



# VP DATA CAPTURE

**XEROX**

*VP Series Reference Library*  
*Version 1.0*

---

*This publication could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.*

*This publication was printed in September 1985 and is based on the VP Series 1.0 software.*

*Address comments to:*

Xerox Corporation  
Attn: Product Education (N2-15)  
701 S. Aviation Blvd.  
El Segundo, California 90245

**WARNING:** *This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to subpart J of part 15 of the FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.*

*Printed in U.S.A. Publication number: 610E06690*

*XEROX®, 6085, 8000, 8010, 860, 820-II, 8040, 5700, 8700, 9700, 495-1, ViewPoint, and VP are trademarks of Xerox Corporation.*

*IBM is a registered trademark of International Business Machines.*

*DEC and VAX are trademarks of Digital Equipment Corporation.*

*Wang Professional Computer is a trademark of Wang Laboratories, Inc.*

*Lotus 1-2-3 is a trademark of Lotus Development Corporation.*

*MS-DOS is a trademark of Microsoft Corporation.*

*The Source is a service mark of Source Telecomputing Corporation.*

*VisiCalc is a registered trademark of VisiCorp.*

*WordStar is a registered trademark of MicroPro International Corporation.*

*Teletype is a registered trademark of Teletype Corporation.*

**© 1985, Xerox Corporation. All rights reserved.**

# Table of contents

---

<b>1. VP Data Capture overview</b>	<b>1</b>
Related information	2
VP Series reference library	2
VP Series training guides	2
Hardware/software requirements	3
Data capture overview	4
Who uses data capture?	6
Source documents	7
Preambles	7
<b>2. Actions and procedures</b>	<b>11</b>
Loading data capture software	12
Preparing source documents	12
Writing preambles	13
Field length and format	15
Comment and name	15
How to structure preambles	16
Preamble appearance	17
Reusing preambles with [SET PREAMBLE]	17
Special formatting characters	18
Tips on writing preambles	19
Activating data capture	20

Applying data capture to folders	20
Ignoring extraneous text	21
Noticing key text	23
More about literals	24
Using literals to indicate date formats	25
Specifying uppercase or lowercase	26
Capturing multi-line rows, vertical layout	27
Indicating absent fields in the source text	28
Capturing subrows and divided columns	29
Capturing combinations of subcolumns	32
Ignoring indentation in the source text	35
Subtotals, fields "after" a group	36
Defining column order	39
Content of source text	40
All main columns are after a group	41
Limitations of data capture	42
<b>3. Property/option sheets and windows</b>	<b>45</b>
<b>4. Error Messages</b>	<b>47</b>
Unacceptable preambles	47
Acceptable preambles	49

# List of figures

---

1-1	Unstructured text	5
1-2	Table	6
1-3	Preamble	8
1-4	Preamble with character counter	9
2-1	Preamble	14
2-2	Preamble with name and comments	16
2-3	Extraneous text ignored	21
2-4	Source document	23
2-5	Preamble	24
2-6	Source document	25
2-7	Source document with multi-line rows	27
2-8	Source document	28
2-9	Repeating groups	30
2-10	Table with subcolumns	31
2-11	Combinations of subcolumns	32
2-12	Preamble for combinations of subcolumns	33
2-13	Source document with subcolumns	34
2-14	Grouping by indentation	35
2-15	Source text	36
2-16	Table	37
2-17	Preamble using "after" property	38
2-18	Source document with main columns after a group	41

*(This page intentionally blank)*

# 1.VP Data Capture overview

---

VP Data Capture software provides a simple and flexible way to take unstructured text (such as employee salary records) from a TTY, VT100, or 3270 terminal emulation window; place it in fields; and create a table from the fields. Once text has been transferred to a table, it can be edited and the table can be reformatted and inserted into a report. Data capture can also be used to create data-driven graphics, such as bar charts.

---

## Related information

---

The following materials provide information related to the VP Data Capture application.

### VP Series reference library

---

- Xerox ViewPoint
- VP Document Editor
- VP List Manager

### VP Series training guides

---

- VP Document Editor
- VP Data Capture

---

## Hardware/software requirements

---

The following are the hardware and software requirements for VP Data Capture:

- A 6085 Professional Computer System, or an 8010 Information System
- Xerox ViewPoint software
- VP Document Editor software
- VP Data Capture software

**Note:** To connect to a mainframe running TTY, 3270, or VT100 protocols to obtain unstructured text, VP Terminal Emulation of TTY, IBM 3270, or DEC VT100 software is also required.

VP Data Capture software and all prerequisite software must be installed, enabled, and running on the workstation. Before using the software, open the application loader icon and verify that the appropriate software is loaded and running.

The sub-tab titled "Application Loader" in the VP Series reference library contains additional information on the application loader.

## Data capture overview

---

Data capture is a way of moving data from a terminal emulation window (VT100, teletypewriter, or 3270) into a table. It is a bridge between the workstation's ability to capture data from a remote host data processing system through terminal emulation capabilities and its ability to display and organize that data in tables, record files, and bar charts.

Before the data capture software was made available, the quickest way to get information from a 3270, teletypewriter (TTY), or VT100 terminal emulation window was to copy it, one field at a time. With data capture software, users can now transfer data from a remote data processing system to a table within a document. Once the data is transferred to the table, the user can edit the data, make formatting changes to the table (such as changing the column widths or tab settings), or even transfer the data to a record file or bar chart for use in mailing lists or reports.

Additionally, for applications and databases that are to be maintained on a host processing system, data capture can be used periodically to take a database snapshot to manipulate at the workstation.

Figure 1-1 is an example of unstructured text that might be generated from a host data processing system.

PERSONNEL REPORT FOR 12/23/82  
MARKETING DEPT.

EMPLOYEE	HIRED	YTD EARNINGS
123456789012345678901234567890123		
Smith, A.	02-11-68	12,778.23
Jones, E.	07-12-82	12,622.11
Price, W.	02-13-68	23,718.23
Liou, D.	12-11-78	26,737.00
Smith, D.	12-21-82	0.00
Anderson, P.	02-11-68	22,778.91
Green, F.	05-05-55	27,128.03
Charles, R.	07-23-80	22,778.23
Leary, T.	02-13-68	23,718.42
Fredholm, R.	12-11-78	22,737.00
Abel, C.	12-11-82	812.00
Carothers, G.	02-11-68	31,778.12
Blue, V.	03-07-75	19,828.03
Priam, H.	07-29-80	26,772.01

Figure 1-1 Unstructured text

Figure 1-2 is an example of a table that could be produced using data capture. Only two of the three columns from the source document shown in Figure 1-1 have been transferred to a table. In general, data capture lets users transfer several or all of the columns of information.

Smith, A.	12,778.23
Jones, E.	12,622.11
Price, W.	23,718.23
Liou, D.	26,737.00
Smith, D.	0.00
Anderson, P.	22,778.91
Green, F.	27,128.03
Charles, R.	22,778.23
Leary, T.	23,718.42
Fredholm, R.	22,737.00
Abel, C.	812.00
Carothers, G.	31,778.12
Blue, V.	19,828.03
Priam, H.	26,772.01

Figure 1-2 Table

## Who uses data capture?

---

Data capture is generally used after a terminal emulation session. Users who have VT100, 3270, or TTY emulation software loaded on their workstations and would like to transfer text from the emulation window to a table will find data capture software very useful.

System Administrators can use data capture software to help organize Clearinghouse databases and other information that can be captured.

## Source documents

---

A source document is a document that contains the data to be used in a data capture operation. Although the data capture software accepts any 6085 or 8010 document as a source document, source documents are generally created as a result of selecting [MAKE DOCUMENT] in the terminal emulation window.

One reason for using the source document that appears after [MAKE DOCUMENT] has been selected in the terminal emulation window is because it contains the raw data that has been transferred from the terminal emulation window. Another reason for using a source document is because data capture only works with unstructured text that contains no tabs or margins. A document that has been prepared on a 6085 or 8010 workstation normally contains tabs and margins; it is, therefore, unacceptable to the data capture software. Data obtained from a TTY, 3270, or VT100 data processing system, on the other hand, uses a single format and spaces to break data into columns.

## Preambles

---

In addition to containing the text transferred from a terminal emulation window, a source document must contain a preamble. A preamble is a set of instructions that specifies what columns and rows should be copied to a table.

Preambles are made up of sentences. Each sentence is made up of clauses that are enclosed in parentheses. Except for the last sentence, (End of Preamble), which is a special indicator declaring the end of the preamble, each sentence specifies a column to be captured and placed in a table.

An example of a preamble written for the data captured in Figure 1-2 is shown in figure 1-3.

```
(Character 1)(Type text)(Format xxxxxxxxxxxxxxxx).  
(Character 25)(Type amount)(Format bb,bb9.99).  
(End of preamble).
```

### Figure 1-3 Preamble

In the preamble shown in Figure 1-3, there are three sentences, each terminated by a period. The first two sentences are made up of three clauses (enclosed in parentheses). The first sentence corresponds to data to be captured in the Employee column, and the second corresponds to data to be captured in the YTD Earnings column. The last sentence (End of preamble) declares that the preamble is complete.

Each clause within a sentence consists of two parts: a property of a column and a value for that property. For example, in the clause (Type amount), the property is type and the value is amount. All clauses work this way. The first word identifies a property, and the rest of the clause specifies a value for it. Thus, the preamble can be thought of as a way of writing a property sheet in ordinary text.

### The character clause

---

Within a preamble, the character clause tells the software where in the source text to look for the data corresponding to each column of the table being produced. The software finds the data to capture by counting character spaces. It then picks up the data described in the type clause (as text, a date, or an amount).

In the example shown in Figure 1-3, (Character 1) is the first character clause in the preamble. It tells the software to begin capturing the data at character 1 of the source text. The second character clause is (Character 25) in the second sentence. It tells the

software to begin capturing data for the second column at character 25.

To determine what number to use, the user must count character position from the lefthand margin of the source text and specify the position at which the field starts.

A character counter typed in a fixed pitch font (such as pica or vintage) across the source document makes character counting simpler.

The preamble in Figure 1-4 contains a character counter.

```
(Character 1)(Type text)(Format xxxxxxxxxxxxxxx).  
(Character 25)(Type amount)(Format bb,bb9.99).  
(Comment  
1234567890123456789012345678901234567890123456)  
(End of preamble).
```

Figure 1-4 Preamble with character counter

The character counter is enclosed in parentheses and preceded by: Comment. In preambles, a comment is an explanatory message intended only for other users who may read preambles. Statements preceded by a comment are ignored by the data capture software.

Another property that can be used to make a preamble easier for others to read is the name property. This property can be used to help indicate the intent of a sentence and of the corresponding table column. If a name is specified, the corresponding column of the table will have that name.

## Special formatting characters

---

In preambles, the format clause determines the width of the field in the source text. In the preamble shown in Figure 1-4, the Employee column is limited to 13 characters (13 x's). The x's are special formatting characters that tell the data capture software that the corresponding field position can contain any character.

Other special formatting characters used in data capture include:

- A      The corresponding field position can contain a letter or a blank.
  
- 9      The corresponding field position can contain a digit or a blank.
  
- Y      The corresponding field position can contain any character except a blank.

These and other special formatting characters help define the data that should be transferred to a table. A complete list of special formatting characters is provided in Appendix A.

## **2. Actions and procedures**

---

The procedures and actions contained in this chapter provide the information necessary to use VP Data Capture software. Included are instructions on how to load the software, create source documents, and write preambles.

---

## Loading data capture software

---

If data capture has never been run on your workstation before, you must load the software. If you are not sure whether it has been loaded, look in the desktop auxiliary menu. If the commands [COPY TEXT TO TABLE], [SET PREAMBLE], and [DISCARD PREAMBLE] are among the options, then the software has been loaded. If not, refer to the section titled "6085 Software Installation" or "8010 Software Installation" in the VP Series reference library.

---

## Preparing source documents

---

Before data capture can be activated, it is necessary to prepare a source document. A source document is the document which contains the data that will be used in the data capture operation. Although data capture will actually accept any 6085 or 8010 document as a source document, the document that results from selecting [MAKE TABLE] in the terminal emulation window should be used. This is because data capture only works with unstructured text, which is text that contains no tabs or margins. A document that has been prepared with tabs and margins is not acceptable to the data capture software. On the other hand, data from TTY, VT100, or 3270 emulation uses a single format and spaces to break data into columns. Data capture relies on this structure to convert data into tables.

To prepare a source document:

1. Ensure that TTY, VT100, or 3270 emulation is running on the workstation and data from the host is displayed in the emulation window.
2. Select [MAKE DOCUMENT] in the terminal emulation window. The resulting document is the source text to be used. Do not attempt to reformat the document before capturing it. It does not matter if the window is wide enough or

if the font is appropriate. There is no need to paginate.

3. Open the source document.
4. Write the preamble. A preamble is a set of instructions that specifies the structure of the columns and rows to be copied to a table. Refer to the section titled "Writing preambles" for instructions on writing preambles.
5. Insert the preamble at the very beginning of the source document. You can do this by typing in the preamble, but there is a better way. Keep the preamble in another document. When it is needed, copy the preamble into the beginning of the source document. That way, you can use a preamble over and over again.
6. When you are satisfied with the preamble, close the document.
7. Select [COPY TEXT TO TABLE] in the desktop auxiliary menu to activate data capture software.

---

## Writing preambles

---

A preamble is a set of instructions that specifies what columns and rows should be copied to a table. It is a way to write a property sheet using ordinary text. For example, the table column property sheet contains a property called **type**. One value of that property is **text**. In a preamble, you would indicate that value by writing: (Type text).

Figure 2-1 contains an example of a preamble.

```
(Character 1)(Type text)(Format xxxxxxxxxxxxxxxx).  
(Character 25)(Type amount)(Format bb,bb9.99).  
(End of preamble).
```

### Figure 2-1 Preamble

Three sentences make up the preamble. Each is terminated by a period. The sentences are made up of clauses, which are enclosed in parentheses.

The last sentence is a special indicator (End of preamble), which declares that the preamble is done and the text to be captured is to follow.

In preambles, each sentence (except End of preamble) corresponds to a column in the source text to be captured. The clauses within each sentence specify the properties and their values. A clause, in fact, is very similar to an option on a property sheet. Each clause consists of two parts: a property of a column and a value for that property. For example, in (Character 1), the property is character and the value is 1. All clauses work this way. The first word identifies a property and the rest of the clause specifies a value for it. A sentence can have any number of clauses.

In Figure 2-1, three properties are expressed in clauses. They are:

- **Character.** Tells the software where in the source text to look for the data corresponding to each column of the table being produced. To determine what number to use, count the character positions from the lefthand margin of the source text and specify the position at which the field starts.
- **Type.** Specifies the data type of the field. Valid choices are text, amount, and date.
- **Format.** Describes what the source text looks like. The format length determines the width of the field in the source text, in characters. For example, (Format xxxxxxxxxxxxx) limits the length of the column to 13 characters. The x's are special formatting characters that are described in the section titled "Special formatting characters."

**Note:** In formats, blank spaces are not ignored. Only one blank space can separate the word "format" from the formatting characters.

## Field length and format

---

There is a shorthand you can use to avoid long strings of x's in simple text formats. Instead of using the format property, use the length property. The value to use is the number of x's you would include in the format. For example, you could write (Length 8) instead of (Format xxxxxxxx). Length can likewise be applied to amount formats; (Length 8) can be used instead of (Format -bbbbbbb).

## Comment and name

---

Two other properties, **comment** and **name**, help make the preamble more readable. A comment is an explanatory message. Comments are ignored by the data capture software. A name is used to name the corresponding column of the table produced by the

software. Figure 2-2 shows a preamble that uses name, comments, and a character counter.

```
(Comment      Preamble prepared by J. D.
Sullivan.)

(name      EMPLOYEE NAME)      (Format
xxxxxxxxxxxxxxxx)
                                (Comment  Last-Name comma
First-Initial).

(name      YEAR-TO-DATE SALARY)
              (Type Amount) (Format BB,BB9.99)
(Character 25).
(Comment
12345678901234567890123456789012345678901234567
89).
(End of Preamble).
```

Figure 2-2 Preamble with name and comments

---

## How to structure preambles

---

Each preamble may look different, depending on what data is being captured from a TTY or 3270 emulation window. The basic steps for writing preambles must be tailored to fit individual needs.

To write a preamble:

1. Create a source document by selecting [MAKE DOCUMENT] in the TTY, 3270, or VT100 window during an emulation session.

**Note:** Refer to the section titled, "Preparing source documents" for instructions on creating source documents.

2. Insert the preamble at the beginning of the source document by typing in the preamble or

copying a prewritten preamble from another document.

3. Insert any comments or names within parentheses.

To make character counting simpler, enter a character counter in a fixed pitch font, such as pica or vintage, across the source document as follows:

1. Select an appropriate spot in the document against the left margin.
2. Type numbers from 1 to 0 (1234567890).
3. Press <AGAIN> until the numbers span the width of the data.

## Preamble appearance

---

In a preamble, either uppercase or lowercase letters can be used, or they can be intermixed. Case does not matter. Additionally, spaces or tabs can be used to help make the preamble more readable; these are ignored by data capture software. Any font or character size can be used. There is no need to try limiting a sentence to one line. The period delimits a sentence; line breaks do not matter. Also, the order of the clauses within a sentence does not matter.

## Reusing preambles with [SET PREAMBLE]

---

Once a preamble has been written, it can be used in any [COPY TEXT TO TABLE] operation. To reuse a preamble, select a source document that contains a preamble, and then select [SET PREAMBLE] in the desktop auxiliary window. If it is a legal preamble, it is established as the "set" preamble and remains in effect until you log off, the workstation is rebooted, or [DISCARD PREAMBLE] is selected in the desktop auxiliary menu.

If a preamble is already set when [SET PREAMBLE] is selected, the old preamble is discarded in favor of the

new one. The message "Removing preamble [YES] [NO]" appears in the message area before the preamble is discarded.

If a preamble in the selected document is not legal, the currently set preamble (if one exists) is not affected.

## Special formatting characters

---

A format contains special formatting characters. These are the "box" characters in field formats, analogous to the Cobol picture clause.

### Text formats

---

The special formatting characters are already defined for U. S. English fields. They have the customary meanings:

- A The corresponding field position can contain a letter or a blank.
- 9 The corresponding field position can contain a digit or a blank.
- X The corresponding field position can contain any character.
- Y The corresponding field position can contain any character except a blank.

### Amount formats

---

The following special formatting characters are already defined for U.S. English fields. They have the customary meanings:

- 9 The corresponding field position can contain a digit.
- B The position contains a digit unless the digit would be an insignificant zero, in which case the

position contains a blank. (An insignificant zero is a leading zero to the left of the decimal point or a trailing zero to the right of the decimal point.)

- , The corresponding field position contains a comma, unless the preceding digit is suppressed because it is insignificant.
- . The corresponding field position contains a decimal point.

## **Tips on writing preambles**

---

The following hints and tips will help you to write preambles.

- Enter a character counter across the document in a fixed pitch font to simplify the process of counting characters in the preamble.
- If too much data is being transferred in the table (headings or other undesirable information from a mainframe), make the format statements more strict by requiring alphabetical (a) or numerical (9 or b) characters only. The new y instruction (any character except a blank), can also help narrow down the information put into a table.
- If the resulting table is blank and the preamble contains format statements that have amounts with decimal points, examine your format statements carefully. Perhaps the software has been instructed to find a decimal point at a certain position, when in reality the decimal point is one character to the right or left.
- A blank table occurs when format statements are too strict and the character count is off, even if only by one character. If the format statement allows for numbers up to 999,999.99, but the highest number is 99,999.99, chances are the decimal point is one character off.

---

## Activating data capture

---

With the source document selected, select [COPY TEXT TO TABLE] in the desktop auxiliary menu. The software does the rest. The hourglass cursor appears, and progress through the operation is indicated in the message area. At the conclusion of the copy operation, a new document is placed nearby on the desktop. It contains a table that is the result of the data capture operation.

In case the preamble is invalid, an appropriate description of the problem appears in the message area. It specifies the nature of the problem and the place in the preamble where the anomaly occurred. To resume work, open the source document, correct the preamble, close the document, and select [COPY TEXT TO TABLE] again.

Pressing <STOP> cancels the operation, but it may result in a partially-completed table.

---

## Applying data capture to folders

---

When a folder is selected and [COPY TEXT TO TABLE] is selected, a folder appears on the desktop. Unless an error is encountered, the folder has exactly the same structure and substructure as the original. All levels in the folder are duplicated.

Data capture software names folders created using [COPY TEXT TO TABLE] in one of the following ways:

- If there is no preamble currently set, the resulting folder name is exactly the same as that of the original folder.
- If a preamble is currently set, the resulting folder name is the same as the original folder, followed by a dash, and the name of the document from which the set preamble came.

## Ignoring extraneous text

Data capture software allows you to capture only text that is useful and copy it to a table. Extraneous text is ignored. In Figure 2-3, for example, the date hired has been ignored and the employee name and earnings have been captured and placed in a table.

Source Document

(Character 1)(Type text)(Format  
xxxxxxxxxxxxxxxx).  
 (Character 25)(Type amount)(Format  
bb,bb9.99).  
 (Comment  
 123456789012345678901234567890123)  
 (End of preamble).  
 PERSONNEL REPORT FOR 12/23/82  
 MARKETING DEPT.  
 EMPLOYEE      HIRED      YTD EARNINGS  
 Smith, A.      02-11-68      12,778.23  
 Jones, E.      07-12-82      12,622.11  
 Price, W.      02-13-68      23,718.23

Table

Smith, A.	12,778.23
Jones, E.	12,622.11
Price, W.	23,718.23

Figure 2-3 Extraneous text ignored

The data capture software ignores the extraneous text in this example by looking at the format clause. The format clause for this example is (Format bb,bb9.99). The software ignores the text that is incompatible with the special format characters, bb, bb9.99.

In general, all the formats of all the fields must be compatible with the source text in order for a row to be stored in the table. (Source text that is all blank is always considered incompatible.) Thus, the formats you write keep extraneous rows out of the table. In the example, each of the text lines (including the blank ones) would have yielded a table row if it were not for this feature. The source text might contain other kinds of extraneous text generated by the host data processing system, including page headings/footings and page numbers. They are filtered out by the same mechanism.

In order to let the software do the job of ignoring extraneous text, make the format as explicit as possible. If in the example both columns were: (Type text) (Format xxxxxxxxxxxx), then the table would have contained extraneous text.

## Noticing key text

Most of the time, the special format characters suffice to identify the text for data capture. Sometimes, though, more context is required. Suppose it is necessary to capture only subtotals for the eastern and central regions in Figure 2-4.

```

Region: Eastern
EMPLOYEE   HIRED   YTD EARNINGS
Smith, A.   02-11-68   12,778.23
Jones, E.   07-12-82   12,622.11
O'Brien, W. 02-13-68   23,718.23
Total Eastern                49,118.57

```

```

Region: Central
EMPLOYEE   HIRED   YTD EARNINGS
Liou, D.    12-11-78   26,737.00
Smith, D.   12-21-82         0.00
Anderson, P. 02-11-68   22,778.91
Green, F.   05-05-55   27,128.03
Charles, R. 07-23-80   22,778.23
Total Central                99,422.17

```

Figure 2-4 Source document

The technique is to notice the word **Total** and indicate in the format clause that only lines starting with that word are to be considered. The following preamble clause indicates this:

(Format 'Total ' xxxxxxxx)

The single quotes around Total are used in the format to delimit text that has to occur literally in the source. The quoted text is called a **literal** for that reason. As shown in this example, literals and special format characters can coexist. Additionally, one can be present without the other. Blanks count within literals; note the blank right after Total.

Only one blank separates Format from the literal. In data capture software, blanks are not ignored. There cannot be more than one blank separating Format from the formatting characters. Trailing blanks are also not permitted.

The appropriate preamble for this example is shown in Figure 2-5:

```
(Character 1)(Format 'Total 'xxxxxxx)(Type text).  
(Character 25)(Format bbb,bb9.99)(Type amount).  
(End of preamble).
```

Figure 2-5 Preamble

---

## More about literals

---

A literal might also consist solely of blanks. By specifying (Format ' 'xxxxxxxxxxx) for the Employee Name column in Figure 2-6, the data capture software is instructed to capture lines that start with a blank.

Field Staff	Page 1
Smith, A.	Marketing
Jones, E.	Administration
O'Brien, W.	Sales
Liou, D.	Sales
Smith, D.	Sales

Figure 2-6 Source document

## Using literals to indicate date formats

Literals are also used to indicate the punctuation involved in date formats. The special format characters are M (for month), D (for day), and Y (for year).

Here are some examples of source file layouts for February 4, 1978, and the corresponding Format clause:

02/04/78	(Format MM '/' DD '/' YY)
02-04-78	(Format MM '-' DD '-' YY)
780204	(Format YYMMDD)
04 02 78	(Format dd ' ' MM ' ' YY)
2 4 78	(Format MM ' ' DD ' ' YY)
04/02/78	(Format DD '/' MM '/' YY)
4/02/78	(Format DD '/' MM '/' YY)
1978.02.04	(Format YYYY '.' MM '.' DD)
Report of 4/ 2/78	(Format 'Report of ' DD '/' MM '/' YY)

Only numerical forms of dates with fixed length are supported. If you omit the format clause for a date column, it is assumed you mean (Format MM '/' DD '/' YY).

## Specifying uppercase or lowercase

---

Since source text copied from a terminal emulation window is sometimes in uppercase letters, data capture provides a way for the letters to be changed when the table is created.

The case property is used to specify either uppercase or lowercase letters. In the value for this property, U stands for uppercase and L stands for lowercase. So (Case L), for example, changes all the text in the affected column to lowercase.

You may also want to specify the case separately for the first character of a word or the first character of the whole field. This is done by two pairs of letters (separated by a space), as follows:

- (Case UL UL) First character of each word uppercase, the rest lowercase
- (Case ULLL) First character of the field uppercase, the rest lowercase

If a case clause is not specified, the case of letters is unchanged from the source text.

Another part of text appearance consists of the font and typeface (bold, italics, underlined). With data capture, all text in the table produced has the same font and typeface, but there is a way to determine what that is. This is done by attaching the desired character properties to the first visible character (not a blank or a tab) of the preamble in the source document. Do this by pressing <FONT>, <BOLD>, <UNDERLINE>, and so forth. These character properties are then applied by data capture to the text it puts in the table.

## Capturing multi-line rows, vertical layout

Some computer-generated reports produce so much information per row that it does not fit on a single text line, as shown in Figure 2-7.

Field Staff			Page 1
1234567890123456789012345678901234567890	1234567890	1234567890	1234567890
Smith, A.	Marketing	St. Louis	
	12,122	1,220	
Jones, E.	Admin.	Chicago	
	10,123	240	
O'Brien, W.	Sales	Chicago	
	32,162	3,640	
Liou, D.	Sales	Detroit	
	16,164	1,326	
Smith, D.	Sales	New Orleans	
	13,322	1,634	

Figure 2-7 Source document with multi-line rows

The numerical fields here occur on the second line of each employee's information record. This type of arrangement is accommodated by a preamble property called `line`. Since they are on the second line, the two numerical fields would be specified as follows:

```
(Type Amount)(Format bb,bb9)(Character 15)(Line 2).
(Type Amount)(Format bb,bb9)(Character 23)(Line 2).
```

The same `line` property can be used for source documents arranged vertically rather than horizontally. In Figure 2-8, for example, the literal "Empl" can be used to make sure that data capture picks up the right data (so it does not think "Marketing" is an employee).

```
(Format 'Empl:   ' xxxxxxxxxxxxxxxxxxxxxxxxx).  
(Line 2)(Length 50)(Character 8).  
(Line 3)(Length 50)(Character 8).  
(End of Preamble).
```

```
Empl:   Smith, A.  
Dept:   Marketing  
Title:  Group Manager
```

```
Empl:   Jones, E.  
Dept:   Administration  
Title:  Bookkeeper
```

```
Empl:   O'Brien, W.  
Dept:   Sales  
Title:  Customer Representative
```

```
Empl:   Liou, D.  
Dept:   Sales  
Title:  Account Manager
```

```
Empl:   Smith, D.  
Dept:   Sales  
Title:  Customer Representative
```

Figure 2-8 Source document

---

## Indicating absent fields in the source text

---

When data is copied from a terminal emulation window into a source document, fields are sometimes left blank. This often happens when fields contain zeros; the field is left blank in the source text rather than cluttering the page with zeros.

If there is no command put in the preamble to indicate that fields may be absent, the entire row will be ignored by the data capture software. This is because data capture performs a pattern match, and any mismatch is cause for ignoring a line of text.

The **required** property is used to avoid this problem. It is used to indicate whether or not a particular field is sometimes absent from the source text. It is one of the standard field and column properties. (**Required no**) means that the field is sometimes absent; a mismatch in that source text position causes an empty spot in the table, but the rest of the row still appears if matched. (**Required yes**) means that a mismatch in that source text position is a mismatch for the entire row. If you omit this property, it is assumed you mean (**Required yes**). At least one column must be required.

---

## **Capturing subrows and divided columns**

---

Sometimes the data in the source text may be formatted in rows with a number of subrows, such as a report of purchase orders in which each purchase order has a varying number of line items. Or the information may be grouped in divided columns, such as a record of managers in which each manager has completed a varying number of management training courses at different times. In data capture, these examples are called repeating groups.

Figure 2-9 shows a report in which employees have different numbers of degrees, and some have none at all.

Smith, A.	Marketing	St. Louis	B.A.	1980
Jones, E.	Admin.	Chicago		
O'Brien, W.	Sales	Chicago	A.A.	1977
			B.S.	1980
Liou, D.	Research	Detroit	B.A.	1955
			M.S.	1957
			Ph.D.	1962
Smith, D.	Sales	Chicago	B.A.	1964
			M.B.A.	1976

Figure 2-9 Repeating groups

Figure 2-10 shows a table one might want to have as the result of data capture. Note that only the employee name and degrees are of interest.

Employee Name	Degrees	
	Degree Level	Degree Year
Smith, A	B.A.	1980
Jones, E.		
O'Brien W.	A.A	1977
	B.S.	1980
Liou, D.	B.A.	1955
	M.S.	1957
	Ph.D.	1962
Smith, D.	B.A.	1964
	M.B.A.	1976

Figure 2-10 Table with subcolumns

The table resulting from the source document in Figure 2-9 contains one major column: Degrees. It also contains two subcolumns: Degree Level and Degree Year.

The preamble for a table with subcolumns must have an additional column type indicating this hierarchical relationship. The column type is called **group** (rather than text, amount, or date).

The preamble for this example is:

```
(Name Employee name)(Character 1)(Format ' '
xxxxxxxxx).
(Name Degrees)(Type group).
(Name Degree level)(Character 35)(Length 6).
(Name Degree year) (Character 42)(Type Amount)
(Format 9999).
(End of group).
(End of Preamble).
```

In the preceding preamble, the subcolumns are described immediately after the corresponding group column. Next, the last subcolumn is followed by the special delimiter sentence (End of group). This separates the subcolumns from further top-level columns. (In the example, there are no further top-level columns, but in other instances there could be a need for them: For example, one might want a righthand column for Location in the table produced.)

Notice that indentation was used to make the hierarchical structure readily apparent to the reader. This is an outline form of writing. Indentation is recommended as an aid to readability, but it is not required. It is ignored by the data capture software.

### Capturing combinations of subcolumns

---

A subcolumn is allowed to have further subcolumns. In a more comprehensive report, for example, each employee's degree might be followed by a listing of major fields of study (see Figure 2-11).

1234567890123456789012345678901234567	Smith, A.	Marketing	St. Louis	B.A.	1980
				Business	
	Admin.				
	Jones, E.	Admin.	Chicago		
	O'Brien, W.	Sales	Chicago	A.A.	1977
				English	
				B.S.	1980
				Economics	
				Philosophy	

Figure 2-11 Combinations of subcolumns

In such a situation, there would be a group column among the subcolumns of Degrees, in addition to the two subcolumns already there. Its subcolumns would constitute a second level of indentation.

The preamble for that example would look like this:

```

(Name Employee name)(Character 2)(Length 11).
(Name Degrees)(Type group).
  (Name Degree level)(Character 36)(Length
6).
  (Name Degree year) (Character 43)(Type
Amount)(Format 9999).
  (Name Courses)(Type group).
    (Character 36)(Format ' '
xxxxxxxxxxxxxxxxxxxxxx).
    (End of group).
  (End of Preamble).

```

**Figure 2-12 Preamble for combinations of subcolumns**

In this case Courses has only one subcolumn. It is always acceptable to have one subcolumn.

There may be several repeating groups at the same level of the hierarchy. For example, each employee might also have a list of current project assignments. This list would be unrelated to the Degrees column. Projects would be a group column at the same level of indentation as Degrees, and its subcolumns (perhaps Project Name and Completion Date) would be like Degree Level and Degree Year. Refer to the example in Figure 2-13.

```

(Name Employee name)(Character 2)(Length 11).
(Name Degrees)(Type group).
    (Name Degree level)(Character 36)(Length 6).
    (Name Degree year) (Character 43)(Type Amount)
(Format 9999).
    (End of group).
(Name Projects)(Type group).
    (Name Project name)(Character 15)(Length 16).
    (Name Completion date)(Character 32)
(Type Date)(Format MM '-' DD '-' YY).
    (End of group).
(End of Preamble).
1234567890123456789012345678901234567890123456789
01234567890
Smith, A.   Marketing St. Louis   B.A.   1980
           Competitive Anal 02-01-84
           Sirius Launch     12-11-83

Jones, E.   Admin.      Chicago
           Equipment Audit 01-01-84
           Apexx Proposal  03-11-84
           '84 Oper'g Plan 01-01-84

O'Brien, W. Sales      Chicago      A.A.   1977
           B.S.   1980
           Apexx Proposal  03-11-83
    
```

Figure 2-13 Source document with subcolumns

## Ignoring indentation in the source text

Sometimes text in the source document uses indentation to show a grouping, as shown in Figure 2-14.

Smith, A.	Marketing	St. Louis
Competitive Anal	02 01	84
Sirius Launch	12 11	83
Levin, E.	Admin.	Chicago
Equipment Audit	01 01	84
Exxon Proposal	03 11	84
'84 Oper'g Plan	01 01	84
O'Brien, W.	Sales	Chicago
Exxon Proposal	03 11	83
Liou, D.	Research	Detroit
Gamma-Ray Study	12 01	84

Figure 2-14 Grouping by indentation

This may present a problem in that the data capture software might mistake indented information for an employee name or a location. The obvious preamble would be ambiguous, in that Competitive can be mistaken for an employee name, for example. Moreover, Anal 02 can be a department and 01 84 can be a location. If this happened, you would notice the problem when you checked the table that data capture software produced.

A way to circumvent the problem is to specify in the preamble that the first character of the employee name cannot be a blank. There is a special format character to accomplish this: Y matches any character except a blank. It joins the previously defined special format characters for text fields: A, X, and 9. In the example, the employee name column would be (Format yxxxxxxxxxxxx).

## Subtotals, fields "after" a group

Using the source text shown in Figure 2-15 you could not create a table like the one in Figure 2-16.

Region: Eastern		
EMPLOYEE	HIRED	YTD EARNINGS
Smith, A.	02-11-68	12,778.23
Jones, E.	07-12-82	12,622.11
O'Brien, W.	02-13-68	23,718.23
Total Eastern		49,118.57
Region: Central		
EMPLOYEE	HIRED	YTD EARNINGS
Liou, D.	12-11-78	26,737.00
Smith, D.	12-21-82	0.00
Total Central		26,737.00
Region: Western		
EMPLOYEE	HIRED	YTD EARNINGS
Leary, T.	02-13-68	23,718.42
Fredholm, R.	12-11-78	22,737.00
Abel, C.	12-11-82	812.00
Total Western		47,267.42

Figure 2-15 Source text

Eastern	49,118.57	Smith, A.	02-11-68	12,778.23
		Jones, E.	07-12-82	12,622.11
		O'Brien, W.	02-13-68	23,718.23
Central	26,737.00	Liou, D.	12-11-78	26,737.00
		Smith, D.	12-21-82	0.00
Western	47,267.42	Leary, T.	02-13-68	23,718.42
		Fredholm, R.	12-11-78	22,737.00
		Abel, C.	12-11-82	812.00

Figure 2-16 Table

Because the Region total YTD Salary entry occurs after the enumeration of employees in the region, another value must be assigned to the preamble to indicate this situation. The property `after` is used in this situation. The value to assign to `after` is the name of the group column after which the subtotal column appears in the source text. (This has nothing to do with the order of the columns in the table produced; `after` describes only the source text.)

In the example, a user would indicate (After Employee List) for the Region Total YTD Salary. This assumes that Employee List is the name assigned to the group column: (Name Employee List). This is a new usage for the name property. It lets one column's preamble sentence refer to another column.

A preamble for the example is shown in Figure 2-17.

(Format ' Region: ' xxxxxxxxxxxxxxxxx).  
(Type amount)(Format bbb,bb9.99)(Character 25)(After Employee list).  
(Type group)(Name Employee list).  
    (Length 13).  
    (Type date)(Format MM '-' DD '-' YY)(Character 15).  
    (Type amount)(Format bbb,bb9.99)(Character 25).  
    (End of Group).  
(End of Preamble).

Figure 2-17 Preamble using "after" property

The preamble works as follows:

- (Format ' Region: ' xxxxxxxxxxxxxxxxx).  
This sentence uses a format statement that specifies Region with one space in front of it and a colon and several spaces after it. Otherwise, the preamble ignores the line.
- (Type amount)(Format bbb,bb9.99)(Character 25)(After Employee list).  
The first clause specifies that the data is numerical. The second specifies that the format allows up to 999,999.99, that the data starts at character 25, and that it follows a group with the name Employee list. Data that occurs before the group Employee list is ignored.
- (Type group)(Name Employee list).  
This line titles a column Employee List and sets up a subgroup.
- (Length 13).  
This line specifies that the employee name is 13 characters long.

- (Type date)(Format MM '-' DD '-' YY)(Character 15).  
This sentence sets up a subcolumn within the group with a format for a date, and indicates that the date starts at character 15.
- (Type amount)(Format bbb,bb9.99)(Character 25).  
This sentence sets up a subcolumn within the group, specifies that it is an amount, and indicates that it begins at character 25.
- (End of Group).  
This sentence ends the subgroup.
- (End of Preamble).  
This line ends the preamble.

---

## Defining column order

---

Although all of the examples given in this chapter have produced columns in the same order as in the source text, this is not mandatory. Columns may come out in any order.

The column order in the table produced is the same as the order of the sentences in the preamble. You may use any order, as long as the subcolumns of a group column are together. Group columns do not have to be at the end. A table can be produced with the group column between two other columns simply by reordering the sentences as follows:

```
(Format ' Region: ' xxxxxxxxxxxxxxxxx).
(Type group)(Name Employee list).
  (Length 13).
  (Type date)(Format MM '-' DD '-' YY)(Character
15).
  (Type amount)(Format bbb,bb9.99)(Character 25).
  (End of Group).
(Type amount)(Format bbb,bb9.99)(Character 25)
(After Employee list).
(End of Preamble).
```

---

## Content of source text

---

Any text formatting properties, special characters, frames, or other non-text objects that may occur in the source text are ignored by [COPY TEXT TO TABLE]. The data is operated on by counting characters as though they were in a fixed pitch font. A tab character in the source text is considered to be a single non-printing character; if tabs occur in the source text, the user must be aware that white space can result from either spaces or tabs, and that the two behave differently.

Data capture considers that a source text "line" is delimited by a line-break (or paragraph-break) character. If there are too many characters between line breaks, the workstation divides the text into several lines when displaying or printing it, but [COPY TEXT TO TABLE] still considers it a single line. This is helpful. It means you do not have to make the source document page wide enough (or the text small enough) to eliminate line wraparound. Source text can be taken from terminal emulation without reformatting it first.

In matching source text against the desired pattern, data capture may be called on to access text beyond the end of a line, or beyond the end of the file; all such characters are considered to be blanks.

The matching of source text against format by data capture is stricter than that which is done when you type into fields; the text must match the format actually specified, not merely be valid for the particular column type. This is because of the difference of purpose. With type-in, the intention is to tolerate user inconsistencies; data capture, on the other hand, must discriminate between valid data and "noise" text in the source.

## All main columns are "after" a group

Figure 2-18 is an example of a source document in which the main columns occur after a group. Notice that the subtotals for each region occur after the employee list for that region.

EMPLOYEE	HIRED	YTD EARNINGS
Smith, A.	02-11-68	12,778.23
Jones, E.	07-12-82	12,622.11
O'Brien, W.	02-13-68	23,718.23
Total Eastern		49,118.57
EMPLOYEE	HIRED	YTD EARNINGS
Liou, D.	12-11-78	26,737.00
Smith, D.	12-21-82	0.00
Total Central		26,737.00
EMPLOYEE	HIRED	YTD EARNINGS
Leary, T.	02-13-68	23,718.42
Fredholm, R.	12-11-78	22,737.00
Abel, C.	12-11-82	812.00
Total Western		47,267.42

Figure 2-18 Source document with main columns after a group

In this unusual case, there is nothing on line 1. This causes trouble for the software, since it does not know where to start a new region. If a situation like this is encountered, deal with it by using an artificial table column. Specifically, include a sentence in the preamble to identify the heading at the beginning of each region's list, using a literal format. Here is a preamble with such an artificial sentence at the beginning:

(Format 'EMPLOYEE HIRED YTD EARNINGS').  
(Type amount)(Format bbb,bb9.99)(Character 25)  
(After Employee list).  
(Type group)(Name Employee list).  
(Length 13).  
(Type date)(Format MM '-' DD '-' YY)(Character 15).  
(Type amount)(Format bbb,bb9.99)(Character 25).  
(End of Group).  
(End of Preamble).

Note that, as always, you need the right number of blanks in a literal. The artificial sentence does not cause a column to appear in the table. Since the format consists of literal characters only, there would be no data to put into a column.

---

## Limitations of data capture

---

If all the advice offered so far leaves you still unable to capture the sample of source text you want, it's possible that you have encountered a limitation of data capture. Some known limitations are as follows:

- **Delimiters.** Data capture extracts text from the source based on counting characters in lines. It assumes that a given field occurs in the same character position each time. If your sample has fields that are delimited by special punctuation characters instead, the sample cannot be accommodated unchanged.

It may help to alter the sample in advance by substituting new-line characters for the delimiters using <FIND>. For example,

```
Abbot, A;Accountant;24411;  
Johnson, C;Management Trainee;66110;  
Song, S;Consultant;01234;
```

becomes:

```
Abbot, A  
Accountant  
24411
```

```
Johnson, C  
Management Trainee  
66110
```

```
Song, S  
Consultant  
01234
```

- **Multiple line fields.** The position of the text in the source to be captured is designated by its position within a designated line. The software does not accommodate source text layouts in which the content of a single field spills over from one line to the next. The limitation applies only if there is an explicit new line character. Ordinary text wraparound at the right margin is all right, since data capture considers only explicit line breaks.
- **"Noise" lines within a row.** Noise characters are present in computer-generated reports to help the user perceive the organization of the data, but they do not themselves comprise data. Examples are page numbering and column headers. In almost all cases, data capture successfully filters this material out. If the noise appears between rows, it is successfully ignored; it does not appear in the table and does not interfere with recognition of valid data. One case not accommodated, though, is that of a row extending to multiple lines, and a noise line intervening between the lines of the single row. In this situation, the best thing to do is remove the offending noise text before starting the [COPY TEXT TO TABLE] operation.

*(This page intentionally blank)*

### **3. Property/option sheets and windows**

---

There are no property sheets, option sheets, or windows associated with VP Data Capture.

***(This page intentionally blank)***

## 4. Error messages

---

There are two kinds of problems that can occur data capture:

- The preamble may be unacceptable to the data capture software. In this case, no table is generated and an error message is posted in the message window.
- The preamble may be acceptable, but the resulting table may contain unexpected data or formatting. Depending on the problem, the preamble can be changed, or, if the problem is a minor one, the resulting table can be edited.

The following paragraphs contain some hints for debugging preambles.

---

### Unacceptable Preambles

---

If the preamble is unacceptable, a message is posted. The message disappears as soon as a key or the mouse is touched. Thus, it is best to write down the error message. Error messages for unacceptable are as follows:

**Error Message:** Preamble error: Preamble not terminated at...*(formatting characters in the preamble are listed)*

**What to Do:** Check that the formatting characters are valid for the type of column. There must be only one blank separating the word Format from the formatting characters. There should not be any other blank spaces, except in amount columns of the languages

in which a blank is a thousands separator (German and Portuguese).

**Error Message:** Preamble error: Preamble not terminated at...

**What to Do:** The preamble is missing the End of Preamble sentence. Check the preamble and add the missing sentence.

**Error Message:** Preamble error: missing left parenthesis at... (*the last property in the preamble, followed by the next character in the document*)

**What to Do:** If the End of Preamble sentence is missing, the next character after the last property is treated as part of the preamble. If this character is not a left parenthesis, the error message is posted. Check the preamble and add the missing sentence.

**Error Message:** Preamble error: Duplicate attribute at: t bbbb format.

**What to Do:** There can only be one of each property per sentence. Therefore, a column cannot have two format properties. Check each sentence and make sure only one format property exists per sentence.

**Error Message:** Preamble error, unknown sibling column at... (*the name of the unknown sibling column*)

**What to Do:** If an "after" clause contains a column name that does not exist, this error message appears. Misspelled column names are the most common cause of this error. Check that the column names exist for all "after" clauses and that they are spelled correctly in the preamble.

**Error Message:** Preamble error: there is a property inapplicable to a group at... (*the inapplicable property*)

**What to Do:** For group columns, properties such as format, length, type, and character have no meaning. Each subcolumn has its own values for these properties. Check the group column and delete the inapplicable property.

**Error Message:** Preamble error: at least one column must be required.

**What to Do:** At least one column must be designated (Required yes). Since the default is (Required yes), make sure that all of the columns are not designated (Required no).

**Error Message:** Preamble error: two fields overlap at a common line/column at 1/15.

**What to Do:** A single position should not be in two sibling columns. For example, in the following preamble, the wide column is 10 characters long and overlaps the next column:  
(name wide column)(character 10)  
(format xxxxxxxxxxxx).  
(name next column)(character 15)  
(format yxx).

---

## Acceptable preambles

---

One common problem with data capture is that the resulting table is blank. When this problem occurs, it is because no match has been found. For a match to occur, the source text must match the preamble at the exact character position. If a field is required and does not match, the row is not copied to the table. To correct this problem, you it is necessary to find one or more columns that have caused the mismatch.

To begin, insert (Required no) for all columns except one. Choose the column which is easiest to match to be (Required yes.). For example, text columns with (format xxxxxx) are easy to match. Then select [COPY TEXT TO TABLE]. Study the resulting table, to tell which columns have caused the mismatch.

Among the most probable candidates are the amount columns, which have the decimal point and the thousand separators in certain positions. If possible, change these formats to ones that are easier to match.

Once the columns that caused the mismatch are corrected, remove the (Required no) inserts from the columns that are required. Select [COPY TEXT TO TABLE] again and check the resulting table.

# Appendix A: Properties

---

The preamble contains one sentence (terminated by a period) for each table column. Each sentence is made up of visible document characters. Each preamble sentence is made up of a set of property-value pairs. Each pair corresponds with one property of the corresponding column. It is represented as a pair of parentheses containing a key word (choosing a property) and then a value for that property. An example of a pair is: (Type amount). An example of a sentence is: (Type amount)(Line 2)(Format BBB.99). Line breaks are ignored, as are spaces between pairs. This flexibility should be exploited by laying out preambles for easy reading; indentation should be used for nesting of repeated groups.

Key words are in English only; uppercase or lowercase letters in key words are treated the same. In the example of the type property, the value is a selection among other key words (amount, date, text, and group). For other properties, the values are more diverse, as will be explained in the following paragraphs.

Format is the most important property. It represents the length of the source text segment to be matched, along with the matching pattern. Matching determines when text fragments are accepted as field values by the [COPY TEXT TO TABLE] operation. The format property contains literals and special format characters (as in the format property of fields). It is necessary to include extensive literals, since that is one way the [COPY TEXT TO TABLE] option has of recognizing valid source text.

## Description of properties

---

Each property has a specific configuration of values it expects. Some have default values, which they assume in sentences in which the values are not explicitly mentioned. The examples in Chapters 1 and 2 of this manual should be consulted as the properties are studied.

The major properties are as follows:

- **Property:** Type

**Value:** Text, date, amount, group

**Default:** Text

**Meaning:** Is the same as the type parameter of fields, columns, and record files. Type specifies the data type of the field (for example text or amount). A group column has a set of subcolumns. The sentences describing the subcolumns must immediately follow the sentence describing the group column. After the last subcolumn sentence, the special (End of group). sentence must appear.

- **Property:** Name

**Value:** A string of text characters

**Default:** A name contrived by the system to be unique (for example, Column23)

**Meaning:** Contains a user-assigned name for this column. The name abides by the same conventions as fields, but may not contain the right parenthesis; case is significant. Leading/trailing blanks are not significant, but embedded blanks are. The name is also placed in

the column headings of the table produced.

- **Property:** Case
  - Value:** One of several specific combinations of S, U, and L
  - Default:** S
  - Meaning:** Causes the case of letters be changed as text fields are stored in the table. The alternatives are:
    - S Same case as in source text
    - U All uppercase
    - L All lowercase
    - UL First character of each word uppercase, the rest lowercase
    - UL UL First character of each word uppercase, the rest lowercase
    - UL LL First character of each field uppercase, the rest lowercase
- **Property:** Character
  - Value:** An unpunctuated positive integer (leading/trailing blanks ignored)
  - Default:** 1
  - Meaning:** Designates the character position at which this field starts. Numbering starts at the left margin with 1. This property is not applicable to a group column, but it is applicable to its subcolumns.

- **Property:** Line
  - Value:** Contains an unpunctuated positive integer (leading/trailing blanks ignored)
  - Default:** 1
  - Meaning:** Designates the line number in the source text containing this field. Numbering starts with 1 and is relative to the sibling fields in the same row (or subrow). If all siblings fit on the same line, they all have 1 for this property. In any case, at least one of the sibling columns must specify line 1.
  
- **Property:** After
  - Value:** The name of a sibling group column
  - Default:** Not after another column
  - Meaning:** Accommodates fields of a record that appear in the source text after a group. Subtotals often occur this way in computer-generated reports. It is invalid to specify line and after in the same sentence.
  
- **Property:** Required
  - Value:** Yes or No
  - Default:** Yes
  - Meaning:** Specifies that when [COPY TEXT TO TABLE] is selected, a row will be created only if every required column is matched. All blanks are not considered a match. If a column is not required, a mismatch simply means a value is not stored for that particular column. Among siblings or subcolumns, at least one must be both required and not appear after another

---

column. Not applicable to a group column (but applicable to its subcolumns).

- **Property:** Comment
  - Value:** Arbitrary text, not including a left or right parenthesis character
  - Default:** Not applicable
  - Meaning:** Contains an explanatory message for readers of the preamble. Ignored by data capture software.
  
- **Property:** Length
  - Value:** An unpunctuated positive integer (leading/trailing banks ignored)
  - Default:** Taken from the format property
  - Meaning:** Specifies the number of characters in the source text to be extracted for the current field. This can be used as a terse substitute for the Format property when the text is featureless. For example, (Length 8) is equivalent to (Format xxxxxxxx) for a text column or (Format-bbbbbbb) for an amount column. When the source text is not featureless, it is best to use the format property instead of length; the program could run faster and is more likely to include exactly the intended information.
  
- **Property:** Format
  - Value:** A combination of literals and special format characters
  - Default:** Depends on the type
  - Meaning:** Specifies that when [COPY TEXT TO TABLE] is selected, format characters

are matched one for one against the source text indicated by the character/line/after properties. Contiguous literal characters are bracketed in the value by single quotes. Literals are not stored in the destination table. Tabs are significant in literals but not elsewhere. Special format characters (analogous to the box characters in the field format property) are outside the quotes; they may be written in either uppercase or lowercase letters. Valid formats depend on the type property. There is no format for a group column (but there are formats for its subcolumns).

**Text format:** The special formatting characters A, 9, and X are already defined for US English fields (refer to the "Special formatting Characters" subsection that follows). There is a new one, Y, which matches any character except a blank. This is indispensable for detecting indentation in the source text; indentation is often used to set off subtotals or other distinguished data. A text format could have literal characters only – no special format characters. This is useful in matching source text that does not have adjacent variable data. (For example, a subtotal line is often set off by preceding it with a line of blanks and dashes). For such fields, selecting [COPY TEXT TO TABLE] does not create a column in the destination table. The default text format is all X's; the number of X's is taken from the length property.

- **Property:** Language
- Value:** One of thirteen languages
- Default:** The default language of the workstation
- Meaning** Determines the default date format and the thousand and decimal separators. U.S. English can be spelled as either US English or USEnglish. U.K. English can be spelled as either UK English or UKEnglish.

---

## Special formatting characters

---

Special formatting characters are as follows:

- A The corresponding field position can contain a letter or a blank.
- 9 The corresponding field position can contain a digit or a blank.
- X The corresponding field position can contain any character.
- Y The corresponding field position can contain any character except a blank.

---

## Amount format

---

The following special formatting characters are already defined: 9 B \* + - , . (the Z formatting character is excluded; it is for left-justified punctuated amounts, a source text layout not supported by data capture software).

There are two enhanced ways to describe negative numbers. First, the minus sign can be trailing (instead of leading). Second, enclosing parentheses can be

used; this introduces two new special formatting characters: ( ).

A maximum of 15 significant digits is supported. Default amount format is all B's, with a leading minus sign (the total number of characters is taken from the length property).

Special formatting characters for amount fields are as follows:

- 9 The corresponding field position can contain a digit.
  - b The position contains a digit unless the digit would be an insignificant zero, in which case the position contains a blank. (An insignificant zero is a leading zero to the left of the decimal point, or a trailing zero to the right of the decimal point.) The position may also contain a comma or decimal point; thus, B is useful when the source text has a numerical column that is not decimal-aligned.
  - \* The position contains a digit unless the digit would be a leading zero, in which case the position contains an asterisk.
  - + A plus sign or minus sign appears in the source text, depending on the sign of the field value. This character can only appear as the first character of the format.
  - A minus sign (hyphen on the standard keyboard) appears only if the field value is less than zero. This character can only appear as the first character of the format or the last character of the format.
  - ( ) Parentheses (surrounding the digits and punctuation) are an alternative way to express negative amounts. The negative quantity is enclosed in parentheses.
  - ,
- The corresponding field position contains a comma. If the special format character to the left of the comma is B or \*, and the field value

is such that a digit would not be displayed there, then the comma is treated as B or \*, respectively. (After the decimal point, read "right" in this rule instead of "left.")

The corresponding field position contains a decimal point.

## Date format

---

Only numeric, fixed-length dates are supported. There are three special formatting characters: M, D, and Y (for month, day, and year, respectively). M must appear twice, as contiguous characters. So must D. Y may appear either twice or four times. The separators between the date components are specified using literals. The default date format is MM '/' DD '/' YY. Additional date formats are as follows:

DD '/' MM '/' YY

YYYY '-' MM '-' YY

YY'' MM'' DD

YYYY '.' MM '.' DD

'Total as of the ' MM '/' DD '/' YY '  
inventory'

*(This page intentionally blank)*