FLAGSTAFF ENGINEERING

IBM OS/6 Word Processing Conversion Programs User's Manual

FLAGSTAFF ENGINEERING
P.O. Box 1970
Flagstaff, Arizona 86002
(602) 779-3341
Document Number 8860-OS6

FLAGSTAFF ENGINEERING

IBM OS/6

WORD PROCESSING CONVERSION PROGRAMS

USER'S MANUAL

FLAGSTAFF ENGINEERING
Box 1970
Flagstaff, AZ 86002
(602) 779-3341
TELEX 705609

Document Number 8860-OS6

TABLE OF CONTENTS

INTRODUCTION

This manual provides program documentation for Word Connection OS/6 document conversion programs. These programs are designed to transfer documents from 8" OS/6 diskettes to IBM PC/XT/AT operating under IBM DOS versions 2.X OR 3.x.

IBM OS/6 document files are transformed to a format known as Revisable Form Text. This format is part of an IBM standard called Document Content Architecture. The RF format provides a standard method of specifying tabs, margins, control codes, etc.

By using the RF file standard, Word Connection programs are capable of exchanging documents between different word processing systems without losing important document information.

SYSTEM REQUIREMENTS

Word Connection programs require the following minimum system configuration:

1. IBM PC, PC/XT, or PC/AT computer.

2. Flagstaff Engineering Diskette Connection 8" diskette drive system.

3. Flagstaff Engineering "CONNECTION" I/O device driver. (See the System Installation Guide and Diagnostic and Utility Manual for more information.)

PROGRAM NAMING CONVENTIONS

Word Connection program names use specific abbreviations to indicate function and system type.

System abbreviations are:

ASC	ASCII text file.
OS6	IBM OS/6 8" diskette file.
RF	IBM PC-DOS Revisable Form Text file.

Function abbreviations are:

XXXTOYYY Transfer files from system format XXX to system format YYY.

PROGRAM USE

To load Word Connection programs, use the standard PC-DOS load command syntax:

```
A:PROGRAM NAME
```

In the program operation sections of the manual, keyboard entries are indicated by bracketed characters.

[ENTER] equals the enter key.
[Y] equals the Y key.

Word Connection programs may be aborted at any time by pressing [CTRL] [C]. Occasionally, the Control-C sequence may not successfully terminate the program. If this occurs, press [CTRL] [BREAK] to end the program.

When Word Connection programs are loaded, a program message is displayed that contains the copyright date. New program versions are released periodically, and this date is used to keep track of program updates.

Program prompts require key entries followed by the [ENTER] key. If the [ENTER] key is pressed without typing a character, the program will default to the first character in the displayed message options. Message options are displayed as:

(Default/Option1/Option2/Option3)

For example, if the [ENTER] key is pressed after the program prompt, 'ENTER DRIVE SELECTION (1/2)', the program will default to the first 8" diskette drive.

Word Connection program documentation is arranged in the following format:

1. Program Name.
2. Use.
3. Miscellaneous Notes.
4. Description.
5. Operation.
6. Defaults.
7. Destructive/Nondestructive
8. Sample Program Run

Word Connection programs may be copied to and executed from any DOS diskette or fixed disk volume desired.

CONVERSION TIPS

OS/6 conversion programs are designed to transfer documents from 8" OS/6 diskettes to PC-DOS revisable form text files, and other IBM PC programs that can access either ASCII or revisable form text files.

To transfer files from an 8" OS/6 diskette to the IBM PC, the following considerations should be kept in mind:

DOS files created by OS/6TORF are revisable form text files written in EBCDIC. Revisable form text files are supported by may word processing packages such as IBM's DisplayWrite 2/3, Microsoft's WORD, and Multimate.

DisplayWrite 2/3 provides an internal function to create DisplayWrite 2 text files from any revisable form text file. The text file will probably need to be repaginated with DisplayWrite 2/3.

Revisable form text files are used by other Flagstaff Engineering word processing conversion programs.

The programs, ASCTORF and RFTOASC, transform ASCII files to revisable form text files, and revisable form text files to ASCII files. These two programs provide a bridge between revisable form text files, and any other application program that can access an ASCII file such as Volkswriter. A Wordstar Non-document file is an ASCII text file.

An example of a transfer application would be:

1. Creation of a OS/6 document file on an IBM Office System/6 word processor system.

2. Transformation of the OS/6 document file to a DOS revisable form text file with OS6TORF.

3. Transformation of the revisable form text file into a DisplayWrite 2/3 text file by using the DisplayWrite 2/3 document functions.

4. Editing and printing the file with DisplayWrite 2/3 and then transforming the DisplayWrite 2/3 edited text file into a revisable form text file.

5. Transforming the revisable form text file into an ASCII file using RFTOASC.

6. Downloading the ASCII file into a laptop computer attached to the IBM PC.

Revisable form text files may be converted to DisplayWrite 2/3 text files by bringing up DisplayWrite 2/3, and then choosing document utilities from the main menu.

Revisable form text files may be converted to Microsoft WORD by running WORD conversion programs.

THIS PAGE INTENTIONALLY LEFT BLANK

OS6TORF

USE:

Copy an OS/6 document to a Revisable Form Text file on an IBM PC-DOS volume.

NOTES:

OS6TORF will properly transfer only OS/6 document files. The OS/6 system creates many different file types. The file types are indicated by the program with the following letters:

C - Communication file
D - Document
S - Segment
T - Segment plus OCL
F - File
O - Stored option list
2 - Segment2
3 - Segment2 plus OCL

When file types other than documents are encountered, OS6TORF will either indicate the file is not a document, or attempt to transfer the file.

The program displays a message indicating 'Unknown OS/6 Control Command' when an unsupported OS/6 control character is found. This message contains a data information line and should not be regarded as an error message since unsupported control characters are ignored by the conversion process.

DESCRIPTION:

The OS6TORF program is designed to copy a OS/6 document to a revisable form text file on any valid DOS volume. The rules for transforming the OS/6 document file to a revisable form text file use an IBM standard called Content Document Architecture (DCA).

The program will identify the 8" diskette and then display a directory of file names. Each file will be displayed on a file information line consisting of job ID letter, file type, and file name.

The program copies document files from the 8" diskette by job letter. A single document file may be copied or the entire diskette may be copied.

The program will copy a document to a user specified DOS file name and volume. A DOS file name must be specified when coping a single file. When the option is selected to copy all files on the diskette, the target file name on the DOS volume will consist of the first eight characters of the OS/6 directory name followed by an extension of '.RF'.

OPERATION:

Load OS6TORF. On completion of the program load, a program description message will be displayed. The program will then prompt the user to enter the DOS file name of a translation table to use while coping the files.

Press [ENTER] to select the default internal table, or enter the file name of a user defined translation table followed by [ENTER].

The program will prompt the user to insert an 8" OS/6 diskette into the drive and then enter the drive number in use.

Insert the 8" OS/6 diskette into the drive and press [2] to select drive 2, or press [1] or [ENTER] to select drive 1.

The program will identify the OS/6 volume name and display a directory of the files on the diskette with a file information line on each document found in the directory. The program will then prompt the user to enter the job letter for the file to be copied.

Enter a job letter from 'A' to 'FF' for the document to be copied, or enter a [99] to copy all documents on the diskette.

If a single file is selected to be copied, the program will prompt the user to enter the job page number to be copied from the document selected. Single pages of the document may be copied by entering the desired page number, or all pages of the document may be copied by pressing [ENTER].

The program will prompt the user to enter the DOS file name that the document file or page should be copied to.

Enter the target DOS volume and file name using standard DOS syntax:

VOLUME ID:FILE NAME.EXTENSION

The program will copy the file(s) selected to DOS revisable form text files. After each file is copied, the program will display a copy complete message, with the number of pages copied for each job. The program will then redisplay a directory of the 8" diskette. The program will then prompt the user to enter another job number to copy.

Select another document to copy by entering the appropriate job number or press [ENTER] to exit the copy operation.

The program will prompt the user to copy from another OS/6 diskette.

Enter a [Y] to copy from another diskette, or enter an [N] or press [ENTER] end the program.

DEFAULTS:

Drive Number:	[ENTER]= Drive 1.
Job Letter:	[ENTER]= Exit prompt.
Page Number:	[ENTER]= Copy all pages.
DOS File Name:	[ENTER]= OS/6 Job Name.
Copy Another Diskette:	[ENTER]= End program.

SAMPLE RUN:

```
A> [OS6TORF] [ENTER]

COPY IBM DOCUMENT TO
REVISABLE-FORM TEXT PROGRAM
COPYRIGHT FLAGSTAFF ENGINEERING XX/XX/XX

ENTER FILE NAME FOR TRANSLATE TABLE-
PRESS [illegible]?  [ENTER]

INSERT 8" OS/6 DISKETTE - ENTER
DRIVE (1/2) WHEN READY.? [ENTER]

DIRECTORY OF OS/6 JOBS ON TESTVOL

ID TYPE    NAME            ID TYPE    NAME
A  -D- TESTFILE            B  -D-  UPDATES
C  -D- NOTES               D  -D-  LETTER
.    .   .                 .    .    .
.    .   .                 .    .    .
EE -D- REPORT1             FF -D-  REPORT2

ENTER JOB LETTER (A-FF/99=ALL) FOR
COPY - PRESS ENTER IF NONE?  [99] [ENTER]

COPY OF DOCUMENT TESTFILE.RF IS COMPLETED
- 14 PAGES WERE COPIED
COPY OF DOCUMENT UPDATES.RF  IS COMPLETED
- 20 PAGES WERE COPIED
COPY OF DOCUMENT NOTES.RF    IS COMPLETED
- 02 PAGES WERE COPIED
COPY OF DOCUMENT LETTER.RF   IS COMPLETED
- 05 PAGES WERE COPIED
COPY OF DOCUMENT REPORT1.RF  IS COMPLETED
- 21 PAGES WERE COPIED
COPY OF DOCUMENT REPORT2.RF  IS COMPLETED
- 17 PAGES WERE COPIED
```

```
DIRECTORY OF OS/6 JOBS ON TESTVOL

ID TYPE    NAME             ID TYPE    NAME
A  -D- TESTFILE             B  -D-  UPDATES
C  -D- NOTES                D  -D-  LETTER
.   .  .                    .   .   .
.   .  .                    .   .   .
EE -D- REPORT1              FF -D-  REPORT2

ENTER JOB LETTER (A-FF/99=ALL) FOR
COPY - PRESS ENTER IF NONE?  [C] [ENTER]
ENTER JOB PAGE NUMBER (199/1-199) - PRESS
ENTER FOR ALL PAGES? [1] [ENTER]
ENTER DOS FILE NAME FOR COPY - PRESS ENTER
TO USE OS/6 JOB ID? [NOTEWORK.RF] [ENTER]

COPY OF DOCUMENT NOTES.RF    IS COMPLETED
- 01 PAGES WERE COPIED

DIRECTORY OF OS/6 JOBS ON TESTVOL

ID TYPE    NAME             ID TYPE    NAME
A  -D- TESTFILE             B  -D-  UPDATES
C  -D- NOTES                D  -D-  LETTER
.   .  .                    .   .   .
.   .  .                    .   .   .
EE -D- REPORT1              FF -D-  REPORT2

ENTER JOB LETTER (A-FF/99=ALL) FOR
COPY - PRESS ENTER IF NONE?  [C] [ENTER]

DO YOU WANT TO COPY FROM ANOTHER OS/6
DISKETTE (Y/N)?  [ENTER]

C>
```

ASCTORF.EXE

Convert ASCII Document to RFT Text File

The ASCTORF program converts PC-DOS ASCII text files into PC-DOS *revisable form text* (RFT) files, which can then be converted into formats used by many different word processors and computer systems. ("ASCII" documents consist of varying-length lines of text where each line is usually ended by a carriage return/line feed pair of characters.)

Since RFT documents use EBCDIC encoding, ASCTORF translates all text characters from ASCII to EBCDIC. An internal translation table is used by default, but the user can specify a custom translation table if any specific character conversions are required for the document.

ASCTORF handles the problem of preserving the format of the document--how text is laid out on the page--by allowing the user to specify an optional Code Generation File which the program will use to place the proper RFT control codes in the output file. If no Code Generation File is used, then ASCTORF will only recognize the following two formatting characters (all other control characters in the input file will be dropped from the output file):

- ASCII form-feed characters (decimal 12, hex 0C) will be interpreted as page breaks

- 2 ASCII carriage returns (decimal 13, hex 0D) in a row will be interpreted as a "hard" carriage return; single carriage returns will be treated as "soft" carriage returns.

If more extensive formatting than this needs to be retained in the RFT document, then you will need to make a Code Generation File to specify the formatting control codes. The use of these files is described below.

The distribution diskette includes a sample code generation file named **ASCTORF.COD**; this file can be used as a departure point and altered to suit a particular conversion need. (Make a copy of this file and modify the copy to preserve the original contents).

ASCTORF will accept wildcard-qualified filenames as input files (where the characters "?" and "*" signify "any character" and "any string of characters" respectively). Thus, a set of files with a common prefix or extension can be converted by giving one command to ASCTORF. For example, if you have a diskette with the following files on it that you want converted to RFT files:

```
    LETTER.DOC
    LETTER1.DOC
    LETTER2.DOC
    LETTER2A.DOC
    LETTER3.DOC
```

you can use the wildcard-qualified filename

```
    LETTER*.DOC
```

to convert all these files in one operation.

If you convert a single file only, you can explicitly name the target (output) drive specifier and file, extension and path names. However, if you use a wildcarded input file name, you can only give a drive specifier (e.g., B:) and path name for the target files: all output files will be placed on the given drive and in the given subdirectory.

Whether or not wildcard-qualified input file names are used, all output files will have the same name as the input file with the extension .RFT appended unless the output filename extension is given explicitly (this is only possible if a single file is being converted).

THE CODE GENERATION FILE

This optional file consists of statements, one per line, that direct ASCTORF to make certain substitutions and conversions from the input to output files. This file is a plain ASCII text file that can be created by any simple text editor or as a Wordstar "non-document" file.

The Code Generation File consists of two types of statements--*format setup* statements and *control translation* statements. Either or both types of statement can be included in the file. If both types of statement are used, the setup statements should precede the translation statements.

FORMAT SETUP STATEMENTS

These statements insert RFT formatting information into the output file; the values given in setup statements will apply to the entire document. The following is a complete list of setup statements. In the Code Generation File, the formatting keyword (such as "LPI") must be separated from its parameter ("n") by exactly one space.

KEY n	(n is the IBM keyboard designator number)
LPI n	(n is the # of vertical lines/inch)
CPI n	(n is the # of horizontal characters/inch)
TOP n	(n is the top margin in # of lines)
BOT n	(n is the bottom margin in # of lines)
LEF n	(n is the left margin in # of characters)
RIG n	(n is the right margin in # of characters from the left edge of page)
DEN n	(n is the # of 1/2 lines vertical spacing)
PGL n	(n is the page length in lines)
PGW n	(n is the page width in characters)

CONTROL TRANSLATION STATEMENTS

These statements cause certain strings of characters in the ASCII input file to be translated into RFT format control characters in the output file. This allows the creator of the ASCII file to use certain sequences of characters that will not appear in the output text, but will be transformed into universally-recognized RFT formatting codes. ASCTORF translation statements must be entered in fixed-field format, since the program expects to find certain parts of the statement in certain positions on the line. The format is:

```
rrrr string specifier
```

where "rrrr" is the 3- or 4-character RFT code specifier from the following table, starting at the first position of the line, and "string specifier" is one of the two forms of string specifiers described below , starting at position 6 of the line.

ASCTORF control translation statements are "backwards"--the translation is from right to left in the statement. That is, the string specified on the right is translated into the RFT code specifier on the left.

STRING SPECIFIERS FOR TRANSLATION STATEMENTS

1. ASCII Text String Specifier:

ASCII text is specified in the form

```
      A"any text"  (optional comments may follow)
```

The quotation marks are *delimiters* that mark the start and end of the text; they can be any character that doesn't appear within the text itself. There must be no spaces between the "A" and the delimiter. The first character after the "A" is taken as the opening delimiter, and this character must be found at the end of the text. Any comment text can follow so long as it is separated from the end delimiter by at least one space.

2. Hexadecimal String Specifier:

Strings of any hecadecimal bytes can be specified in the following form:

```
      X"xx xx ... xx"  (optional comments)
```

The quotation marks act as delimiters as above. There must be no spaces between the "X" and the opening delimiter. The hexadecimal bytes are given as pairs of hex digits (0-9, A-F), each pair separated by one space. By using hex string specifiers, non-displayable characters can be translated.

RFT CONTROL CODES

The following list gives all the RFT codes used by ASCTORF. The 3-or 4-character code name is coded in the code translation statement; the meaning of each is given below.

BUS	-begin underscore
EUS	-end underscore
BSUP	-begin superscript
ESUP	-end superscript
BSUB	-begin subscript
ESUB	-end subscript
BBLD	-begin bold
EBLD	-end bold
BK	-begin keep block
EK	-end keep block
BOS	-begin overstrike
EOS	-end overstrike
CRE	-carrier return
RCR	-"required" carrier return
IT	-indent tab
HT	-horizontal tab
PE	-page end
RPE	-"required" page end
ATF0	-align text field (resume normal formatting)
ATF1	-align text field (period alignment)
ATF2	-align text field (comma alignment)
ATF3	-align text field (center text)
ATF4	-align text field (end-side alignment)
ATF5	-align text field (colon alignment)
ATL0	-align text line (normal)
ATL1	-align text line (center line)
ATL2	-align text line (align right)

SOME PITFALLS IN DOING CONTROL CODE TRANSLATION

Depending on the source of the ASCII text file, it may have to be "massaged" before using ASCTORF to convert it. The main problem is that of handling *toggle* sequences; these are controls that are turned on at the start of a block of text and must be turned off at the end of the block. Printer controls such as bold and underscore are usually toggle controls. The problem is that some text processors, such as Wordstar, use the same character to mark both the start and end of such a block. If such a file is converted using ASCTORF without pre-processing these control codes, the wrong RFT codes will be written to the output file. (The "begin" codes would be placed wherever the ASCII codes were found, and no "end" codes would be written.) If this is the case with the program you're using to produce the ASCII files, then you may need to run the Flagstaff Engineering utility program FILETRAN to convert identical codes to a toggled sequence of unique turn-on and turn-off codes.

SAMPLE CODE GENERATION FILE

```
KEY 103          -select a standard keyboard
LPI 8            -set 8 vertical lines per inch
LEF 0            -left margin of zero characters
RIG 70           -right margin at 70 characters
PGL 88           -page will be 88 lines long
PGW 90            and 90 characters wide
BUS  A"{bus}"    -translate the following
EUS  A"{eus}"     markers in the
BSUP A"{bsup}"    ASCII text into
ESUP A"{esup}"    RFT control codes:
BK   A"{bk}"
EK   A"{ek}"
IT   A"{it}"
HT   A"{ht}"
```

RUNNING ASCTORF

To run ASCTORF, type its name at the DOS command prompt. After displaying an introductory message, you will be asked for the name of a translation table file:

```
Enter name of translation table file:
```

If you are using a custom translation table, enter its name here. Otherwise, press <ENTER> only to use the built-in ASCII-to-EBCDIC table.

You will then be asked for the name of the optional Code Generation file to be used in translation. As with the translation table file, press <ENTER> to specify that no Code Generation file is needed, or enter the name of the file.

Next, the program will ask for the input file(s) to be converted. This can be either a single file (fully-qualified with drive, path, name and extension if needed), or a wildcard-qualified name to convert a number of files.

After the input file is given, the program will ask for the output file. If a wildcard-qualified filename is given for the input file, only a drive and/or a path specifier can be given for output. If a single file is being converted, then a fully-qualified name can be used for output. If the <ENTER> key only is pressed at this prompt, the input file(s) are converted to files with the original names plus an extension of .RFT, and all these file(s) are written into the *current directory*--that is, the directory you were in when you started ASCTORF. As each document is converted, its full name will be displayed.

After each conversion (or set of conversions) is complete, you will be asked again for an input file. To continue, enter the name of a file. To end the program, simply press <ENTER> only at this point.

SAMPLE RUN OF ASCTORF:

```
C>asctorf

ASCII text to Revisable Form (DCA) text file conversion.
Copyright Flagstaff Engineering Ø6/3Ø/86

This program converts ASCII text files to Revisable Form (DCA) text.
If an optional Code Generation file is used, specified byte sequences
in the input ASCII file may be converted to DCA format control
specifications.  If no Code Generation file is used, no special format
control such as underline, bold, etc. will be included in the DCA file.

If no output file name is given, the default is the
input file name with an extension of .RFT.  The input file
name may be a single file name or a wildcard name.  If the
input name is a wildcard name, the output name may specify a
path.  For example "C:\RFTDIR\" would place all output files
on the specified subdirectory.  Output file names would be the
same as the input with an extension of .RFT.

Enter name of translation table file (<ENTER> if none):    <ENTER>
Enter name of Code Generation file (<ENTER> if none):  asctorf.cod

Enter input ASCII file name: \doc\mastlist.doc
Enter output Revisable Form text file name: \temp\
Converting \doc\MASTLIST.DOC
Conversion complete for \temp\MASTLIST.RFT
Enter input ASCII file name:   <ENTER>
C>
```

Convert RFT Text File to ASCII Document

RFTOASC is the opposite of ASCTORF (see previous section): it converts revisable form (RFT) text files to standard ASCII text files, using the rules of an IBM standard called *Document Content Architecture* (DCA); the terms "RFT" and "DCA" are used interchangeably. This allows many word-processing programs and computer systems to share files with applications that use standard ASCII files.

RFT files encode some formatting information (such as tabs and paragraph indents) using control codes which may have no meaning in the ASCII file. Normally, RFTOASC will take some predetermined action when certain of these codes are encountered, and simply omit the rest. For example, the end of a page will normally be marked by the ASCII end-of-page character; tab characters will be expanded to the proper number of spaces. Codes with no standard ASCII meaning, such as begin and end overstrike, will be dropped from the output file.

Flagstaff Engineering has added the option of using a *code generation file* with RFTOASC. This is an ASCII text file containing control code translation statements that allow the user to select translations for many of the RFT codes which would normally default to a preset value or be dropped from the file. Thus, the user can specify what characters are inserted in the file when any of these control codes are found in the RFT document file. The inserted characters can be specified as either ASCII text or as hexadecimal characters, allowing device- or software-specific codes to be put in the file. This allows many of the RFT formatting to be retained in the ASCII document. The format of statements in the code generation file and the RFT codes that are recognized are described in the following section.

A sample code generation file named RFTOASC.COD is included on your distribution diskette; this file can be used as-is for some applications, and can be modified to suit a particular conversion need. (If you modify it, make a copy and modify the copy to preserve the original contents.) The sample file converts RFT control codes into ASCII strings of their mnemonic names inside curly braces--for example, the Begin UnderScore code becomes the ASCII string "{BUS}".

RFTOASC converts the EBCDIC characters of the RFT file to ASCII characters. The program uses an internal translation table by default, but the user can specify a custom translation table if any specific character conversions are required for the document.

In order to convert a number of similarly-named files, wildcard-qualified input filenames can be given, where the characters "?" and "*" signify "any character" and "any string of characters," respectively. Thus, a range of files with a common prefix or extension can be converted in one operation (see example in previous section).

If you convert a single file only, you can explicitly name the target (output) file and path names. However, if you use the wildcard option, you can only give a path name for the target files: all output files will use the input filename with the extension .ASC appended.

USING A CODE GENERATION FILE WITH RFTOASC

There are two types of statements which can be included in the code generation file: *control code translation statements*, and *format display statements*.

Control code translation statements convert RFT codes into user-defined strings of characters. These statements are entered in fixed-field form as follows:

```
    cccc A'aaa  ...  aaa'      or

    cccc X'xx xx ... xx'
```

where:

col. 1-4 ("cccc") is a 2- to 4-letter RFT code
from the following table;
col. 5 must be left blank;
col. 6 to end of line specifies the characters which the RFT code will be converted to in the output file:
-"aaa" indicates ASCII characters inside apostrophes;
-"xx xx" indicates pairs of hexadecimal digits inside apostrophes.

RFT CODES USED IN CODE TRANSLATION STATEMENTS

BUS	Begin underscore
EUS	End underscore
BSUP	Begin superscript
ESUP	End superscript
BSUB	Begin subscript
ESUB	End subscript
BBLD	Begin bold
EBLD	End bold
BK	Begin keep
EK	End keep
BOS	Begin overstrike
EOS	End overstrike
CRE	"Soft" carrier return -defaults to hex 0D 0A if not given
RCR	Required carrier return -defaults to hex 0D 0A if not given
IT	Indent tab -if not specified, required number of spaces inserted
HT	Horizontal tab -if not specified, required number of spaces inserted
PE	Page end -defaults to hex 0C if not given
RPE	Required page end -defaults to hex 0C if not given
ATF0	Align text field (resume normal formatting)
ATF1	Align text field (period alignment)
ATF2	Align text field (comma alignment)
ATF3	Align text field (center text) -defaults to centering with spaces
ATF4	Align text field (end-side alignment)
ATF5	Align text field (colon alignement)
ATL0	Align text line (normal alignment)
ATL1	Align text line (center line) -defaults to centering with spaces
ATL2	Align text line (right alignment)

USING FORMAT DISPLAY STATEMENTS

The second type of statement in an RFTOASC code generation file makes the program place information about the RFT input file in the output file. This information is given as ASCII text in readable form. The following statements can be included in the code generation file:

SHOW FORMAT Places message in output file indicating page size, margins, format changes, etc.

SHOW TABS Places message in output file indicating tab stop locations, types and format changes.

SHOW MARGIN TEXT Places header and footer text in output file where it is found in the RFT document.

EXAMPLE CODE GENERATION FILE STATEMENTS:

```
BUS  A'{bus}'
```

places text "{bus}" in output file when begin underscore code found in RFT input file.

```
EBLD A'{ebld}'
```

places text "{ebld}" in output file when end bold code found in RFT file.

```
CRE  X'0D 0D 0A'
```

places three bytes (two ASCII carriage returns and ASCII line feed) where "soft" carrier return found in RFT file.

RUNNING RFTOASC

To run RFTOASC, type its name at the DOS command prompt. After introductory message, you will be asked for the name of a translation table. If you are using a custom translation table file, enter its name here. Otherwise, press <ENTER> only to use the built-in EBCDIC-to-ASCII table.

The next prompt asks for the name of a code generation file, described above. Enter its name here and press <ENTER>, or press <ENTER> only if no code generation file is being used.

Next, the program will ask for the input file(s) to be converted. This can be either a single file (fully-qualified with drive, path, name and extension if needed), or a wildcard-qualified name to convert a number of files.

After the input filename is given, the program will ask for the output filename. If a wildcard-qualified filename is given for the input file, a drive and/or a path specifier can be given for output. If either of these is given, the output file(s) will be placed on the given drive and in the given subdirectory. If a single file is being converted, then a fully-qualified name can be used for output. If the <ENTER> key only is pressed at this prompt, the input file(s) are converted to files with the original names plus an extension of .ASC, and all these file(s) are written into the *current directory*--that is, the directory you were in when you started RFTOASC. As each document is converted, its full name will be displayed, including any path names.

After each conversion (or set of conversions) is complete, you will be asked again for an input filename. To continue, enter the name of a file. To end the program, simply press <ENTER> only at this point.

SAMPLE RUN OF RFTOASC:

```
C>rftoasc

Copy IBM Revisable form (DCA) text to an ASCII file.
Copyright Flagstaff Engineering 06/30/86

This program converts revisable-form text documents to ASCII
text files.  If an optional Code Generation file is used,
certain DCA codes can be translated to user-specified
character strings.  If no Code Generation file is used, DCA
codes such as tabs and indents will be expanded to the
correct number of spaces and other DCA codes will be dropped
from the output file.

If no output file name is given, the default is the
input file name with an extension of .ASC.  The input file
name may be a single file name or a wildcard name.  If the
input name is a wildcard name, the output name may specify a
path.  For example "C:\ASCDIR\" would place all output files
on the specified subdirectory.  Output file names would be the
same as the input with an extension of .ASC.

Enter name of translation table file (<ENTER> if none):    <ENTER>
Enter name of Code Generation file (<ENTER> if none):      <ENTER>

Enter input Revisable-Form text file: \testdata\sample1.rft  <ENTER>
Enter output ASCII file name: \testdata\
Converting \testdata\SAMPLE1.RF
Conversion complete for \testdata\SAMPLE1.ASC

Enter input Revisable-Form text file:   <ENTER>
C>
```