous paragraph is printed. If there are less than 6 lines left on the current page, start a new page. Print a blank line, increase the subsection count by 1, and print (left-justified) the subsection number and <string> as subsection heading, followed by a blank line. Any <text> input lines that follow are printed as the body of a paragraph, the first line of which is indented INDENTP spaces.

\$W \$T [<string>] TITLED PARAGRAPH

The last line of the previous paragraph is printed, followed by a blank line. If there are less than 4 lines left on the current

page, start a new page. Print the <string> left-justified as the title of a paragraph, followed by a blank line. Any <text> input lines that follow are printed as the body of a paragraph, the first line of which is indented INDENTP spaces.

\$W \$ [n] <text> PARAGRAPH

The last line of the previous paragraph is printed, followed by a blank line. Then a new paragraph is started, taking <text> and any following <text> lines as the text of the paragraph. The first line of the new paragraph is indented n spaces (n must be in the range $0 \leq n < LL$). If n is omitted, indentation is INDENTP spaces. Because n may be omitted, <text> must not start with an integer. (If a paragraph with standard indenting does start with an integer, either omit n and <text> and put the text on the next line, or set n=INDENTP and use <text> in any form.)

\$W \$AP APPENDIX

The last line of the previous paragraph is printed. Chapter numbering is reset to zero and appendix processing is initiated.

\$W \$. [<string>] APPENDIX HEADING

This command produces an appendix heading if the \$AP command has been issued. The heading is printed in the form:

\$D #

Appendix 1: [<string>]

etc. The commands \$.. and \$... are not permitted in appendices.

\$.. Item List Commands

The five commands of this group are used to create "item lists" to enumerate items. Two of the commands may be used to generate automatically numbered lists.

An "item list" normally is ended when the next command is encountered; exceptions are the break command \$B, and the display commands \$D and \$C. The latter use the indentation position as left-hand margin.

\$W \$(x) <text> ITEM, EXPLICITLY NUMBERED

x is a string of one or two characters, not containing a right parenthesis. Print last line of previous paragraph and a blank line. Print (x) or (xx) followed by <text> and indent all following text lines (up to next command) 5 additional spaces. Treat the item text like a paragraph (i.e. conclude with an extra blank line when next command is encountered).

\$W \$(z) <text> ITEM, MARKED BY SPECIAL SYMBOL

z is either of the symbols . or - or o enclosed in parentheses. Print last line of previous paragraph followed by a blank line. Print z followed by text lines (up to next command) printed indented by

5 additional spaces. Treat item text like paragraph (i.e. conclude with an extra blank line when next command is encountered).

\$W \$V <string> VERBAL LIST (SHORT)

Create list where items are "numbered" by words. Print last line of previous paragraph, followed by a blank line. Print <string> followed by any text from ensuing <text> input lines. Indent all following text lines by k+1 spaces, where k is the length of <string>, up to the next command.

Conclude with an extra blank line when next command is encountered.

\$W \$W <string> VERBAL LIST (LONG)

Create a list where items are defined by a title line consisting of <string>.

Print any text lines indented 5 places.

<text> input lines are treated like a paragraph.

Conclude with an extra blank

line when next command is encountered.

\$W \$N [n] <text> LIST NUMERICALLY

cf. the description of \$A following in the next paragraph.

\$W \$A [n] <text> LIST ALPHABETICALLY

Create a list of items where items are numbered automatically, either by numbers increasing by 1 (in the range 0,1,...,20) or by the consecutive letters a, b, c,... of the alphabet. Print last line of previous paragraph, followed by a blank line. If n is present, reset item counter to indicated value n if a numeric list is desired; in case of an alphabetic list n must be 1 for initialization. If n is omitted, increase the list parameter by 1. Print (x) or (yy) where x is a digit or a letter and yy are numbers in the range between 10 and 20. Indent all following text lines additional 5 spaces; treat item <text> and following text lines like a paragraph.

Conclude with an extra blank

line when next command is encountered.

\$.. Display Commands

There are two types of display commands. They serve to copy information from input lines to output lines, one by one. They should be used whenever information is to be displayed in fixed format. Both commands cause a previous line to be terminated.

\$W \$D [n [m]] [x] DISPLAY

Copy information from the input lines that follow into the output lines using the current indentation as left margin; the parameters, if present, have the following meaning:

\$I 5

\$V n -
delete first n characters from each input line;
if omitted default is n=0

\$V m -
take next m characters from each input line;
if omitted default is m=remainder of line

\$v x -
is the escape character that ends the display
sequence, if found as first character of an
input line; x may not be a digit unless n and m are specified, the
default is x="\$".

\$I 0

\$P

\$W \$C [<string>] CENTER

Any previous line is terminated and printed. Then the <string>
is printed centered on a new line using the current
indentation as left margin.

\$.. Layout Controlling Commands

The commands of this group are used to control the layout of the
output e.g. left margin control, and empty lines or pages.

\$W \$I [n] INDENTATION

The indentation constant in use is reset to n. All following lines -
starting with the next output line - are printed with the left margin
indented n positions with respect to the leftmost printing position.
With n not defined or n=0 the leftmost print
position on the line is addressed.

\$W \$B [n] BREAK and INSERT BLANK LINES

The last line is ended without right justification; then n empty
lines are inserted. If n is omitted, it is assumed to be

zero.

\$W \$L [n] INSERT BLANK LINES

After the next line has been printed, n empty lines are inserted. The default value for n is 1.

\$W \$FP [n] INSERT FREE PAGES

When a page has been completed, n free pages are inserted. The default value for n is 1.

\$W \$P TERMINATE PAGE

After the current line has been completed, start a new page. If requested before, the headline and page number are printed.

\$. How to operate RESY

At the IPP Computer Installation

RESY may be run either as a job under OS/MVT without terminal control or interactively with a DIABLO 1620 Terminal connected to an INTERDATA.

\$ It is assumed that the user is familiar with the AMOS system, as RESY input usually will be prepared with the AMOS File Editor and job submission also will be carried out via AMOS.

\$.. Using RESY with a Batch Job

RESY may run as a batch job under OS MVT on the IBM 360/91 (MVT/91) or under OS MVT on the AMDAHL 470 V/6 (MVT/470).

The necessary JCL statements to execute the RESY load module are stored in the AMOS segment SYS:JOBS.RESY.

\$ As output device for the RESY job either a line printer with upper and lower case character printing or - in addition - an AMOS segment may be used. The only high speed line printer with upper and lower case characters available at IPP computer center is usually attached to the IBM 360/91.

Therefore, a RESY job should be submitted to MVT/91 (AMOS command S91 before job submission).

\$ There are two ways to let the RESY job handle input and output:

\$N 1 the text input is read from the OS SYSIN data set; the edited report is printed onto the line printer.

\$D #

To MVT/91 XSUBMIT a segment containing:

```
    $$ SYS:JOBS.RESY
```

#

followed by the text of the report, or a SUBSTITUTE statement referencing the segment which contains the report text.

\$N the text input is not submitted with the control statements, but is read directly from an AMOS segment while the RESY job is running. The edited report is printed on the line printer and also written into an AMOS segment (cf. 3.2).

\$D #

XSUBMIT a segment containing:

```
    $$ SYS:JOBS.RESY
    // EXEC RESY,SOURCE='<segment specification>',
    // REPORT='<segment specification>'
```

where

#

\$ <segment specification> ::= <userid>:<filename>.<segname>

\$ If the parameter REPORT is omitted, no AMOS segment is

written.

\$ In order to obtain reproducible output for offset printing etc., the DIABLO printer of Remote Station 8 can be used. Remote Station 8 is located at building D2/1st floor and is attached to MVT/470.

\$ The RESY job should be submitted to MVT/470 with the job setup preceded by two additional control commands:

\$D

```
/*ROUTE PRINT REMOTE8
/*CONTROL FORM=1
```

\$.. Printing a Report with the DIABLO Terminal

The segment generated by RESY can only be printed with a special job using a DIABLO terminal

connected to VM/370. As an AMOS user, submit this job to MVT/470 with the following AMOS commands:

\$D

```
?S470
?SUBMIT SYS:JOBS.DIAPRINT
PROBLEM?
TIME?
REGION?
FURTHER PARAMETERS?CLASS=L
```

\$ 0

When the job has reached the go-step, press the "BREAK" key on a VM/370 DIABLO terminal and type in the command:

\$C

\$C DIAL MVT 3D

\$C

The segment name and a few print control parameters have to be specified at the terminal. Before the last question (start printing?) is answered, the DIABLO should be positioned to print directly above the next horizontal paper perforation. After all printing has been completed, empty input will end the print job and switch the terminal back to VM/370.

\$.. Using a DIABLO 1620 as RESY Terminal
Implementation and description are forthcoming.

\$B 15

\$T Acknowledgements

The authors wish to thank Mrs. Inge Precht for useful discussions. We also appreciate the work of Mrs. Marianne Glandien, who typed this report in "RESY".

\$HD A P P E N D I C E S

\$P

\$AP

\$LL 60

\$. Summary of RESY Commands

\$D

Parameter Specification Commands

\$CO n [m n]	Column Setting
\$PL n	Page Length
\$HD [<string>]	Header Definition
\$PN [n]	Page Numbering
\$LL n	Line Length
\$LN [n]	Line Numbering
\$PI n	Paragraph Indentation
\$JC [n]	Justification Control
\$CN n	Chapter Numbering

Report Title Page and Table of Contents

\$RN <string>	Report Number
\$RD [<string>]	Report Date
\$RT <string>	Report Title
\$RA [<string>]	Report Author
\$RS <text>	Report Summary
\$RC	Generate Table of Contents

Paragraph Descriptor Commands

\$. <string>	Chapter Heading
\$.. <string>	Section Heading
\$... <string>	Subsection Heading
\$T <string>	Titled Paragraph
\$ [n] <text>	Paragraph, n Indentation
\$AP	Appendix
\$. [<string>]	Appendix Heading

Item List Commands

\$(n) <text>	List Item, Explicit Number
\$(.) <text>	List Item, Symbol (. - o)
\$V <string>	List Item, Verbal (short)
\$W <string>	List Item, Verbal (long)
\$N [n] <text>	List Item, Numerical
\$A [n] <text>	List Item, Alphabetical

Display Commands

\$D [n [m]] [x]	Display Begin
x	Display End, default x="\$"
\$C [<string>]	Center

Layout Controlling Commands

\$I [n]	Indentation
\$B [n]	Break, Insert n Blank Lines
\$L [n]	Insert n Blank Lines
\$P	Terminate Page
\$FP [n]	Insert n Free Pages

\$. Error Indicators

In case RESY detects an error in an input line, this line is printed with an error indicator in the gap between line number and the first position of the line.

\$ Note that error indicators are not stored in a RESY generated AMOS segment.

\$ List of error indicators and their meaning:

\$(TR) line truncated;

\$(--) line discarded i.e. too many lines in case of Report Summary;

\$(**) parameter error in command line;

\$(??) command unknown.

\$P

\$HD Sample Input Text

\$. Sample Text

The text on the following pages is the input to RESY of this report. A comparison of this input with the output as produced by RESY should help the user to understand how RESY commands

are used and what their effect is.

\$P

\$LL 72