Marist College Report Writing using Focus

Edition Modification Date: June 2002

TABLE OF CONTENTS

I. Introduction

Focus can be used as a database package and/or as a reporting tool to extract data from a database. At Marist, Focus is used as a reporting tool. The databases we extract information from are the various IA Systems:

Marist is using four IA Systems:

SIS	*Student Information Systems* SIS stores student information. This includes their biographic, demographic, and academic information including courses they have taken, grades received, financial aid, and billing information.
ADS	*Alumni Development System* ADS stores alumni information. This includes their biographic, demographic, and gift information.
HRS	*Human Resource System* HRS stores employee information. This includes their biographic, demographic, job code, and benefit information.
FRS	*Financial Records System* FRS stores payroll information and accounts payable/receivable information.

The reports you can create with Focus include:

List — A LIST is a report normally printed on paper presented in tabular format. For example:

Student Name	**Student ID**	**Major**	**Classification**
Jane Doe	999-99-9999	COMJ	Senior
John Smith	888-88-8888	Engineering	Junior

Label — Address Labels are available. At Marist we create address labels for students, alumni, faculty, and staff.

Matrix — A MATRIX is normally a compilation of information depicted in table format. For example:

	Benoit	Gregory	Sheahan
Female	6	6	162
Male	6	6	136

Nonprint — A NONPRINT is basically a list that is saved to a file. It can be formatted in a variety of ways. It is normally downloaded to be used in another application, such as a Word Processor or Database Package. It can also be used in another mainframe application.

Access to the various IA systems is determined by job function. This means you may only have access to one IA system, for example—SIS. If you only have access to sign-on to SIS and use specific screens, then you also only have access to report from SIS. The security is such that you will only have access to particular systems AND fields within that system.

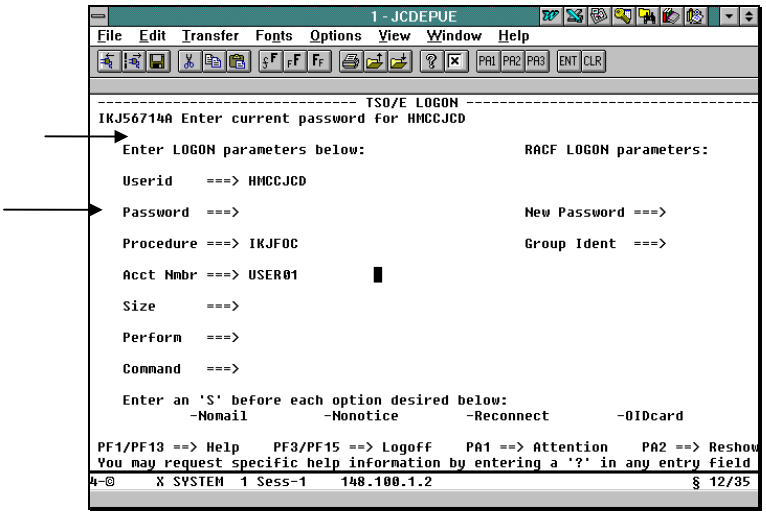
II. Sign-on procedure

The Focus Report Writer can be accessed through your TSO account. The following procedures explain this method.

When you come to the Marist Fox log-on screen - type **TSO** at the **System ID =➔** field. Press **ENTER**. You will then be brought to a secondary screen where you will once again type **TSO** and press **ENTER**.

Now enter your HMxx user ID that will vary by department (for example, your User ID might be HMCCxxx - CC standing for Computer Center (A very old name for Information Technology Department)).

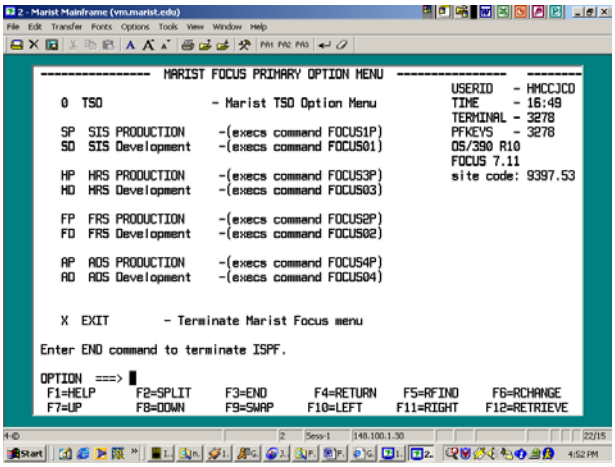
The following screen will display, prompting you for your *password*: (Note: your TSO password must be at least four characters: 3 letters and 1 number.)



Type your **password**. Press **Enter**.

Three*** will display. Just press **Enter** when they display.

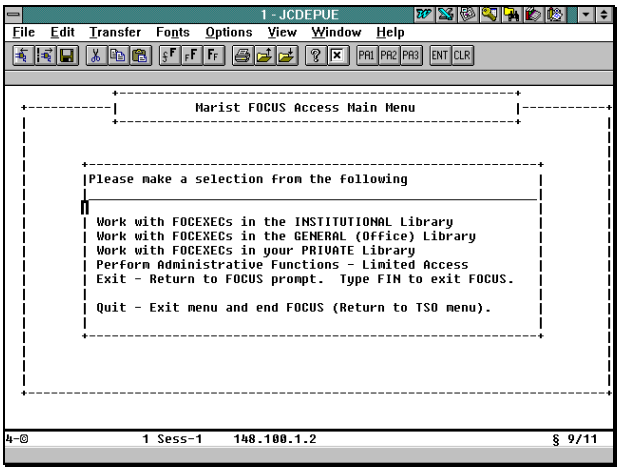
The following screen will display.



Type the two letter character for the system from which you wish to report at the ***OPTION*** **===>** line. Press **Enter**. For instance, to report from SIS Production, type **SP** at the ***OPTION*** **===>** line and Press **Enter**.

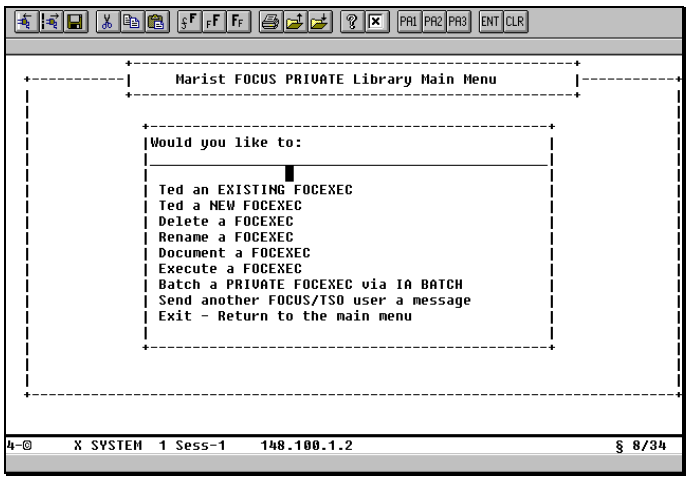
A screen with double greater than signs will display as follows. (This is the Focus prompt) Type **EX MENU**. Press **Enter**.

A menu similar to the following will display:



Tab down to *Work with Focexecs in your Private Library*. Press Enter.

Your *Private Library* menu will display:



You have completed the sign-on process.

III. Sign-off procedure

Whether you have been creating or editing a FOCEXEC or have just finished batching a FOCEXEC, when you press F3 you will be returned to the *Marist Focus Access Main Menu*. Tab to *Quit* and press **Enter**.

A message will display which reads Leaving Focus. Press Enter when *** appears. When the *** appear, press **Enter**.

Type X on the *Option* ===> line on the *Marist Focus Primary Option Menu*. Press **Enter**.

At *Ready*, type LOGOFF. Press **Enter**.

IV. Changing your password

When you use Focus, you are using a *TSO* account. Your password for TSO will expire every 90 days. At that time you will be required to change your password before proceeding. You can, however, change your password as often as you would like.

Information Technology, in accordance with a request by the Auditors at Marist College, requires that your TSO password be at least 4 characters and can be up to 8 characters. It must begin with a letter and must contain a number.

To change your password, type your old password in the *Password field* of the *TSO/E Logon* screen. Then TAB to the *New Password* field. Type your new password. Press **Enter**. You will be asked to type your new password (for verification) again. When complete, the TSO Logon screen will process and you will be at a Focus prompt. (See Sign-on procedure if you require assistance in logging on.)

V. Focus Libraries

Three libraries are/will be available to you as a Focus user.

Private Library	All FOCEXECS you create will be stored in your Private Library. This library is accessible by choosing *TED an Existing Focexec* from the Focus menu. The Focexecs, which are displayed after choosing this option, can only be accessed by you.
General/Office Library	The General (or Office) Library is one which is managed by your office *Report Coordinator*. This library is one, which can be accessed by anyone in your office with Focus access. The *Report Coordinator* moves Focexecs from Private Libraries to the General Library.
Institutional Library	The Institutional Library has been available since February 1996. Contained within this library, are Focexecs, which are general in nature and designed to receive input from you, the user, and create a report, based on your answers. Most of the reports from SIS will be available in SIS Plus, there may be a few minor changes.

VI. Documenting your reports

Naming your exec

When you create a new Focexec, the first step is to name the exec. The name can only contain eight characters or less, no special characters. The standard naming convention at Marist is as follows:

Any focexec you create for your private library, the first two characters of your Focexec must be the last two characters of your TSO Account (not your subcode). For instance, if your TSO userid is HMCCJCD, then HMCC is your MAIN account and denotes the office/area you work in. The last three characters are your subcode (usually your initials or some part of your initials.) If HMCC is your main account, then the last two characters are CC so any Focexec you create should be named CC_ _ _ _ _ _ (CC and up to six other characters.)

All Focexecs in the general library for your office will be managed by your area's Report Coordinator. This person may be you or it may be someone in your office/area designated to manage reporting for your area. When a focexec is placed in the general library, the first character of the name must be a G. The second and third character must be the two characters of your TSO account for your area. See paragraph above. For example, a focexec placed in the Computer Center's general library might be named GCCundgr.

Documenting your exec

As described in the library section of this guide, you currently have access to your private library and your general (office) library. Depending on your function within your office/area you will have responsibility for at least your private library and maybe the general as well. The following information applies to both. You must create a file on your account (if it has not already been set up for you) named $$DOC. This file will display at the top of your private library. It should contain the name of the Focexec available in your private library and a short description of what the Focexec does (prints). As this file grows, it will become an index of your Focexecs, which you can open and read when you are looking for a particular Focexec. Without this file you would need to open every Focexec until you found the correct one.

To create your $$DOC file, choose *Creating a New Focexec* from the Focus menu. Name your exec $$DOC. As you write Focexecs, record them in this file as follows:

CCUNDRGD	This Focexec selects currently registered undergraduate students. It prints their name, id, and current attempted hours sorted on major.
CCGRAD	This Focexec selects currently registered graduate students. It prints their name, id, sorted on major.

Within your Focexec, you must state who created it, the date it was created, and other information critical to documenting the exec. **After** you create a Focexec, choose *Document a Focexec* from your Marist Focus Private Library Main Menu.

You will be prompted to choose an exec from your library. Select one and press Enter. Next you will be prompted to enter your Name and Office, A one-line description of your Focexec, and then a detailed description of your Focexec. If you have included and -INCLUDES in your exec these will be copied in. If you include any &variables, you will need to edit your EXEC and describe these. The purpose of documenting your report is twofold. First, it gives the user of the report information about what the report will do and secondly this information can be extracted at a later date to track system usage.

Headers/Footers you should include on every report

Your reports should have identifying information on every page such as a Descriptive Title, the current Date, the name of the Exec, etc. The simplest way to handle this is to create a Focexec called _ _Head (replacing _ _ with the first two characters of your HM account) and copying the Exec in every time you need it. To do this, create a new Focexec entitled:

_ _HEAD

(Replacing _ _ with the first two characters of your HM account)

On the command line type CASE M (Press Enter) to use Mixed Case and type the following exactly as it appears. Be sure to pay attention to CASE.

```
HEADING CENTER
"Marist College Confidential"
"Descriptive Title of Report"
"Name of Exec     &DATE </2"
FOOTING BOTTOM
"Please direct any questions concerning this report to NAME at EXTENSION"
```

Save this Focexec by pressing **F12** and on the *Command Line*, then type **FILE** (**Press Enter**).

Copying Header/Footer into your Focexec

When you are ready to copy this exec into your existing Exec, you must pay attention to placement. Headers/Footers should be placed within your LAST TABLE FILE REQUEST. So if you've used any hold files and then joined files together, make sure that you copy the HEADER/FOOTER exec into your LAST Table File Request—the one after the JOIN. You can do this easily by placing your cursor in the Prefix Area, which precedes the TABLE FILE XXX line in your exec. Type a forward slash - /. Press Enter. This will make the TABLE FILE XXX line the *current* one. Press F12 to move your cursor to the Command Line. Type:

GET FOCEXEC(_ _ HEAD

If you copy the Exec to the wrong place in your Focexec, you can move it by using **MM** and **F** in the Prefix Area of the Editor. (See the *TED Editor* Section—Prefix Area Commands.)

Both the Heading and Footing will print on every page of your report. Headers/Footers are explained in more detail in *Report Formatting (Section XVII)*.

Amper Variables

If you include amper variables such as those, which prompt you for current semester, major, etc. you should add some type of reference to them so that your answers to these questions print with your report. This will assist anyone reading the report in interpreting the criteria you used to run the report. The following illustrates how to accomplish this with a SUBFOOT (You could also create a title page as you see in the Institutional Library.)

```
ON TABLE PAGE-BREAK AND SUBFOOT
"Semester variable used was &xxx, Major variables used were &maj, etc."
```

This line can be placed before the last END in your exec:

```
TABLE FILE RTFILE
HEADING CENTER
“Marist College Confidential”
“All Div. of Humanities Undergraduates including CUM GPA’s”
Print RT010 , etc.
FOOTING BOTTOM CENTER
“Please direct any questions concerning this report to NAME at EXTENSION”
“ “
“ “
ON TABLE PAGE-BREAK AND SUBFOOT
“Semester variable used was &SEM, Majors entered were &MAJ”
END
```

These lines will create a page break at the very end of your report and print out what you filled in for the amper variables.

Adding Comments to your Focexec

You may wish to add comments to your Focexec which tell you or someone else in your office what the Exec does, how to run it, etc. You can do this by adding comment lines. You have two choices of comment lines:

-* A dash asterisk is used when you only want to see the comments when you open the file and read it.

-TYPE A dash TYPE is used when you want the information (text) you put on your comment lines to print out along with your Focexec. This is particularly useful if you are using Amper variables. You can write a paragraph out on the top of your screen that reminds you what data to input.

An example:

```
-TYPE This Focexec selects undergraduate freshmen
-TYPE only. They must be full-time to be included in
-TYPE report. It will print their ID, Current Attempted
-TYPE Hours, and Classification.
TABLE FILE RTFILE
PRINT RT14C RT020
BY RT010
IF RT005 EQ &SEM
IF RT14C EQ ‘FR’
IF RT020 GE 12
END
```

The text following the **–TYPE** displays on your screen as your Focexec is running.

VII. TED Editor

All Focexecs are written and edited in the TED Editor. From the Focus Prompt (> >) type TED FOCEXEC(name of Focexec) or if you are using EX MENU, you will be using the TED Editor if you choose to Create or Edit a Focexec.

The initial screen will display without a prefix area. You can use a prefix area to copy lines, insert lines, etc. To add prefix, type **NUM ON** on the Command Line in the TED Editor or type **EDIT**. NUM ON will put numbers in the prefix area, EDIT will put five ===== signs in the prefix area. Although you can use the prefix commands listed in the second table in either of these methods, NUM ON is sometimes more helpful if you get an error while processing your Focexec. If there is a problem with a particular line of your Focexec, Focus will tell you the line number. With NUM ON, you can see that line number.

Keys

Key	Function	Key	Function
PF1	Meaning of the keys and Help information	**PF7**	Move BACK one full screen
PF2	Insert line after cursor	**PF8**	Move FORWARD one full screen
PF3	Quit	**PF9**	
PF4	PAINT	**PF10**	Move to RIGHT one half page
PF5	Repeat last command	**PF11**	Move to left one half page
PF6	RECALL last command	**PF12**	Command Line

Prefix Area Commands

Command	Function	Command	Function
==/==	Becomes current line	**==DD==**	Delete block
==Dn=	Delete n lines	**==MM=**	Move block
==In=	Insert n lines	**==CC=**	Copy block
==An=	Add n lines	**==P==**	Move or copy Prior
==“n=	Duplicate n times	**==“”=**	Duplicate block
==Mn=	Move n lines	**==SP=**	Split line (at cursor)
==Cn=	Copy n lines	**==J==**	Join line (at cursor)
==F==	Move or copy Following	**==P==**	Move or copy Prior

Command Line Operations

(Typed on the command line. There are many listed here. Some you will use often, others you may never use.)

Command	Function
Add n	Add n lines after current line
Backward n	Move backward n pages
Bottom	Go to bottom of file
CAse m|u	**M**ixed upper/lower case|**u**pper case
Cdel	Delete line pointed to by cursor
Change /old/new/ n	m Change old to new n times on m lines (or * *)
CINS	Insert line after line pointed to by cursor
Curline n	Set current line to specified line position
DOwn n	Forward n lines
Duplicat n target	Duplicates from current line until target n times
Edit	Mode with 5 character prefix area
File :fileid:	Save file as fileid and end session
FILEName newfilename	Change the default filename used for FILE and SAVE commands (CMS and TSO)
FOrward n	Forward n pages
Get fileid	Get a file or get stack if no fileid given
Help	Retrieve the HELP file
Input string	Insert line after current line
Join	Join line after cursor to cursor position
LEft n	Move one full screen to left or n columns
LEFTP	Move one half screen to left
Locate /string/	Locate a string, search forwards
LOWercas target	Set print to lowercase-from current line till target
Next n	Move forward n lines
Number on|off	Set up prefix area with numbers
Qquit	Quit if changes
Quit	Quit if no changes
RECover n	Recover lines that were deleted
RESet	Reset to original mode; cancel pending prefix operations
RIght n	Move one full screen to right or n columns
RIGHTP	Move one half screen to right
RUn :parms:	If editing a FOCEXEC then file and execute it with the specified parameters
SAve :fileid:	Save file as fileid and continue
Scale	Display a scale at the top of the screen
Split	Split line at cursor position and create a new line
TEd fileid	Edit another file
Top	Go to top of file
Type	Set mode to data display, no prefix area
Up n	Backward n lines
UPPercas	Set print to uppercase from current line till target
- /string/	Backward search
=	Repeat last command
?	Show last command
?F filename	Show fields in file filename
?FF filename	Show fields, aliases and formats

NOTE: In the above list of commands the upper-case letters show the minimum letters needed to issue a valid truncated command.

Command Repeat. Any command which is preceded by a "&" remains on the command line and is not erased when the ENTER key is pressed. The common commands you will use include:

MOST COMMON COMMANDS	THEIR FUNCTION
TOP	Moves you to the top of your file
BOTTOM	Moves you to the bottom of your file
EDIT	Adds the prefix area so you can use prefix commands
NUM ON	Adds line numbers in the prefix area.
SAVE	Saves your work but doesn't close the file
FILE	Files your work and closes the file
LOCATE /string/	Especially useful for searching your $$DOC file looking for certain keywords like LOCATE /major/
Qquit	Quit even if you've made changes. Your changes are not saved.
Quit	Quit if you have not made any changes.
RUN	You've finished editing your FOCEXEC and wish to run it on line.
CASE M	When adding headings/footings set the case to mixed if you wish so that your text is not all uppercase. (Uppercase is default).

VIII. Creating a simple FOCEXEC

Required elements of a FOCEXEC

A Focus report request will contain at least the following elements (most of the time):

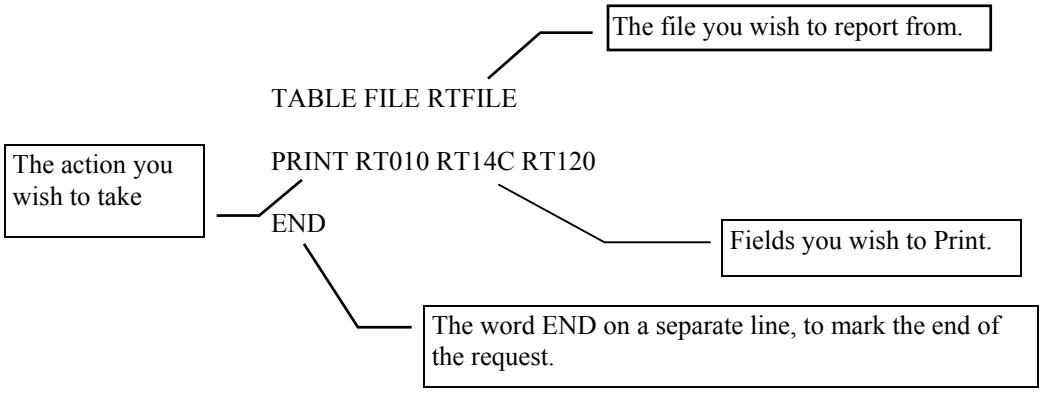


TABLE FILE

The word TABLE is defined as *create a tabular report*. The table file command in your Focexec tells Focus where to find the data you wish to report from. For instance, if you are an SIS user and wish to print Student ID, Classification and Major, you will use the RTFILE. This is where the semester (term) information about a student is stored. If you are an HRS user, and wish to print employee's home addresses, you will most likely use the EAFILE. If the data you wish to report from exists in more than one file, you will first join the two or more portions together. The TABLE FILE command will reflect the HOST FILENAME. This will be discussed in JOINING FILES. (I do not recommend recreating the above example EXACTLY as written. It is for example only.)

VERBS

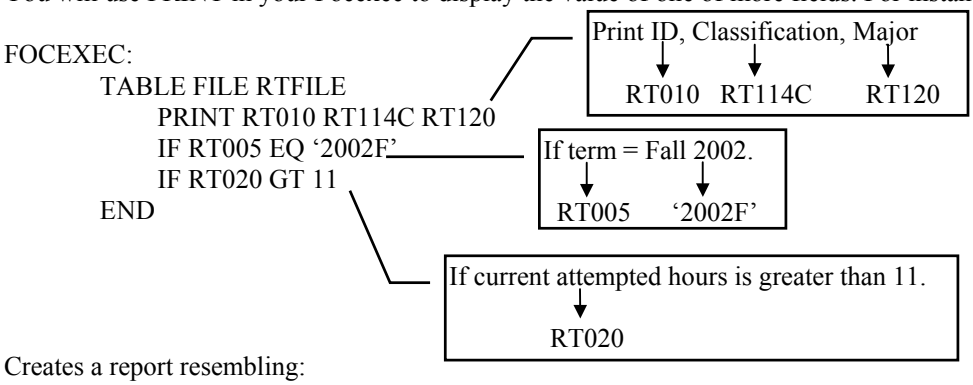
You will use VERBS to display your data. For instance, to print a student's id, name, and GPA, you will use the PRINT verb: PRINT

The four verbs you will use are:

PRINT	Displays field values
LIST	Displays field values and puts a *number* to the left of each record.
SUM	Adds values of numeric fields and displays the result or displays the last value for alphanumeric data
COUNT	Displays the number of times a value that appears in the database for a given field.

The PRINT verb

You will use PRINT in your Focexec to display the value of one of more fields. For instance:

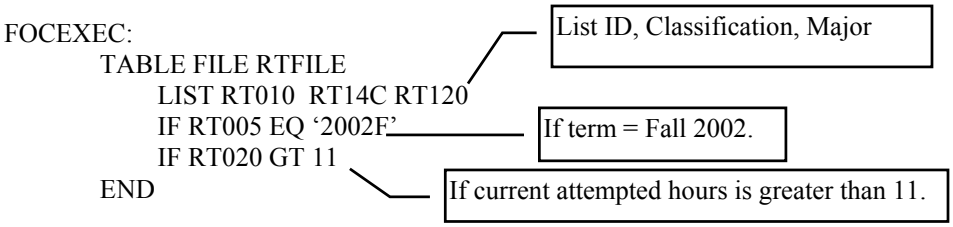


Creates a report resembling:

Student ID	Classification	Major
009240099	JR	ENGL
999556666	SR	COMR
333445555	FR	CIS
444556666	SO	MATH

The LIST verb

To display individual field values and number each row, type the command LIST followed by the name of each field you want in the report. For example



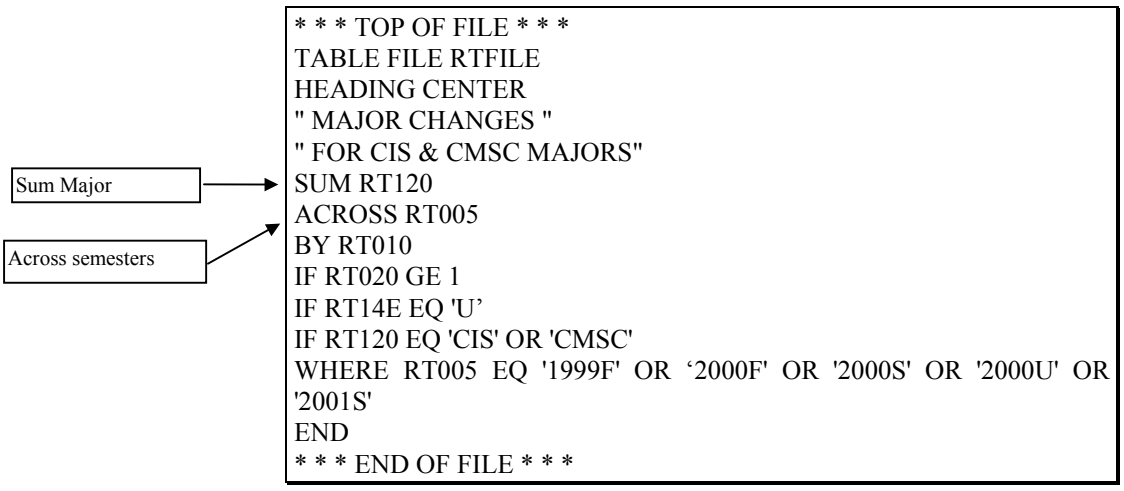
Creates a report resembling:

List	Student ID	Classification	Major
1	009240099	JR	ENGL
2	999556666	SR	COMC
3	333445555	FR	CIS
4	444556666	SO	MATH

The SUM verb

To display the total value of one or more numeric fields, use the SUM command followed by the name of the field you want in the report. Also, you can use SUM on alphanumeric fields. This will help to eliminate duplicate records from printing. Something you can do with Focus is look at data across semesters. For instance, you can look at a number of semesters and print the primary major for a group of students during that semester. This would indicate when a student changed his/her major. For instance:

This FOCEXEC:




```
* * * TOP OF FILE * * *
TABLE FILE RTFILE
HEADING CENTER
" MAJOR CHANGES "
" FOR CIS & CMSC MAJORS"
SUM RT120
ACROSS RT005
BY RT010
IF RT020 GE 1
IF RT14E EQ 'U'
IF RT120 EQ 'CIS' OR 'CMSC'
WHERE RT005 EQ '1999F' OR '2000F' OR '2000S' OR '2000U' OR
'2001S'
END
* * * END OF FILE * * *
```

produces the following report:

MAJOR CHANGES
FOR CIS AND CMSC

	1999F	2000F	2000S	2000U	2001S
Student Number					
000000000		CIS			
111111111		CMSC		CMSC	CMSC
222222222		CMSC		CMSC	CMSC
333333333				CMSC	
444444444				CMSC	
555555555				CMSC	
666666666				CMSC	
777777777	CMSC	CIS	CIS	CIS	CIS
888888888	CMSC		CMSC		
999999999	CIS	CIS	CIS	CIS	CIS

Student Id #777777777 was a CMSC major in the Fall of 1999. In the Spring of 2000, he/she switched to CIS and remained a CIS major until at least Summer of 2000.

What if you did not use SUM and instead used PRINT (a detail verb). This is what your report would look like:

	2000F	2000F	2000S	2000U	2001S
Student Number					
000000000		CIS			
111111111		CMSC			
				CMSC	
					CMSC
222222222		CMSC			
				CMSC	
					CMSC
333333333				CMSC	
444444444				CMSC	
555555555				CMSC	
666666666				CMSC	

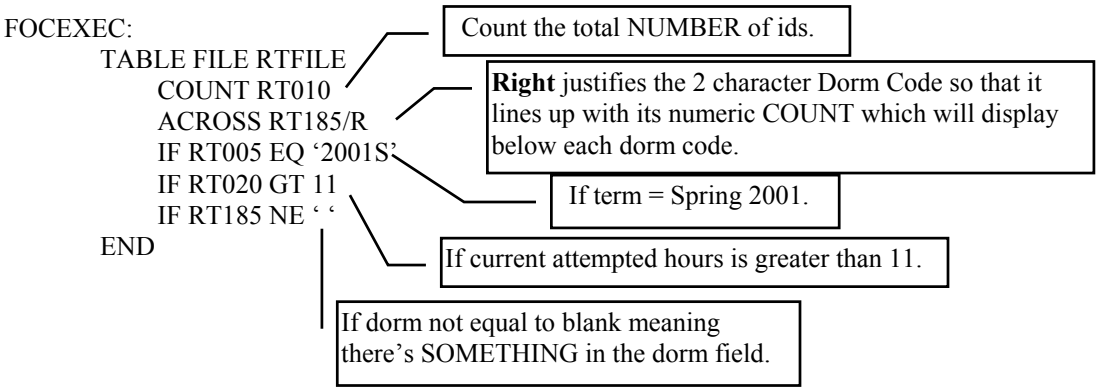
777777777	CMSC					
		CIS				
			CIS			
				CIS		
						CIS
888888888	CMSC					
			CMSC			
999999999	CIS					
		CIS				
			CIS			
				CIS		
						CIS

You can see how your report became much longer. This is because when you SUM the Primary Major field as in the first Focexec, it looks through the RTFILE, pulls every record for a student who was active in at least one of the semesters specified and BEFORE printing out your report, pulls only the **last occurrence** of the Major for the semester. Using SUM therefore, compacts your report in this example, thereby improving readability.

In the second example however, using a detail verb such as PRINT, Focus is giving you detail about each student's record—even if the student did not have a major in a particular semester. You are seeing each individual record. SUM compacts your printout by reading the student's record. If they have CIS or CMSC for the semester specified it grabs that records. Then, before it gives you back the data, it strips out all but the last value for the semester.

The COUNT verb

To display the total number of values for a particular field use COUNT followed by the name of that field. For instance, to find out how many students reside in each dorm:



Creates a report resembling:

	Dorm, Code									
BE	CH	GA	GR	LE	MA	MR	SH	TC	TH	WC
31	436	282	34	319	136	335	125	31	334	208

This Focexec is a matrix. The exec looks at the dorm field. If there's something in the dorm field, in COUNTS how many times a new id number is associated with a dorm code. It does NOT add up the id numbers. It simply counts the number of occurrences and produces the above matrix.

END

The END command must be typed at the end of every Focexec you write.

Maximum Number of Fields

The maximum number of fields that you can name in a request is 95. The maximum length of an output line is 256 columns.

Step by Step for Creating, Saving, Running your exec

1. Sign onto your TSO account and proceed through the *Sign-on Procedure* section
2. When the Marist Focus Access Main Menu displays you can either Create a new Focexec or TED and old one. (If you create a new one, you must name it first then you will be put in the TED Editor.)
3. Type EDIT or NUM ON to turn the prefix area on.
4. If you want to copy another Focexec into your new one type GET FOCEXEC(nameofFocexec) Press Enter.
5. If you don't wish to copy another Focexec in, go to the prefix area, insert some lines and begin typing your exec.
6. When you have finished, press Enter.

   Type one of the following on the Command Line:

RUN	To run your report on-line
SAVE	To save your work and stay in the file
FILE	To save your work and exit the file

7. After issuing any of the above commands, you may print your report. See *Printing in Focus*
8. Don't forget to document your report by typing EX DOCUMENT at a Focus Prompt.

IX. Exercise Set One—Verbs

Exercises will be provided to you in class. The exercise set you receive is dependant upon the primary database for your office—SIS, HRS, FRS or ADS.

X. Organizing Report Data

After you have a basic understanding of how to retrieve and print data from your files, it's time to move on to organizing data. This section will cover:

- BY, for sorting vertically
- ACROSS, for sorting horizontally
- using BY and ACROSS together

BY, for sorting vertically

Let's look at the following example:

```
TABLE FILE RTFILE
PRINT RT010 RT120
END
```

Focus knows from this Focexec that you wish to look at the RTFILE and print STUDENT ID (RT010) and MAJOR (RT120). It does not, however, know in what order you wish to organize this data, so it gives it to you as it finds it similar to:

111111113	SPEC
111111114	SPEC
111111115	SPEC
111111116	SPEC
111111117	SPEC
111111118	SPEC

Let's assume you want to group the majors together and you don't want to repeat the printing of the major code unless it's a different code. So, this is what you want:

ACCT	111111111
	222222222
BUS	444444444
	555555555
	666666666
BUSA	777777777
	888888888

You can achieve this by using a BY statement in your Focexec. Let's take the example above and modify it.

```
TABLE FILE RTFILE
PRINT RT010
BY RT120
IF RECORDLIMIT EQ 100
END
```

We have modified this exec to print all of the id numbers (RT010) that fall under a specific major (BY RT120). Remember the PRINT verb is a detail verb. When you want to know every ID number, then use PRINT. The BY verb GROUPS and SORTS your data. It also appears in the far left column. All of the fields you list in the PRINT statement display to the RIGHT of the fields which are contained in the BY statements.

You can use more than one BY statement in your Focexec. When you use more than one, the first sortfield you name is the *primary* sortfield. Others are *secondary* sortfields.

Let's modify our Focexec further to sort first on major (RT120) and then by class year—frosh., soph, etc. (RT14C)

```
TABLE FILE RTFILE
PRINT RT010
BY RT120
BY RT14C
IF RECORDLIMIT EQ 200
END
```

Note: Remember—the IF statement which limits the RECORDLIMIT is there for example because I don't want to search the entire RTFILE at this point in our class.

Lets see an example of our report:

Prim Mjr 1	Class	Student Number
ACCT	FR	333334444
	SO	0000xx667
BUS	FR	111111110
		222222000
		000000333
		444333333
		998899889
	JR	777667766
	SO	454545454
		900090009
BUSA	SO	222222222
BUSF	FR	45x322212
		9-8877dd7
	JR	999999999

The first BY statement—BY RT120 is the Primary sort field. It prints in the very first column. The second BY statement—BY RT14C is a Secondary sort field. It prints to the RIGHT of the first BY statement. (If there were more BY statements, they would continue to print to the RIGHT of the one before it.) After the BY statements, the PRINT fields will begin printing. Therefore after BY RT14C, Focus goes back up to your PRINT statement and prints the ID Numbers (RT010).

IMPORTANT The maximum number of sort phrases you can include in a single report request is 32, five of which can be ACROSS phrases. (Next section.)

ACROSS, for sorting horizontally

Let's assume we are looking for a total count of alumni in each reunion year for a specific range. We could use ACROSS to display this information horizontally:

```
TABLE FILE LPFILE
COUNT CID_LP
ACROSS REUN_YR_1
IF REUN_YR_1 EQ '2001' or '2002'
END
```

The COUNT statement is counting the number of Alumni ids within each Reunion Year (ACROSS REUN_YR_1). It is only looking at 2001 and 2002 because of the IF statement—IF REUN_YR_1 EQ '2001' OR '2002'.

This Focexec would produce a report similar to:

```
 Primary,Degree,Year 1
2001          2002
883           742
```

What if you wanted to look at Gender and print totals for each on the same report? You could modify the Focexec:

```
TABLE FILE LPFILE
COUNT CID_LP
ACROSS REUN_YR_1
BY SEX_LP
IF REUN_YR_1 EQ '2001' or '2002'
END
```

Your report would now look like:

```
    Primary,Degree,Year 1
     2001   2002
Sex
----------
F     504    76
M     379    666
```

Changing the Sort Sequence

As you saw in the examples for BY and ACROSS you can determine how your report looks AND how the data is sorted. You learned that if you do not use a SORT phrase, Focus displays the records as they appear in the database. When you use a SORT phrase, Focus sorts from low-to-high (A to Z and zero to nine). To reverse the order and sort from high to low, add the keyword HIGHEST to the SORT phrase.

For example:

```
TABLE FILE RAFILE
PRINT RA005
BY HIGHEST RA380
IF RECORDLIMIT EQ 100
END
```

This exec produces the following report:

GPA	Student Number
4.000	002002002
	003003003
3.950	004004004
3.940	005005005
3.860	006006006
3.840	002007007
3.783	008008008
3.750	009009009
3.710	003003002
3.700	002444333
	009009009
3.675	002002003
3.670	009888888
3.663	002333333
3.654	003333333
3.650	099999999

If you omit highest, the report would have sorted the GPA from low to high.

Selecting a Range of Records for the Sort

If you only want a subset of records—CUM GPA from 3.5 - 4.0—for instance, you can append this **screening condition** to the sort phrase. For example:

```
TABLE FILE RAFILE
PRINT RA005
BY HIGHEST RA380 FROM 3.5 TO 4.0
END
```

Please note: another method for screening records is to use an IF or WHERE statement discussed later in this manual.

XI. Exercise Set Two – Organizing Report Data

Exercises will be provided to you in class. The exercise set you receive is dependent upon the primary database for your office—SIS, HRS, FRS or ADS.

XII. Selecting Records

When you choose a subset of records for your report, you are screening the records based on some criteria. Focus has two functions, which enable you to screen records:

- IF
- WHERE

This section describes how to use these two screening commands.

IF

To determine which records to choose, Focus compares a field in the database (file) to a value you supply using a relation (such as "equal to" or "greater than".) For example:

Let's assume you are using the RTFILE. At present, the RTFILE contains approximately 170,000 records. You certainly don't want the entire RTFILE. You want a subset of that file. You want to select all records, which have the major field (RT120) equal to English Literature (ENGL). You could write the following in your Focexec:

```
IF RT120 EQ 'ENGL'
```

RT120 is the major field in the RTFILE. EQ is a "relation" which means Equal To. ENGL is the 4-letter code for English Lit. Focus will use this IF statement to look through the RTFILE and return ALL records to you which contain ENGL in field RT120.

What if you only want the present semester, say Fall 2002? Then you can include two IF statements as follows:

```
IF RT120 EQ 'ENGL'
IF RT005 EQ '2002F'
```

RT005 is the TERM field in the RTFILE. '2002F' is a valid value for RT005. It means 2002 Fall.

> Determining valid values for fields
>
> You can determine valid values for fields by using PF1 in IA. If you know the field name, you can type it in the first field and press Enter. A list of valid value will display. If you don't know a field name, you can go to an IA Screen, which displays the information you want. Put your cursor where the information prints out on the screen and press PF2. Again, you will get a list of valid values provided you have access to look at that field and your cursor is in the right position on the screen.

The order of the IF statements in the Focexec **doesn't** matter to Focus. You can use as many IF statements as you need to define the selection criteria.

Notice that in the examples above, ENGL and 2002F are enclosed in single quotes. All non-numeric data must be enclosed in single quotes for IF and WHERE statements. You do not need the quotes for numeric data.

Where

WHERE is very similar to IF. The syntax is very similar, meaning you would write:

WHERE RT120 EQ 'ENGL';

One difference is the addition of a semi-colon at the end of the WHERE. Another is that you can string multiple criteria in one WHERE statement, for example:

WHERE (RT120 EQ 'ENGL') AND (RT005 EQ '2002F');

NOTE: For compound expressions, enclose the test phrase in *parentheses* as demonstrated above.

WHERE also allows you to select records based on calculations, for example:

WHERE (RT020 + RA320) GE 30;

means add the student's current attempted credits to his total credits and if this number is greater than or equal to 30, I want his record (maybe it's Spring and I'm looking for potential sophomores to hire for the coming Fall semester.) NOTE: Enclose calculations in parentheses to give them priority over other operations.

WHERE can be used to compare fields to one another.

WHERE FIELD1 EQ FIELD2;

Relations for WHERE and IF

The relation defines the test condition and can be any of the ones in the following table.

To test for records with field values	Use the relation	Given the example	Focus displays
Identical to the test value	EQ IS LIKE (only for WHERE)	**IF RT005 EQ '2002F'** **WHERE RT005 IS '2002F'**	Returns all records which have a term of Fall 2002.
Different from the test value	NE IS-NOT	**IF RT185 NE ' '** **WHERE RT185 IS-NOT ' '**	Returns all records which do not have a blank dorm code
That have specific characters in a specific location	IS with a mask EQ	**WHERE RT120 IS 'COM$*'** **IF RT120 EQ 'COM$*'**	Return all students that have COM at the beginning of their major, (so all Communication majors.)
That do not have specific characters in a specific location	IS-NOT with a mask	**WHERE RT120 IS-NOT 'COM$'**	Exclude all Comm. majors.

To test for records with field values	Use the relation	Given the example	Focus displays
That have specific characters anywhere in the test value	LIKE with a mask (for WHERE only CONTAINS	**WHERE AA003 like '%smith%'** **WHERE AA003 CONTAINS 'smith'**	Selects all students with smith anywhere in their name.
That do not have specific characters anywhere in the test value	NOT LIKE with a mask (for WHERE only) OMITS	**WHERE AA003 NOT LIKE '%smith%'** **WHERE AA003 OMITS 'smith'**	Selects all students who do not have smith anywhere in their name.
Greater than the test value	GT	**WHERE RT020 GT 11** **IF RT020 GT 11**	Selects full-time students only.
Greater than or equal to the test value	GE	**WHERE RT020 GE 12** **IF RT020 GE 12**	Selects full time students only
Less than the test value	LT	**WHERE RT020 LT 12** **IF RT020 LT 12**	Selects part time students only
Less than or equal to the test value	LE	**WHERE RT020 LE 11** **IF RT020 LE 11**	Selects part time student only
Within a range of test value	FROM ... TO	**WHERE RA380 FROM 3.0 TO 4.0** **IF RA380 FROM 3.0 to 4.0**	Selects students with a cum. gpa from 3.0 to 4.0
Outside a range of test value	NOT-FROM... TO...	**WHERE RA380 NOT-FROM 3.0 TO 4.0** **IF RA380 NOT-FROM 3.0 TO 4.0**	Selects students who do not have a cum gpa between 3.0 to 4.0

Rules for IF and WHERE

You can use any fieldname in your IFs and WHEREs as long as it:

- exists in the Master File Description for the file you are reporting from, or
- exists in a Master File Description for a file joined to the database you are reporting from, or
- was created as a DEFINEd field for this file prior to this TABLE request, or
- is a COMPUTEd field in this request

NOTE: To screen on COMPUTEd fields or on the results of direct operations, you must use the word TOTAL in the selection statement.

Limiting the Number of Records: RECORDLIMIT

Although we have not defined RECORDLIMT up to this point in this manual, we have used it repeatedly in most of the exercises. We use RECORDLIMIT to test our Focexec—make sure it is correctly constructed before batching it. We use RECORDLIMIT when running our exec on-line. When we are ready to batch it, we remove this line.

RECORDLIMIT is used with an IF statement, for example:

IF RECORDLIMIT EQ 100

means process my Focexec using only the first 100 records you find which meet the criteria I have specified.

Realize, though, when you use a RECORDLIMIT the data returned on your screen is inaccurate. If you look for students registered in the Spring 2002 semester with a cum gpa of 3.0 or higher, using a record limit returns only the FIRST 100 records found. When you remove the RECORDLIMIT, your report will be accurate.

Using masks

When screening your data with IFs and WHEREs you are using literals. Literals are either alphanumeric characters or a date, which you enclosed, in single quotes. For example, when limiting your selection to only resident students you use the following statement:

IF RT205 EQ 'R'

RT205 is the "term residency" field. If a student is a resident, she gets an R in this field. If she is not a resident, she gets an N. R and N are literals—they are alphanumeric characters. You must enclose them in single quotes.

WHERE Masks

When you are screening data you can use the full literal or just part of it. This is called a MASK. Lets say you want all Communication Majors. Instead of writing:

WHERE RT120 EQ 'COMA' OR 'COMJ' OR 'COMD'; etc.

you can simply write

WHERE RT120 IS 'COM$';

This means the first three characters have to be COM, the last character can be anything. Since all of the Communication majors begin with COM, this would select all of them.

This could also have been written as

WHERE RT120 LIKE 'COM%';

The difference? IS 'COM$' says the character in the fourth position can be anything.

LIKE 'COM%' says only the first three characters need to be COM, the rest of the characters, can be anything. (Actually, since RT120 is only a four character field, each of the % is really only looking in one more position. If we had been looking at a field such as Name, however, the COM% would have the ability to pull all persons whose name begins with COM and anything succeeds that, like Comare, Combach, Comcade, etc.)

So, using a $ sign with WHERE IS looks in a specific position, % with WHERE LIKE allows any number of characters after the specified ones.

Likewise you could reverse this to say:

WHERE AA003 LIKE '%SMITH%';

Now you are looking the SMITH anywhere in a person's name.

What if you don't know what the first 2 characters are but the 3rd, 4th & 5th must be DAN. You would write

WHERE AA003 LIKE '_ _ DAN';

IF Masks

The $ sign can be used with IF statements or $* may be used to mark the rest of the field value as irrelevant. For example:

IF RT120 EQ ‘A$*’ (Equivalent to WHERE RT120 LIKE ‘A%’)

XIII. Exercise Set Three—Selecting Records

Exercises will be provided to you in class. The exercise set you receive is dependant upon the primary database for your office—SIS, HRS, FRS or ADS.

XIV. MFD—Master File Descriptions

Introduction

From the time a student first inquires about Marist College, to the time she enrolls, takes classes, graduates and becomes an Alumnus, information about this student is stored in IA—Marist's computer system used to maintain employee, vendor, & student information. The same is true for an employee, payroll, and accounts payable/receivable.

This section describes how this information is stored.

Fields

Let's consider the example of a student filling out a form in your office. When you identify the need for a paper FORM in your office, you decide what information will go on the form and where you will file it when it is complete.

The form may resemble:

Name: _____	Date: _____
Position Applying For: _____	Relevant Coursework:_____
Hours Available to Work: _____	

If you decided to store this information on a computer, you would look at your form and determine that you need FIELDS for each piece of information, for example a field for Name and a field for Date.

Field Definitions

You want your data to be consistently entered every time, no matter who is doing the typing. Therefore, you make certain decisions about your fields. NAME will be 32 characters long. DATE must be entered as MM-DD-YYYY (03-18-1962). POSITION APPLYING FOR is a 25 character field but there exists a list of 7 valid positions for which the person can apply.

Then you decide that NAME will be an alphabetic field. DATE will have a special DATE format (because you may have to perform calculations on it), POSITION will be alphanumeric because it could contain both alphabetic and numeric characters.

The field names you specify are sometimes short and understandable only to the person who designed the database—you. What happens when you print out these fields on a report? Maybe you should include a TITLE field that prints at the top of each column when you are reporting.

What you are doing is creating a Master File Description.

MFDs for Marist

An MFD is a Master File Description. It is a document, which describes the file. It tells you the name of the file, the key(s) which uniquely identify each record(s) in the file, the field names, field length, how data is stored, and the title which will print out at the top of each column when you print a field.

MANY files exist at Marist. Some of the IA Databases have more files than others. For instance, SIS has 26 files, while ADS has 16. You have been given the MFD(S) which corresponds to the system you have access to in IA. The MFDs, as delivered from IA, are short and sometimes not easily understood until they are explained by someone familiar with an MFD and the file you are working with. For this reason, I have supplied you with several commonly used files. Once you understand how to read an MFD, you should apply this knowledge to understanding the MFDs for other files from which you wish to access information.

The Key

A key is used to uniquely identify a record within a file. For instance, in the RTFILE, new information is entered every semester for every enrolled student. One piece of this information is the student's major. Let's assume that in Fall of 2001 a student is an English Writing major. In the RTFILE there is a record with this student's id number, his major (English Writing) and a field to identify the semester for which this was true—Fall 2001. Now it's Spring of 2002 and the student has changed his major. So there is another record in the RTFILE which has his ID, his new major (English Lit) and a field to identify the semester for which this is true—Spring 2002. What makes each record unique (so that you can pull out the one you want?). Well it's a combination of fields. Let's look at the beginning of the MFD:

Field Name	Alias	Field Format (how it prints)	Field Format (how it's stored)	Description	Title
SEGNAME = ROOT		SEGTYPE = S0			
GROUP=ROOT_KEY		ALIAS=KEY	A14	A14	KEY
RT005		A05	A05	TERM_RT	Term
RT010		A09	A09	SID_RT	Student,Number

You see two references to "keys". The first is ROOT_KEY, Alias = KEY. Then you see A14. A14 is the field's *format*. (See *Field Formats* section)

What this means is that the Primary key for this file is alphanumeric and 14 characters long (A14). If you add up the field length for the first two fields: RT005 & RT010, they add up to 14 AND each of them are alphanumeric. So two fields make up the Primary Key.

So again, what makes the two records, which exist, for the student who changed his major unique? RT005—TERM. One record has 2001F in the term field and the other has 2002S in the term field. So when you use (RT005 EQ '2002F') in your Focexec, you will get the record which says this student is an English Writing major.

Title (which prints on your report)

A Title for a field is the text, which will print out as a column heading when you print that field. For instance when you print out a student's id number from the RTFILE, at the top of that column the heading will read:

Student
Number

The title's for each field are in the last column of your MFD. If they have a comma, that means the text which follows will print on a new line. That's why the title Student Number printed on two lines as shown above.

Note: You do have the ability to change this title by using an AS phrase discussed in *Report Formatting*.

Long Field Names and Alias

There is another column on this MFD. It contains data like: SID_RT, , etc. These are called long field names. In SIS and FRS, long field names are just a description field. You do not use them. They may just help you to identify what a field is used for. However, in HRS and ADS long field names are actually used as fields. If you are an ADS or HRS user, long field name is in the first column (Field Column) of your MFD and the cryptic field names like LC721 or EAFHB are used as their descriptions. So, no matter what IA system you report from, you will always use the fields, which are displayed in the first column of your MFDs when you are writing your Focexec. If you use SIS or FRS, then these will look like: RT010, FS042, etc. If you are an ADS or HRS user, these fields will look like: ANN_SLRY or CMPGN_CD_LH, etc.

An alias would be similar to a "nickname" for a field—something you could use instead of using a field name. If your file has an alias for a field, it would display in the second column of the MFD. Presently, we do not have aliases for fields so they are not used. The only time they are used is for the KEYS and SUBKEYS for files. In this instance, you will see ALIAS=KEY in the MFD. You may use this alias.

Even if aliases were present for your MFD, Marist does not encourage you to use them because when you create your report you will be joining files together and creating hold files. These aliases cannot be used in hold files.

Field Formats

The third and fourth columns on your MFD contain the field formats. The first one is how the field is used the second is how Focus stores the data. Focus fields are either alphanumeric, numeric, or date fields. The format tells you what kind of data the field can contain, how many characters the field can contain, and most importantly, how to display the field. Both alphanumeric and date fields are left justified when they print. Numeric fields are right justified.

If The Format Is	The Field Is	It Can Contain	It Can Have
A	Alphanumeric	Alphabetic and numeric characters and special characters, including conventional dates (See *Conventional Dates* section.) Even though some alpha-numeric fields contain numbers, you cannot use alphanumeric field value in calculations.	1 to 256 characters
I, F, D, or P	Numeric	Numeric characters only, including conventional dates; for use in calculations I=whole number D, F, and P can indicate different decimal number formats.	I: 1 to 9 positions D: 1 to 15 positions P: 1 to 15 positions F: 1 to 9 positions
YYMD, MDYY, etc.	Date	Date value stored as integers	Not applicable

Formats for Decimal Numbers

Formats for decimal numbers contain two sets of numbers separated by a decimal point. The two numbers indicate the maximum length of the field and the number of digits following the decimal point. For example, D15.6 indicates that the full field length is 15. Of these 15 positions, 6 positions are reserved for decimals and one position is reserved for the decimal point. Thus, eight is the maximum number of digits possible to the left of the decimal point. (15 - 6 - 1 = 8)

How Focus stores and displays numbers

Focus supports four different numeric formats:

- I for Integer
- F for floating point decimal, single-precision
- D for floating point decimal, double-precision
- P for packed decimal

The format determines whether the number can be a decimal, the maximum field length, and how Focus stores and displays the number.

Formats F and D are rounded for display but the full number is stored.

With I and P, first the number is rounded and THEN the number is stored.

Pay special attention to a numeric format of a field if you will be performing calculations because this could greatly affect the result of your calculation.

Dates and Date Formats

Focus provides two date formats

- Date formats (*smart dates*)—not used at Marist
- Numeric or alphanumeric formats with date edit options (*conventional dates*)

Conventional Dates

A date is a conventional date if the field format looks like: A08YYMD, or I8YYMD. It has a specified format and field length up to 8 bytes. See chart below for the various date components:

D	Day—a number from 1 – 31
M	Month—a number from 1 to 12
Y	Year—the last two digits of the year
YY	Year—The four digit year

You can determine how your date will display by looking at the MFD and referring to the chart below:
In previous versions the system date was only displayed as MM/DD/YY format. Now you can have any valid component format after the &DATE variable however, you may not use a format that attempts to change the component delimiter from slash (/) to dash (-), or dot (.) or remove the slash via the vertical bar (|).

The syntax for the new function is:
&DATEformat where format is:

Format	**Description**	**Output**
YMD	YY/MM/DD	98/01/21
YYMD	YYYY/MM/DD	1998/01/21
MDYY	MM/DD/YYYY	01/21/1998
Q Calendar	Quarter number	Q1
JUL	Julian date YYDDD	98/021
WtMtDYY	Weekday Month day, year WWW MM/DD/YYYY	WED, JAN 21 1998
WrMtrDYY	Weekday, Month, day, year format	WEDNESDAY, JANUARY 21 1998
MtrDYY	Month, day, year format	JANUARY 21, 1998

An example would be:
-TYPE TODAY IS &DATEMTRDYY

You can perform calculations with dates. To calculate conventional dates you need to define a new field. For example:

Subtraction: New field/format = YMD (first date, second date)
MDY
DMY

When subtracting one conventional date from another make sure that the first date is the earlier of the two dates. So, if you subtract Jan 1 from Jan 31, you would write:

```
Define File xxxxx
  Jan/I4=YMD(960101,960131)
```

Recurring Data/Segments

Sometimes, certain fields are repeated within each record. For instance, in the RAFILE, a student's GPA could exist twice in RA380 because he could have pursued both his undergraduate and graduate degrees at Marist. How can you have two different values in a field? Well you can if you store this information in a SEGMENT, which refers back to a field(s) in the ROOT segment. What does this mean? Well take another look at the MFD on the second page of this section. The very first line reads: SEGNAME=ROOT. This is the ROOT segment. Information contained in the ROOT is unique. When you need to specify a field, which might have more than one value, you separate it from the ROOT and put it in another SEGMENT. The RAFILE has a segment titled: RA040SG:

SEGNAME=RA040SG	SEGTYPE=S0	PARENT=ROOT	OCCURS=RA030	
RA305	A02	A02	CAREER_STU	Stu,Crer
RA310	A05	A05	LAST_TERM	Stats,Last,Term
RA315	P06.2	P06.2	CUM_ABA_RES_CREDIT	Cum ABA, Res Cred
RA318	P03	P02	LEGAL_WRITING	Cum Legal, Writing
RA319	P09.2C	P04	CUM_ATTEMPTED_HRS	Cum,Hours,Attempted
RA320	P09.2C	P04	CUM_EARNED_HRS	Cum Hours,Earned
RA325	P09.2C	P04	CUM_QHRS	Cum,Quality,Hours
RA330	P09.2C	P04	COLL_HIST_QPTS	Cum,Quality,Points
RA375	P09.2C	P04	CUM_TRANSFER_HRS	Cum,Transfer,Hours
RA380	P08.3	P04	COLL_HIST_GPA	Cum GPA
RA381	P07.2	P03	CUM_CALC_BASIS	Cum,Calc,Basis
RA040_ORDER	ALIAS=ORDER	I03	I04	

Look at the first line. It reads: PARENT=ROOT, OCCURS=RA030. This means it is a child of the root segment. You can identify it as a recurring segment because it reads: Occurs: RA030. RA030 is a counter field. It keeps track of how many segments exist for each student/record. If RA030 equals two and you print out RA380, you will get TWO GPAs!.

So how do you get the GPA you want? You make sure you use RA305 and ask for the appropriate career level. For instance to obtain a student's undergraduate GPA be sure to include RA305 EQ 'U' in your Table Request.

What if the student has two GPA's and you want the first one? You can use the last field displayed in this segment—RA040_ORDER (Alias = ORDER). This field keeps track and numbers each occurrence. The first GPA is "1", the second is "2" and so on. So if you write the following in your exec:

```
PRINT RA005 RA030 RA380
IF ORDER EQ 1
```

Focus will look at the student's RA040SG segment. Regardless of how many GPA's the student may have it will return the one numbered "1" to you. This can be a useful tool when you are working with statistics or historical data.

XV. Joining Files

Determining you need to join files

When the report you are attempting to create has fields in two or more different files, then you need to join these files. For instance, if we report from the RTFILE we can get information such as the student's id number, their class year (junior, senior, etc), whether they're an undergraduate or graduate and other like information. We cannot print the student's name from the RTFILE. If we want the student's name on the report, we need to join the RTFILE to the AAFILE (which contains student names).

The "How Tos" of Joining

To join two files we must first determine that they CAN be joined. You can do this by looking for common fields between the two files. By common, we mean their field formats. For example, if the key in one file is alphanumeric, 9 characters long (and contains an ID number), then you must identify a field in the next file that has the same FIELD FORMAT (but not necessarily the same name). The field formats must be identical.

Look at the two fields that you wish to join. One of them must be the KEY or the first part of the key in one of the files. For instance, two files which you CAN join:

RTFILE

SEGNAME=ROOT	SEGTYPE=S0			
GROUP=ROOT_KEY	ALIAS=KEY	A14	A14	
RT005	A05	A05	TERM	RT,Term,Code
RT010	A09	A09	STU_ID	Student,Number
DEFINE SUBKEY_RT/A9 = RT010	A09	A09		Title='Key'

AAFILE

SEGNAME=ROOT	SEGTYPE=S0			
GROUP=ROOT_KEY	ALIAS=KEY	A09	A09	
AA002	A09	A09	STU_ID	"Student,Number"
AA003	A32	A32	STU_NAME	"Student Name"

The first file depicted is the RTFILE. (You are only seeing the very beginning portion of this file.) The second is the AAFILE. Again, you are only seeing a portion of the file. In looking at these two files we see that the *subkey* in the RTFILE is equal to the *key* in the AAFILE. Both are A09. The rule says that to join two fields "one of them must be the KEY or the first part of the key in one of the files." In the AAFILE we are talking about the KEY, so we have determined that we can now join the files.

The very first file you specify in your JOIN statement will become the HOST file. The second file you reference in your JOIN statement will become the CROSS-REFERENCED file. If the files being joined have identical keys, it doesn't matter which file is the host and which one is the cross-referenced. Given this situation (keys are identical) usually the host is the larger file or the one which will most of the fields will come from on your report.

The cross-referenced file is the one which **MUST** have the common field(s) as either it's key or the "first part" of the key. So look at the two files again. Which one has the common field(s) as it's key or the first part of it's key? It's the

AAFILE. So we now know that the RTFILE must be the host file and the AAFILE must be the cross-referenced file. Our join must join the RTFILE to the AAFILE.

Subkey_RT, is no longer a standard field of the MFD, it has been added to be compliant with the previous version of SIS (88.1). In the above example the field RT010 could easily be joined to the key of the Aafile, since both are a field format of A9.

Creating a HOLD file before a JOIN

The RTFILE, as of the date this manual was published, contains upwards of 170,000 records. The AAFILE contains approx. 84,000 records. You DO NOT want to join these together without first extracting as much information as you can from the RTFILE. This will eliminate manipulating thousands of records we don't want.

The most efficient way of creating this join is to first consider what criteria will you specify for the RTFILE? What fields will you want to work with in your report?

Let's assume we want the current semester, students who are currently registered, only undergraduates, their class year, and whether or not they are a resident. The fields we wish to printout include id number, class year, and resident/non-resident.

Before doing a JOIN we extract this information from the RTFILE as follows:

```
TABLE FILE RTFILE
PRINT RT14C RT205
BY SUBKEY_RT
WHERE (RT005 EQ '2002S') AND (RT020 GE 12) AND (RT14E EQ 'U');
ON TABLE HOLD AS STUDS
END
```

or BY RT010

We print class year, residency status

BY SUBKEY_RT or BY RT010

> Note: It's important that we carried the SUBKEY_RT or RT010 into our hold file because this is what we will be joining to the AAFILE.

WHERE TERM is equal to Spring 2002, CURRENT ATT. HOURS is greater than or equal to 12 and CAREER is equal to undergraduate.

Then we save this file as a HOLD file called STUDS

Please Note: The filename above—STUDS is an example of a file name. You can name your hold file anything you'd like provided you limit yourself to eight characters beginning with an alphabetic character. For instance valid names include: FALL2002, SPRING2002, etc. It is recommended to give a logical name to the hold file.

Now we can write our join statement:

JOIN SUBKEY_RT IN STUDS TO KEY IN AAFILE AS J1

Or

JOIN RT010 IN STUDS TO KEY IN AAFILE AS J1

> Note: AS J1 is optional but I.T. recommends you name your joins. When the same file is used in multiple joins, the AS and unique JOINNAME is required so that one JOIN does not overwrite one another.

Now let's look at our new Focexec so far:

```
TABLE FILE RTFILE
PRINT RT14C RT205
BY SUBKEY_RT
WHERE (RT005 EQ '2002S') AND (RT020 GE 12) AND (RT14E EQ 'U')
ON TABLE HOLD AS STUDS
END

JOIN SUBKEY_RT IN STUDS TO KEY IN AAFILE AS J1

                or
JOIN RT010 IN STUDS TO KEY IN AAFILE AS J1
```

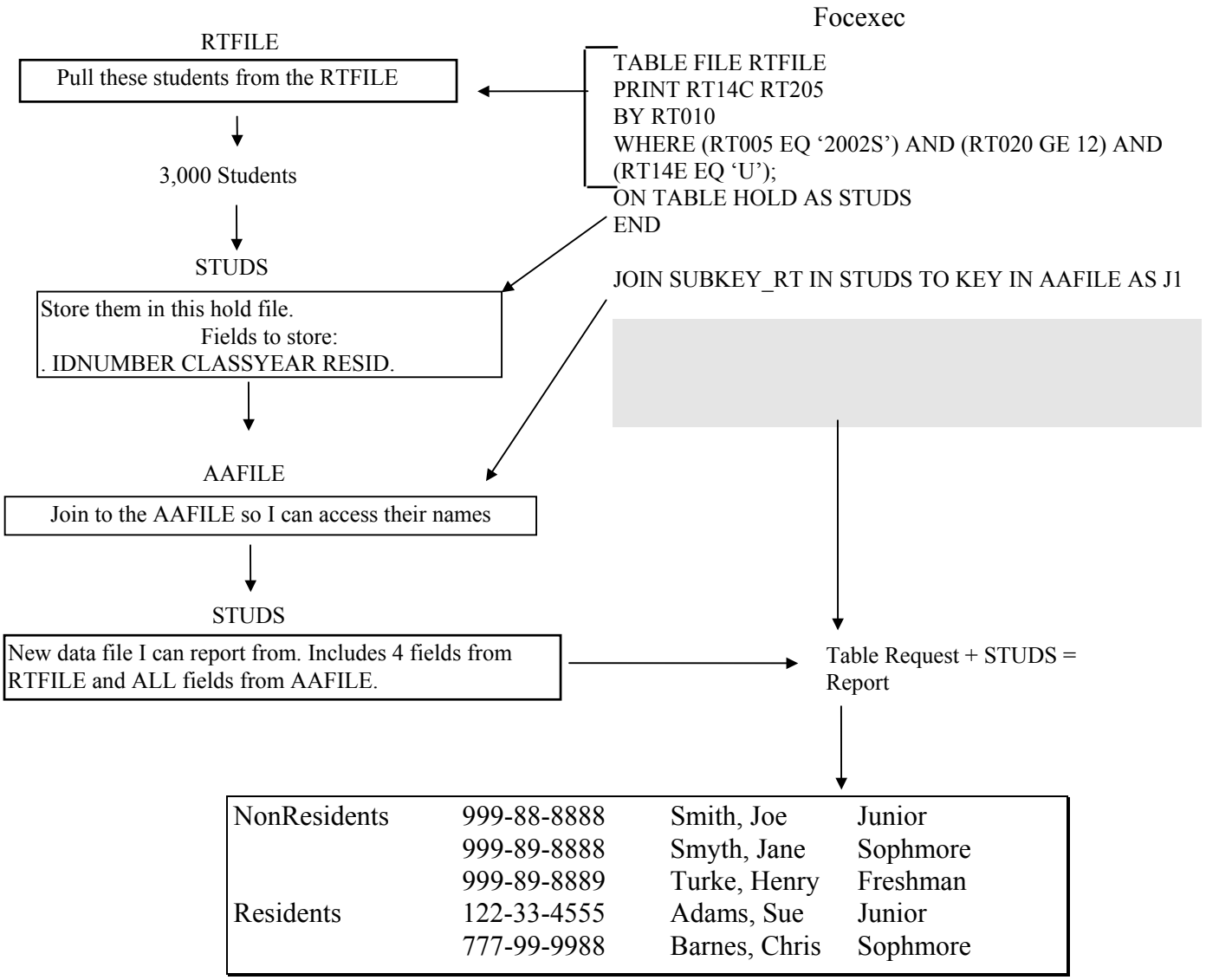
or RT010

After we extract only the records WE NEED from the RTFILE, we join it to its matching records in the AAFILE. We are working with a much smaller data set here than if we had tried to join the ENTIRE RTFILE to the ENTIRE AAFILE.

This new file we have created called STUDS has it's own MFD! We have created a new file available during THIS Focus session from which we can report. So now let's issue another TABLE FILE Request:

```
TABLE FILE RTFILE
PRINT RT14C RT205
BY SUBKEY_RT
WHERE (RT005 EQ '2002S') AND (RT020 GE 12) AND (RT14E EQ 'U')
ON TABLE HOLD AS STUDS
END

JOIN RT010 IN STUDS TO KEY IN AAFILE AS J1

TABLE FILE STUDS
PRINT AA002 AA003
BY RT205
END
```

Notice how we can now use any of the fields we *selected* from the RTFILE and ALL of the fields we matched to in the AAFILE. It's important to understand what fields are available to you at this point. In the above example we first pulled information from the RTFILE. We only pulled TWO FIELDS! These are:RT010, & RT205. You then join these four fields of information about each student to all of the fields for each student in the AAFILE. This means any fields in the AAFILE are available to you but only four from the RTFILE. After you type TABLE FILE STUDS toward the bottom of your focexec, you can use verbs with only the fields available to you. Understanding this concept will assist you in creating the hold files you need before your joins.


Focexec
RTFILE
Pull these students from the RTFILE
3,000 Students
STUDS
Store them in this hold file.
Fields to store:
. IDNUMBER CLASSYEAR RESID.
AAFILE
Join to the AAFILE so I can access their names
STUDS
New data file I can report from. Includes 4 fields from RTFILE and ALL fields from AAFILE.
TABLE FILE RTFILE
PRINT RT14C RT205
BY RT010
WHERE (RT005 EQ '2002S') AND (RT020 GE 12) AND (RT14E EQ 'U');
ON TABLE HOLD AS STUDS
END
JOIN SUBKEY_RT IN STUDS TO KEY IN AAFILE AS J1
Table Request + STUDS = Report
NonResidents 999-88-8888 Smith, Joe Junior
999-89-8888 Smyth, Jane Sophmore
999-89-8889 Turke, Henry Freshman
Residents 122-33-4555 Adams, Sue Junior
777-99-9988 Barnes, Chris Sophmore

Define Based Join

Not all JOINs are this easy. Sometimes, there exists no common fields in the KEY between two files. When this occurs you need to create a field. This is called a DEFINE-BASED JOIN.

This field can be created permanently in the MFD, or can be created within a FOCEXEC by a DEFINE statement. If you are creating a field in a DEFINE command, you must issue the DEFINE after the JOIN command but before the TABLE REQUEST. A JOIN command clears all fields created by DEFINE command for the host file and the joined structure.

Let's look at the RCFILE and the RTFILE

RTFILE

SEGNAME=ROOT	SEGTYPE=S0			
GROUP=ROOT_KEY	ALIAS=KEY	A14	A14	
RT005	A05	A05	TERM	RT,Term,Code
RT010	A09	A09	SID_RT	Student,Number

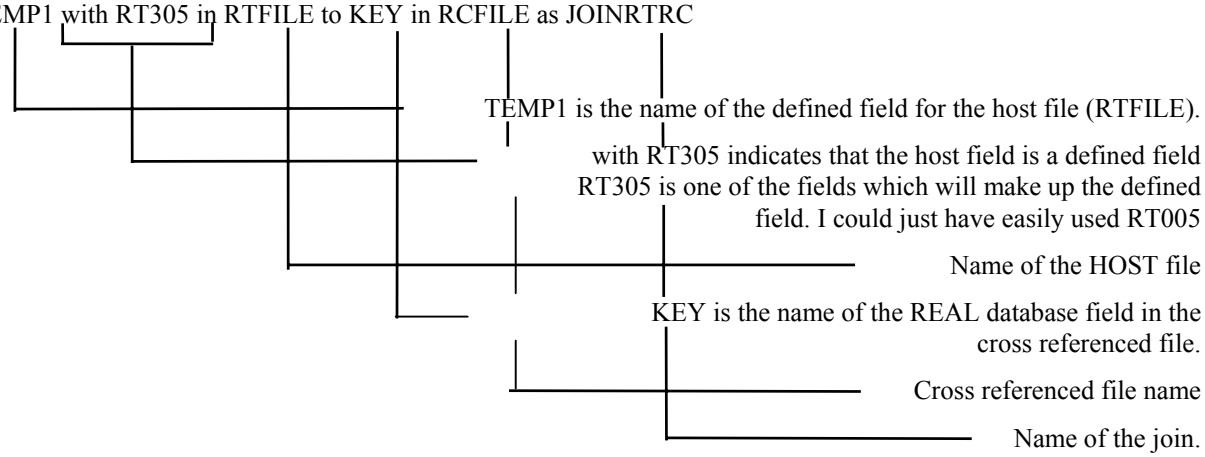
SEGNAME=RT020SG	SEGTYPE=S0	PARENT=ROOT	OCCURS=RT296	
RT301	A01	A01	SPE_STATUS	Stat
RT305	A11	A11	SECTION_ID	Crs sect ID

RCFILE

SEGNAME=ROOT	SEGTYPE=S0			
GROUP=ROOT_KEY	ALIAS=KEY	A16	A16	
RC005	A05	A05	TERM	'Term'
RC010	A11	A11	Section_ID	'Crs Sect ID'

We have to use the key from one of these files. The key in RC includes Term and Course ID. We have the same fields in RTFILE but they are not in the same order. So, we will create a new field called TEMP1. (This new field will "join" the fields together in the proper order.):

JOIN TEMP1 with RT305 in RTFILE to KEY in RCFILE as JOINRTRC



Now we need to define this new field:

```
DEFINE FILE RTFILE
TEMP1/A16=  RT005 | RT305
END

TABLE FILE RTFILE
. . .
```

Important:

You must issue the DEFINE after the JOIN but before the TABLE request.

Here is our Focexec up to this point:

```
JOIN TEMP1 with RT305 in RTFILE to KEY in RCFILE as JOINRTRC
DEFINE FILE RTFILE
TEMP1/A14=  RT005 | RT305
END

TABLE FILE RTFILE
. . .
```

Multiple Joins

Sometimes it is necessary to join more than one file. For instance you may want student name from the AAFILE, term information from the RTFILE, and career statistics such as GPA from the RAFILE.

You can include up to 16 join statements for any combination of files. These joins commands are active only for the current Focus session or until the JOIN is removed.

The following would join AA, RT, and RAFILEs.

JOIN RT010 in RTFILE to KEY in AAFILE as JOIN1

JOIN RT010 in RTFILE to KEY in RAFILE as JOIN2

To make sure you successfully joined the files, issue a CHECK command:

CHECK FILE command:

```
CHECK FILE RTFILE PICT
```

The example of joining RT, AA, and RP would work but, again, you are joining everything to everything. This would take an enormous amount of processing time. So first retrieve only the records you need from the RTFILE. Then join this hold file to the AAFILE (*Creating a HOLD file before a JOIN* in *Joining Files* section of this manual*.)*

Then join this smaller "RTFILE" to the RAFILE.

Unique vs. Non-unique joins

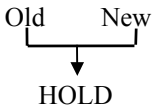
When one record in one file matches only one record in another file then you are creating a unique join. (The information is unique in each file). Sometimes, however, you have one record for someone in one file (AAFILE) but have multiple records for this person in another file (RTFILE). This is called a NON-UNIQUE join. The syntax for writing this join is slightly different. It would read:

JOIN fldname in HOSTFILE name to ALL fldname in CROSSREF-FILE as JOINNAME

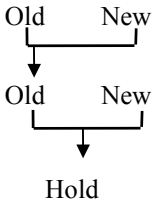
The MATCH Command

Match is another method for extracting and creating datasets in IA. It is commonly used when a JOIN would not be possible. Generally, it works like this: you write one set of commands to retrieve information from one file, then write commands to retrieve information from a second file. Focus then compares information between the two files based on relational operators, BY fields, and verbs. The result is written to a temporary file until it reaches the END of your request. The final records are saved to a hold file - (Something which you can then REPORT from!).

The MATCHing, which occurs, depends on the concept of "old" and "new" files. Old refers to the first file named in the request and new refers to the second file. The result of EACH match creates an old file until the END statement is reached. So if you merge two files the following happens:



If you were matching three files:



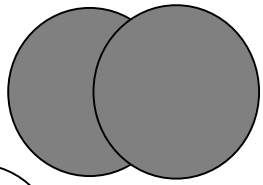
The *relational operator* in these two examples was OLD-OR-NEW. This means that records found in BOTH the first file (old) plus the records from the second file (NEW) will be placed in the hold file. This is just one set of relational operators. Here are some others:

AFTER MATCH HOLD :	OLD-OR-NEW
	OLD-AND-NEW
	OLD-NOT-NEW
	NEW-NOT-OLD
	OLD-NOR-NEW
	OLD
	NEW

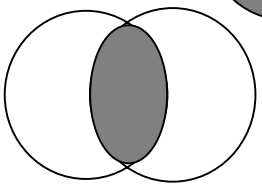
If you do not specify a hold file name then "hold" will be used as the name. The default relational operator is OLD-OR-NEW if you do not specify an AFTER MATCH line.

Below is a graphical representation of the data you end up with in your hold file. The shaded area represents the final extracted data:

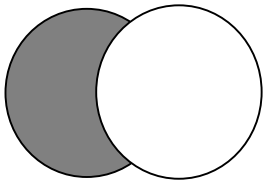
OLD-OR-NEW — All the records from **both** the old and the new file are placed in the HOLD file.



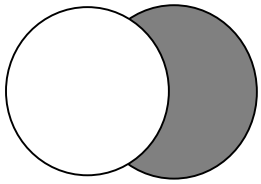
OLD-AND-NEW — Records that appear in BOTH the old and new files go to the HOLD file.



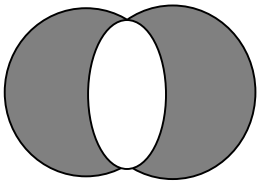
OLD-NOT-NEW — Records that appear only in the OLD file will appear in the HOLD file.



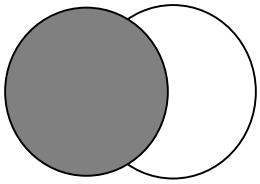
NEW-NOT-OLD — Records that appear only in the NEW file will appear in the HOLD file.



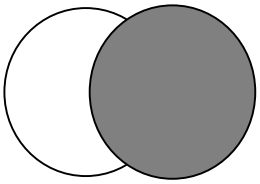
OLD-NOR-NEW — Only records which are in the old but not in the new file, or in the new file but not in the old will appear. (The complete set of non-matching records.)



OLD — Only records from the old file will appear in the HOLD file.



NEW — Only records from the new file will appear in the HOLD file.



Remember we said that the match depends on relational operators (above), BY fields, and verbs. Using a BY means you are sorting. The first BY statement you use is called a high-order sort. Both the At least one pair or "high-order" sorts is required.

It is also particularly useful when you need to compare information between two files where the information may or may not exist in one of the files.

Generally, to perform a match, the files much have a data field with the same format in common.

With Match processing, you access one file and select the first set of fields you want to use, sorted BY the field you will be matching. ACROSS, WHERE TOTAL, and COMPUTE statements are not allowed in a MATCH command. The results are stored in a temporary HOLD file. Focus will tell you how many records were held.

MATCH FILE file1	This is the old file
PRINT AAA BBB CCC DDD BY field RUN	use the RUN command instead of END so that Focus knows that you want to continue processing

The next file is accessed, the fields that are required from this file are specified, and the file is sorted by the field you will match on. The results of this selection are also stored in a temporary HOLD file. Focus will tell you how many records were held.

FILE file2	This is the new file
PRINT EEE FFF GGG HHH BY field AFTER MATCH HOLD (as name) criteria	Criteria for the match: OLD-or_new (default) OLD-AND-NEW OLD-NOT-NEW NEW-NOT-OLD OLD-NOR-NEW OLD NEW

The results of the two HOLD files are compared and MATCHed according to the criteria specified, and the result is stored in another HOLD file. Focus will tell you how many records were matched.

Match processing generates a single segment HOLD file. The resulting HOLD file can be used just like any other file. Like any other HOLD file, the first field in the HOLD files has been assigned the alias E01. The contents of the HOLD file can be printed by issuing the following:

```
TABLE FILE HOLD
PRINT SEG.E01
END
```

In the following example we will use MATCH to pull anyone who had housing in either the Fall of 2001 OR the Spring of 2002. We will then compare the results to anyone who is registered for the Fall 2002 semester who does not have housing and has at least 3 credit hours. We will use AFTER MATCH HOLD AS FILENAME OLD-AND-NEW. This means we want only those people who have records in BOTH sets of data. Refer to the chart on the previous page for an explanation of OLD-AND-NEW.

```
MATCH FILE RTFILE
BY RT010
WHERE (RT005 EQ '2001F' OR '2002S') AND (RT185 NE '  ')
RUN

FILE RTFILE
BY RT010
WHERE (RT005 EQ '2002F') AND RT185 EQ '  ') AND (RT020 GE 3)
AFTER MATCH HOLD AS HOUS OLD-AND-NEW
END

JOIN RT010 IN HOUS TO KEY IN AAFILE AS J1

TABLE FILE HOUS
PRINT AA002 AA003
END
```

HOLD files

We first discussed HOLD files in some detail when we joined the RTFILE to the AAFILE in *Creating a HOLD file before a JOIN* in the *Joining Files* section of this manual. This is one use of a hold file. You could also create a hold file to be downloaded to a personal computer. Here are several reasons for using HOLD files:

1. Large Files: To save processing time, and conserve on the space required to manipulate large files, you will want to extract only the information that you need to process from the large files. This is what we did when we created a hold file from the RTFILE before joining it to the AAFILE.
2. Sorting of COMPUTEd fields. We have not discussed these fields yet but basically they are calculations you perform on the fly. Since COMPUTEd fields are generated at report time, they cannot be sorted in the TABLE request that created them. So to sort on them, you first save them in a HOLD file and then issue a second TABLE requests where you can sort the field.
3. Sorting SUMmed and COUNTed fields. We HAVE discussed SUMming and COUNTing in some depth! Counting and adding happen during the execution of the TABLE request, therefore the results are not available for sorting. Similar to COMPUTEd fields, you can use a hold file to store the results of these requests and sort them in subsequent TABLE requests.
4. Converting multi-segment IA files to Focus single segment files

There are two ways to create a hold file:

1. If you want to see the report printed, issue the HOLD command after the TABLE request.

```
TABLE FILE RTFILE
PRINT RT010
BY RT005
END
HOLD as studs
```

2. If you don't need to see the report, you can issue the hold as part of the table request.

```
TABLE FILE RTFILE
PRINT RT010
BY RT005
ON TABLE HOLD as studs
END
```

The HOLD file contains all the data fields from the report. If you want to see the MFD for this hold file, you can issue the command TED HOLD MASTER.

Focus stores the data from the TABLE request in that file. Heading, Column titles, footings and page numbers are not stored. So if you want any of these, create them in your last TABLE REQUEST when you are actually using the HOLD file to report from (that's if you are not downloading it).

Save Files

The SAVE command selects your data from a file you specify (RTFILE for example) and stores it in a flat file. Unlike HOLD, it DOES NOT create a MFD. There are no headings, title lines, or subtotals in your SAVE file, just RAW data. The fields are saved one next to another with no DELIMETER between fields. This make the SAVE command ideal for extracting data for maintenance transactions or for downloading to a PC.

The format of the SAVE command is:

ON TABLE SAVE AS NONPRINT

If you are downloading the file, you will want to add the field delimiters yourself between the printing of each field. You can do this by DEFINEing a field that prints a field delimiter—a semi-colon for example:

```
TABLE FILE RTFILE
PRINT RT120
BY RT010IF RT020 GE '12'
IF RT14C EQ 'FR'
IF RT005 EQ '2002S'
ON TABLE HOLD AS FROSH
END

JOIN RT010 IN FROSH TO KEY IN AAFILE AS J1

DEFINE FILE FROSH
SUBDEL/A1 = ',';
DELIM/A1 = CTRAN (1,SUBDEL,107,94,DELIM);
END

TABLE FILE FROSH
PRINT AA002 DELIM AA003 DELIM RT120
ON TABLE SAVE AS NONPRINT
END
```

Although effective, you will probably want to modify the above procedure to concatenate the delimiter to each field, then print the new field name. For instance, create a new field which concatenates AA002 with the DELIM field and print the new field name. You will eliminate unnecessary spaces in your downloaded file with this method.

Refer to section entitled *Printing* for information regarding how to submit for a NONPRINT job.

More on SAVE Files

The information supplied above for creating a raw data file is effective but more options exists which you may wish to make use of.

Options for the ON TABLE SAVE command:

ON TABLE SAVE AS NONPRINT FORMAT *DIF*
WP
LOTUS
SYLK

In the example above the options in italics are listed vertically. You would choose ONE of these options:

DIF	Captures the entire report output including headings, footings, etc., and creates a file which can be easily incorporated into most Calc-type packages. The DIF file is an EBCDIC file.
WP	Captures the entire report output including headings, footings, subtotals, etc., and creates a file which can be easily incorporated into most word processing packages. The WP file is an EBCDIC file. NOTE: Carriage controls are not generated in the file. In order to produce them, specify the TABPAGENO option in a heading or the SET PAGE=OFF command.
LOTUS	Captures all the columns of the Focus report request. All alpha fields are enclosed in quotation marks. Each column is separated by commas. The LOTUS file is an EBCDIC file and has a fieltype of PRN in CMS, and allocates a scratch dataset to the file HOLD.
SYLK	Captures all the columns of the report request for MicroSoft's spreadsheet program Multiplan. The SYLK file is an EBCDIC file.

XVI. Exercise Set Four—Joining Files

Exercises will be provided to you in class. The exercises you receive will be representative of your everyday reporting needs.

XVII. Report Formatting

In *Organizing Report Data* we discussed two different ways of organizing your report—BY and ACROSS. These two sort options change the way your output looks. In this section we will look at enhancing your report by adding header and footers, separating report sections, and renaming column headings.

Headers and Footers

Three kinds:

- Report headings and footings—text which prints only at the START and END of a report
- Page headings and footings—text that prints at the top and bottom of EACH PAGE
- Section headings and footings—text that prints at the start and end of a SECTION (created by sort statements)

Report Headings and Footings

Report headings and footings appear once on each report. This type of heading is created using SUBHEAD for a heading and SUBFOOT for a footing.

The subhead will display only on the first page of your report. You can use a report heading to explain the report to the recipient for whom the report was created. For example,

```
ON TABLE SUBHEAD
"Marist College"
"Fall 2002 Communications Majors"
"Produced on DATE"
```

ON TABLE SUBHEAD is used in your last TABLE REQUEST. This means if you have joined files together to select your data, include your SUBHEAD command in the last TABLE REQUEST of your Focexec. ON TABLE in the above example means that the heading applies to the entire report (table), and not just to a portion of it.

To include a footing at the end of the report you would write:

```
ON TABLE SUBFOOT
"Please direct any questions about this report to NAME at EXTENTION."
```

As with the SUBHEAD statement, the SUBFOOT is used in your last TABLE REQUEST. The subfoot in the example above will print after the last line of the report has finished printing.

Your ON TABLE SUBHEAD and SUBFOOT phrases may be placed anywhere in your TABLE REQUEST. In the example below, they have been placed after the verb phrases.

```
TABLE FILE RTFILE
PRINT RT010 RT120
ON TABLE SUBHEAD
"Marist College"
"Fall 2002 Communications Majors"
"Produced on DATE"
ON TABLE SUBFOOT
"Please direct any questions about this report to NAME at EXTENTION."
```

When your report prints, there will be no space between your heading and the start of your report or between the footing and the end of your report. You can change this to separate them from the body of your report by using either of these options:

1. Add one or more blank lines after the report heading or before the report footing

2. To print the report heading and/or footing on a separate page, either as a cover sheet or as a final page for the report.

We will use both options for the report heading. First we include the following line to insert a page break after the heading:

```
ON TABLE PAGE-BREAK AND SUBHEAD
```

Then we decide to skip 20 lines before the heading so it will print toward the middle of the first page (we use <20/) for this. Then, for each line, we decide what column to begin printing in (we use <## for this):

```
"</20 <30 Marist College"
"<25 Fall 2002 Communication Majors"
"<30 Produced on DATE"
```

Here is our finished heading:

```
ON TABLE PAGE-BREAK AND SUBHEAD
"</20 <30 Marist College"
"<25 Fall 2002 Communications Majors"
"<30 Produced on DATE"
```

Notice that the text of the heading is typed in upper and lower case. This is how we want it to look on the report. To do this, type the command CASE M in the TED editor before typing the heading. (Press Enter)

Since our report footing is only one line, we will keep it on the same page but will skip two lines after the last report line by inserting </2 at the start of the footing text line:

```
ON TABLE SUBFOOT
"</2 Please direct any questions about this report to NAME at EXTENTION."
```

Page Headings and Footings

Page headings and footings print on every page of your report. At Marist we require that you include page headings in every exec you create as follows:

```
HEADING CENTER
"Marist College Confidential"
"Descriptive title of report"
"Name of Focexec          &DATE   </2"
```

Including these lines on every page of your report insures: the reader understands information contained in report is sensitive, tells them what is contained in the report, the name of the exec should they need to see the source and indicates the date of the data. If several pages of the report should become separated from the entire report, these identifying lines become even more critical.

&DATE in the above example will print the current date (date report was run). We included the </2 at the end of the heading to skip two lines between the heading and the body of the report.

Focus left-justifies headings so we include HEADING CENTER in our statement to print the heading in the center of the page. You can also use a footing in your report as follows:

```
FOOTING CENTER
"Produced by:  YourName"
```

Just as sub-foot prints directly under the last line of data on your report, so does the footing. You can force the footing to the bottom of every page:

```
FOOTING BOTTOM CENTER
"Produced by:  YourName"
```

In testing the printing of footers and headers at Marist we have found the following:

1. If you include a FOOTING BOTTOM CENTER in your exec and print to the laser printer, your output will print as it should.
2. If you include FOOTING BOTTOM CENTER in your exec and print to CENTRAL, the footing is too close to the bottom of the page and your report will not print as it should. Here is a work-around when printing to CENTRAL:

```
FOOTING BOTTOM CENTER
"Produced by:  MyName"
" "
" "
```

This will print several "blank" lines under my footing and the report will print properly.

Separating Sections

Let's assume we are producing a report, which prints ID Name and GPA of students by Major. Each time the major changes we want to visually separate that section by including an underline when the section ends:

```
BY RT120 UNDER-LINE
```

What if we wanted to add a sub-title every time the major changes such as COMC Major? We could include the following:

```
BY RT120 SUBHEAD
"<RT120 Primary Major"
```

In the example above you are sorting on RT120 (Major) and then each time the major changes, you are including a subheading which reads the four letter abbreviation which prints that major and the "words" Primary Major. However, in the example above the "Major" would print twice. To eliminate this problem, you can use NOPRINT as follows:

```
BY RT120 NOPRINT SUBHEAD
"<RT120 Primary Major"
```

If you had several BY statements in your request you could add

```
BY FIELD SKIP-LINE
```

to the last BY statement so each time this field changes, Focus would skip a line before beginning the next section. If you did not have several BY statements in your request but wanted to skip a line between the subhead and the beginning of the data you could:

```
BY RT120 NOPRINT SUBHEAD
"</2 <RT120 Primary Major"
```

You can include field names and/or totals for numeric data in subfootings as well. For example:

```
BY RT120 SKIP-LINE SUBFOOT
"Total credits for <RT120 <ST.RA380"
```

These two lines will sort on major, skip a line when the major changes, and then print a subfoot which prints Total for ENSC major: Sum of total credits . Using totals in footings makes more sense when you are using numeric data as would the Business Office or the Financial Aid Office.

Limits for headings and footings

You can use up to nine different headings and footings with a maximum of 57 heading and footing text lines in a single report.

Please note: If you use a *report* footing and a *page* footing in the same report, Focus prints the report footing on a separate page.

Specifying Column Positions

With sub-heads, sub-foots, and with fields, you can specify what column you want this information to print in:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 NOPRINT SUBHEAD
"<9 <RT120 MAJOR"
```

<9 begins printing the SUB-HEAD in the 9th column.

You can do the same with fields. When Focus prints your data, it uses the full field length for each field value skips two spaces and begins printing the next full field length. For instance, this request:

```
TABLE FILE AAFILE
PRINT AA002 AA003 AA010
BY AA005 NOPRINT
END
```

produces the following report:

Student Number	Student Name	Birth Dt
001122334	Adams, Sue	1976/06/13
002233445	Adrown, Kris	1976/07/16
034455667	Allen	1977/07/16

Notice how much space is allocated for printing out the student's name. This is because this field is defined as 32 characters long. Even though none of the names in our report take up 32 spaces, that many spaces are still allocated. To increase the readability of our report, let's move the Birth Date closer to the Student Name:

```
TABLE FILE AAFILE
PRINT AA002 AA003 AA010 IN 35
BY AA005 NOPRINT
END
```

We have added IN 35 to the request to tell Focus we want AA010 which is birth date to print in column 35. In heading and footings we specified the column by using a left caret. When printing fields in our report, however, we use IN.

Let's look at the modified report:

Student Number	Student Name	Birth Dt
001122334	Adams, Sue	1976/06/13
002233445	Adrown, Kris	1976/07/16
034455667	Allen	1977/07/16

Another method for specifying a column in which to print a field is to use +N where N is a number. For example:

```
TABLE FILE AAFILE
PRINT AA002 AA003 AA010
BY AA029 IN +5
END
```

This prints AA029 5 spaces in from where it would normally print. This is a nice way to indent.

Suppressing the Printing of a Column

Sometimes you wish to suppress the printing of a column. This is especially true when sorting. For instance, you may wish to print ID, NAME, MAJOR. You wish to sort on name but you don't want the name printing twice. To suppress the printing of this column, you will use NOPRINT:

```
Let's assume you have joined the RT to the AAFILE.

TABLE FILE HOLDNAME
PRINT AA002 AA003 RT120
BY AA005 NOPRINT
END
```

AA005 is a special sort field. You would want to print AA003 as the name but sort on AA005. The Table Request above would accomplish this.

Separating Report Sections

Underline

As you saw in the header and footer section, you can visually separate sections of your report by adding an underline. You can do the same with your SORT fields. For instance, let's print ID and MAJOR. We'll sort on MAJOR and add an underline every time the MAJOR changes:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 UNDER-LINE
END
```

The resulting report:

Prim Mjr 1	Student Number
ACCT	001133445
BUS	077654321
	078556434
	079989778
BUSF	008976576

Skipping a Line

Instead of drawing an underline, you could simply skip a line to increase readability: For instance:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 SKIP-LINE
END
```

As with underlines, you would use skip-line with a BY statement.

Double-spacing your report

To double-space an entire report, you can use SKIP-LINE with a field in your report. For instance:

```
TABLE FILE RTFILE
PRINT RT010 RT120 SKIP-LINE
BY RT120 NOPRINT
END
```

Note: You can only use skip-line ONCE in each report. To skip lines in headings and footings, use the caret instead: </N where N is the number of lines you wish to skip.

Printing the fields for each record on more than one line—using OVER

You can use OVER when printing out your fields to print information for someone/something on more than one line. For instance, let's print student id, name, major, OVER their Attempted Credits and GPA.

First you would pull the records you need from the RTFILE. Then you would join this to the AAFILE (name) and the RAFILE (GPA). After you have completed this part of your exec the last TABLE REQUEST would look like:

```
TABLE FILE STUDS
PRINT AA002 AS '' AA003 AS '' RT120 AS 'Major:' OVER
      RT020 IN 10 AS 'Att. Crds:' RP460 AS 'CUM GPA:'
ON AA002 SKIP-LINE
END
```

When you use OVER in a Focexec, you will no longer have any column headings at the top of each column as you normally do in a report. Title's/Headings for each field, if you want them on your report, need to be added with an AS phrase. These will print to the LEFT of the field, not over it. Even if you do not want any title next to a field, still include AS ' ' after the field. If you don't, it will shift the field over to the right leaving room for a title that doesn't print. So to eliminate the space, use AS ''. This will replace NOSPACE with the SPACE left for a title.

Notice on the second line of printing in the Focexec, we say IN 10. We do this to indent the second line slightly. This adds readability to the printout. We have also added on AA002 SKIP-LINE. After each set of data, it will skip a line. Here is the resulting report:

```
111223333     Bunny, Bugs                  MAJOR:  PSYS
     Curr. Att. Cred.     15.00  CUM GPA:     .000

222334444     Bird, Tweety                 MAJOR:  COMR
     Curr. Att. Cred.     15.00  CUM GPA:    3.130

33344555      Duck, Donald                 MAJOR:  COMJ
     Curr. Att. Cred.     15.00  CUM GPA:    2.513
```

Paginating your report

Focus numbers your pages consecutively beginning with one. There are several formatting techniques you can use to restart the numbers in sections, to force page breaks, and to keep sections together:

Forcing a page break:

To force a page break when a section changes, use it with a BY statement or in an ON sortfield statement:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 PAGE-BREAK
END
```

Note: If using PAGE-BREAK and SUBHEAD together the word AND must be used: BY RT120 PAGE-BREAK AND SUBHEAD.

Reset the page number to one after each page break

To reset the page number back to one each time your section changes, follow the page-break command with the command REPAGE:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 PAGE-BREAK REPAGE
END
```

Eliminating Page Numbers

To eliminate the printing of page numbers on your report include the following in your Focexec:

```
SET PAGE=OFF
```

You can add this line to the top of your Focexec by itself.

Keeping sections together

Sometimes you want Focus to check to see if it can fit all of the records for a section on one page and if it can't, you want it to begin a new page for that section. You could do this by using NOSPLIT:

```
TABLE FILE RTFILE
PRINT RT010
BY RT120 NOSPLIT
END
```

Formatting your text

There are several formatting techniques you can use to:

- rename column headings
- eliminate column headings
- change the NODATA symbol

Rename Column Heading

In Chapter XIV, Master File Descriptions, you learned how to determine what "title" will print out at the top of each column by looking at the MFD. You have the ability to change the title by using AS:

```
TABLE FILE RTFILE
PRINT RT010 AS 'Student ID,Number'
BY RT120 AS 'Major' PAGE-BREAK
END
```

The example above will print two new column headings: *Student ID Number* and *Major*. We have broken the title *Student ID Number* to two lines by using a comma. Enclosed the new title in single quotes.

Eliminate Column Headings

You can suppress the printing of column headings by adding ' ' after the AS phrase:

```
TABLE FILE RTFILE
PRINT RT010 AS ' '
BY RT120 AS 'Major' PAGE-BREAK
END
```

Change what prints when there is NODATA

When you are printing several fields of information for a person or account, sometimes there is no data found for that field. By default Focus puts one period in that field. This is sometimes difficult to see or read. You can change what prints when no data is encountered as follows:

Include the following in your last table request:

ON TABLE SET NODATA '*new value*'

where *new value* is up to 11 characters. For instance let's change the period to dashes:

ON TABLE SET NODATA '-----'

XVIII. Exercise Set Five-Formatting Reports

You will modify Focexecs created in previous exercises, adding formatting techniques to enhance the design of your reports.

XIX. Summarizing Numeric Data

With Focus, you have the ability to print totals, mean values, extreme values, and percentages. The chapter will cover:

- producing various totals
- how to use special operations such as averages, percentages, minimum and maximum values, etc. by using direct operators.

Producing Totals

Four types of totals:

- totals for numeric values in rows
- totals for numeric values in columns
- totals for one or more report sections
- grand totals

Row and Column Totals

You will use ROW-TOTAL or COLUMN-TOTAL to provide totals for rows or columns respectively. Use ROW-TOTAL either in a verb phrase or in an ON TABLE phrase that you place AFTER all BY phrases.

Example of ROW-TOTAL:

```
TABLE FILE RTFILE
COUNT RT010 AND ROW-TOTAL
ACROSS RT185/R
IF RT005 EQ '2002F'
IF RT020 GT 11
IF RT185 NE ' '
END
```

produces the following report:

	Dorm, Code								
BE	CH	GA	GR	LE	MA	MR	SH	TH	TOTAL
31	456	306	34	327	136	335	125	363	2116

Another way of writing this:

```
TABLE FILE RTFILE
COUNT RT010
ACROSS RT185/R
IF RT005 EQ '2002F'
IF RT020 GT 11
IF RT185 NE ' '
ON TABLE ROW-TOTAL
END
```

The difference? In the first example we are specifying a field on which to total. In the second example, we are saying total all of the numeric data in the report. The first is specific, the second is all inclusive.

Example of COLUMN-TOTAL:

```
TABLE FILE RTFILE
COUNT RT010 AND COLUMN-TOTAL
ACROSS RT120
BY RT005
IF RT005 EQ '2001F' OR '2002S'
END
```

produces the following report:

	Prim ,Mjr 1									
	ACCT	AMST	ARTH	ARTT	ARTU	BIOL	BUS	BUSA	BUSB	BUSC
Term										
2001F	133	11	7	44	7	119	53	12	58	2
2002S	126	12	7	36	10	132	4	7	53	0
TOTAL	259	23	14	80	17	251	57	19	111	2

As in ROW-TOTAL, COLUMN-TOTAL can be added to a verb phrase or in an ON TABLE phrase AFTER all BY phrases. So the above focexec could have be written as:

```
TABLE FILE RTFILE
COUNT RT010
ACROSS RT120
BY RT005
IF RT005 EQ '2001F' OR '2002S'
ON TABLE COLUMN-TOTAL
END
```

Use ON TABLE COLUMN-TOTAL when you want to total all numeric data in your report.

Sometimes you want to limit the columns, which are totaled. For instance, what if you had four fields printing out which contain numeric data but you only want to calculate totals for two of these fields?

You could write:

ON TABLE COLUMN-TOTAL FIELD1 FIELD2

Replacing field1 and 2 in the example above with the fields on which you wish to total. *Note:* Focus does not allow you to do this with ROW-TOTAL. For this reason, we do not recommend you use ROW-TOTAL in a multi-set request. A multi-set request is when you use multiple verbs and sort phrases. To provide totals that exclude the values of one or more numeric fields in a row, you must DEFINE or COMPUTE the value. This will be discussed in *Defining New Fields.*

When Focus prints out your totals, it will use the format of the field(s) you are totaling. If there are differing formats, it will use the default of D12.2. If however, the totals you are calculating are too large to fit in the format Focus is using, then only asterisks will display. To fix this, you need to lengthen the field by wither using a DEFINE statement or by appending a slash followed by the appropriate format (I,F,D, or P) and the desired field length.

When using ROW- and COLUMN-TOTAL, Focus will provide totals for all numeric fields you specify EXCEPT those contained in a BY PHRASE.

You can use ROW-TOTAL and COLUMN-TOTAL together in one request. For example:

```
TABLE FILE RTFILE
COUNT RT010 AND ROW-TOTAL COLUMN-TOTAL
ACROSS RT120
BY RT005
IF RT005 EQ '2001F' OR '2002S'
IF RT120 EQ 'ENGL' OR 'ENGW' OR 'ENGU'
END
```

produces the following report:

	Prim, Mjr 1			
	ENGL	ENGU	ENGW	TOTAL
Term				
2001F	46	54	50	150
2002S	57	40	61	158
TOTAL	103	94	111	308

Section Totals

Focus enables you to provide four types of totals for sections using:

- SUBTOTAL
- SUB-TOTAL
- RECOMPUTE
- SUMMARIZE

You can produce section totals for groups (those specified in BY phrases). For example, if your report contains sort phrases, which sort on TOTAL CREDITS and GPA, you can provide totals for either TOTAL CREDITS or GPA, or you can provide totals for both of them. When you request section totals, Focus will automatically create grand totals for your report.

The command you choose depends on the number of section totals you want, and on the presence of COMPUTEd values or values resulting from direct operations. See table below:

To produce totals for	Use the command
A single section (for example, for TOTAL CREDITS but not GPA)	SUBTOTAL
Multiple sections (for example, both TOTAL CREDITS and GPA)	SUB-TOTAL
A single section with at least one COMPUTEd value or a value that results from a direct operation	RECOMPUTE
Multiple sections with at least one COMPUTEd value or a value that results from a direct operation	SUMMARIZE

SUBTOTAL and SUB-TOTAL

To produce section totals, associate the command with the sort field for which you want totals. For example:

```
BY RA320   SUBTOTAL [MULTILINES] [FIELD1...FIELD 2]
           SUB-TOTAL
           RECOMPUTE
           SUMMARIZE
ON SORTFIELD
```

Use MULTILINES when you need totals for sections, which only have a single row of data. Field1 ... Field2 are fields you name if you wish to override the default and specify which fields you want totals for.

The command you will use, as mentioned above, differs based on the number of sections you want to produce totals for. For example:

To provide totals for GPA but not TOTAL CREDITS, you could write:

```
BY TOTAL CREDITS        OR    BY TOTAL CREDITS
BY GPA SUBTOTAL               BY GPA
                              ON GPA SUBTOTAL
```

To provide totals for a specific field and all fields, which precede it, associate the command with the innermost sortfield for which you want totals. For instance, in:

```
BY TOTAL CREDITS
BY GPA
```

we want totals for GPA (the innermost sortfield) and all sortfields which precede it (TOTAL CREDITS). So we could write:

```
BY TOTAL CREDITS        OR    BY TOTAL CREDITS
BY GPA SUB-TOTAL              BY GPA
                              ON GPA SUB-TOTAL
```

You can use an AS phrase for totals:

```
BY TOTAL CREDITS        OR    BY TOTAL CREDITS
BY GPA SUBTOTAL               BY GPA
                              ON GPA SUBTOTAL AS 'Subtotal for GPA:'
```

If you do not use an AS phrase to create your own title, Focus will print *TOTAL SORTFIELD. You can visually separate report sections by adding some blanks to the beginning of the AS phrase. This will indent the title. For example:

```
ON GPA SUB-TOTAL AS '      Subtotal for GPA:'
```

If you don't indent it, it will print at the margin.

RECOMPUTE

If a focexec contains a computed field, and you want a total for just a single section, you must use RECOMPUTE to create section totals. As in the examples above for subtotal and sub-total, you can do this either as part of a verb phrase or as an ON field statement:

BY RA320 RECOMPUTE or ON RA320 RECOMPUTE

This command provides totals for RA320 alone.

SUMMARIZE

If a focexec contains a computed field and you want section totals for more than a single section, you must use SUMMARIZE. As you saw with SUB-TOTAL, you can provide section totals for the innermost sort field and the sortfields, which precede it by associating SUMMARIZE with the innermost sortfield:

```
COMPUTE GPA RATIO/D4.2 = RA320/RA380
        BY RT105
        BY AA005 SUMMARIZE
```

The above example is not a good example of the compute statement. It is used simply as an example of summarizing. AA005 is the innermost sortfield so it will summarize AA005 and all of the sortfields which precede it: RT105.

Limiting sortfields for Summarizing

The example above provides summary totals for RT105, AA005, and RATIO. To limit summarization to the ratio only, rewrite that group of statements:

```
COMPUTE GPA RATIO/D4.2 = RA320/RA380
        BY RT105
        BY AA005 NOPRINT
ON AA005 SUMMARIZE RATIO
```

Suppressing totals for sections with just one record

Providing a summary total for a section, which contains only one record is redundant. To eliminate the summary total, by adding MULTILINES to the command. MULTILINES suppresses totals for groups that contain only one record. See the section on SUBTOTAL and SUB-TOTAL (above) for the proper syntax of using the MULTILINE option.

Paginating Report

If you report is created with the intention of distributing to several recipients you may want to include page break when the sortfield changes and then repaginate the numbers beginning with one. To do this use the commands PAGE-BREAK and REPAGE:

```
BY RT105 PAGE-BREAK REPAGE
```

Suppressing Grand Totals

Given the above example, you would probably want to suppress grand totals from printing at the end of the report. You can do this by placing

```
ON TABLE NOTOTAL
```

in the next-to-last line of your request.

Supplying a Total for a Sort Group

When you want to give the total number of "people" let's say with a specific sort group, you can use count to print the total. For instance, let's create a report, which prints the ID number and Name of Freshman sorted on Major. Then let's count how many people are in each major and print that count. We could write:

```
TABLE FILE RTFILE
PRINT RT120
BY RT010
IF RT020 GE '12'
IF RT14C EQ 'FR'
IF RT005 EQ '2000F'
ON TABLE HOLD AS FROSH
END

JOIN RT010 IN FROSH TO KEY IN AAFILE AS J1

TABLE FILE FROSH
COUNT AA003 AS 'Total,in,Major'
BY RT120 AS 'Major'
PRINT AA002 AA003 AS 'Name'
BY RT120
END
```

The above focexec uses two verbs. A request, which uses two verbs, is called a multi-set request. When you use more than one verb in a request, LIST or PRINT can only be used once, as the last verb. The BY phrase must be repeated.

An example of the resulting report:

Major	Total in Major	Student Number	Name
ACCT	27	117820775	Sieh, Albert
		121604273	Monck, Jessica M
		126720939	Nawrocki, Frank A
		128628446	Forte, Donna M
		130681806	Russo, Kristyn M
		130748609	Mitchell, Sinead M
		131708208	LaGattuta, Scott M
		134608718	Moyd, Kiel
		138702935	Holloway, Stuart H
		156769161	Rankin, Jennifer S
		174627178	Darragh, Michael Paul
		287606441	Sommers, Matthew R
AMST	2	038406151	Mekrut, William P
		098682640	Glaser, Allison E
ARTH	1	095620891	Lofaro, Gina E

Special Operations—average, percentage, etc.

Focus allows you to perform direct operations on numeric fields by adding a prefix to the fieldname in a verb phrase. With these operators, you can:

- average
- total
- find percentage
- find lowest and highest value for a field
- find the sum of the values for a field
- find the number of values stored for a field

You use direct operators with a verb phrase. Since these operators act on aggregated values, you want to use them with a verb that aggregates—SUM (or it's synonym WRITE) or COUNT. This table describes which direct operators you can use:

To	Use the direct operator	With the verb
Count the number of values for a field	CNT.	WRITE or SUM
Add the values for a field	SUM.	WRITE or COUNT
Calculate the average value for a field	AVE.	WRITE, SUM, or COUNT
Identify the lowest value for a field (for date fields, identifies the earliest date)	MIN.	WRITE, SUM, or COUNT
Identify the highest value for a field (for date fields, identifies the most recent date)	MAX.	WRITE, SUM, or COUNT
Provide the aggregate value for a sort group (in section headings and footings.)	ST.	not applicable
Provide the aggregate value for a tield (only for headings and footings)	TOT.	not applicable
Calculate the percentage of a column value to the column total	PCT.	WRITE, SUM, or COUNT
Calculate the percentage of a COUNT to the total count	PCT.CNT.	WRITE, SUM, or COUNT

CNT.

When you want to count the number of occurrences when the verb is SUM, use CNT.. Notice there's a period after CNT.. Let's say we want to create a report which shows the total number of quality points for a student and count the number of semesters he has been active.

We could use two verb phrases and two BY phrases such as:

```
TABLE FILE RTFILE
PRINT RT005
BY RT010
IF RT005 EQ '2001F' OR '2002S' OR '2002F'
ON TABLE HOLD AS RTSTUD
END

JOIN RT010 IN RTSTUD TO KEY IN RAFILE AS J1

TABLE FILE RTSTUD
SUM RA330 AS 'QUALITY,POINTS'
BY RA005
COUNT RT005
BY RA005
END
```

We can produce the same report by changing the COUNT and BY phrases to:

```
SUM RA330 AS 'QUALITY,POINTS'
CNT.RT005
BY RA005
END
```

Notice that we eliminate the need for a multi-set request—more than one verb and sort phrase.

SUM.

To add the values for a field when the verb is COUNT, use SUM. For example:

```
COUNT RT005 AS 'Semesters'
SUM.RP430
BY RP005
END
```

AVE.

To find the average value for a field, use AVE. with either of the verbs that aggregate—SUM or COUNT.

Let's say we want a report which displays a student's average GPA over three semesters. We could do this with:

```
TABLE FILE RTFILE
PRINT RT005
BY RT010
IF RT005 EQ '2001F' OR '2002S' OR '2002F'
ON TABLE HOLD AS RTSTUD
END

JOIN Rt010 IN RTSTUD TO KEY IN RAFILE AS J1

TABLE FILE RTSTUD
SUM RA380 NOPRINT
 AVE.RA360 as 'Average GPA'
 BY RA005
END
```

This would produce the following report:

Student Number	Average GPA
003344224	3.13
987788778	2.513
998788878	3.45
999887878	2.767

The above is a portion of the resulting report. To determine the average GPA for each student Focus adds the GPAs for the three semesters. It counts the number of times a value for the GPA occurs and divides the total GPA by the number of GPAs recorded for the student.

Direct operators in COMPUTE statements

The syntax for using a direct operator in a compute statement:

COMPUTE newfield/format=dir.operator.field;

for instance:

COMPUTE NUM/I4=CNT.RT010;

CNT.RT010 would then be used in your report in your report in a verb phrase, such as:

```
SUM RT120 as 'Major'
    CNT.RT010 as 'Number in major'
```

You could then total this column in your report in an ON TABLE SUBFOOT command by including your new computed field: NUM like:

```
ON TABLE SUBFOOT
"</2 Total in Major:  <50 <NUM"
```

TOT.

In a heading or a footing, you can provide the aggregate value for a field by using TOT. and the field. For example:

```
ON TABLE SUBFOOT
"</2 Total Expenses:  <TOT.RTXXX"
```

MIN. and MAX.

Let say you want to print out the lowest and highest GPA for each major for the Div. of Communications. You could do this with MIN. and MAX.:

```
TABLE FILE RTFILE
PRINT RT120
BY RT010
IF RT005 EQ '2002F'
ON TABLE HOLD AS RTSTUD
END

JOIN RT010 IN RTSTUD TO KEY IN RAFILE AS J1

TABLE FILE RTSTUD
SUM MIN.RA380 AS 'LOW GPA'
  MAX.RA380 AS 'HIGH GPA'
IF RA380 NE 0
BY RT120
END
```

This focexec would produce a report similar to:

RT120	Low GPA	HighGPA
ACCT	1.068	4.000
AMST	1.629	3.320
ARTH	2.466	4.000
ARTT	.964	3.607

This is just a portion of the report. What if you wanted minimum, mean, and maximum? You would add AVG.RP460 to your report. Congratulations—you are doing statistics!

PCT.CNT.

PCT.CNT. is used to obtain percentages of a COUNT based on a total count. For instance, let's create a report which counts the number of full-time students within each major and provides a percentage of that value in ratio to the total number enrolled. So the question we are looking to answer is if there are 300 Science majors at Marist, what percentage is that of all of the currently registered students?

We could use PCT.CNT. to accomplish this:

```
TABLE FILE RTFILE
SUM RT120 AS 'PRIMARY,MAJOR'
        CNT.RT010 AS '# OF,STUDENTS'
        PCT.CNT.RT120 as '% of ALL,MAJORS'
BY RT120 NOPRINT
ON TABLE SUMMARIZE
IF RT005 EQ '2002F'
END
```

An example of your resulting report:

PRIMARY MAJOR	# OF STUDENTS	% OF ALL MAJORS
ACCT	122	2.35
AMST	12	.23
ARTH	7	.13
ARTT	37	.71
ARTU	10	.19
BIOL	133	2.56
BUS	5	.10
BUSA	6	.12
BUSB	53	1.02
BUSF	130	2.51
BUSI	22	.42
BUSM	136	2.62
BUSO	4	.08
BUSP	51	.98
BUSU	121	2.33
CHEM	16	.31

XX. Defining New Fields

With Focus, you have the ability to create new fields. This makes it possible for you to include information on your report that is not stored in an IA Focus file as a value. You can define new fields for the purpose of:

- Calculating the values based on mathematical and logical relationships.
- Editing the way a field displays. For instance, change a student id number from 999445555 to 99-44-5555.
- Pulling the value of two or more fields together. For instance, if first name and last name were two separate fields, you could "concatenate" these two fields together by creating a new field for them.
- Changing the way the values for a fields print out on your report. For instance, you could change the decimal format of a numeric field.
- Translating the value for a field—DECODE the field.

There exist two ways for you to define new fields. You can use DEFINE or COMPUTE. There are two major differences between these methods.

1. The point at which Focus determines the value for the temporary fields,
2. The period for which the newly created field is available. DEFINEd fields are available for the reporting session or until cleared. COMPUTEd fields are available only for the TABLE request that contains the COMPUTE.

A new field which displays the value of a calculation

The RTFILE in IA SIS supplies you with the current attempted hours of a student. The RAFILE supplies you with a student's total credits. What if you wanted to print a new column on your report, which represented the "addition" of these two fields. You want to display the sum of total credits plus current attempted credits. You can do this with a DEFINE:

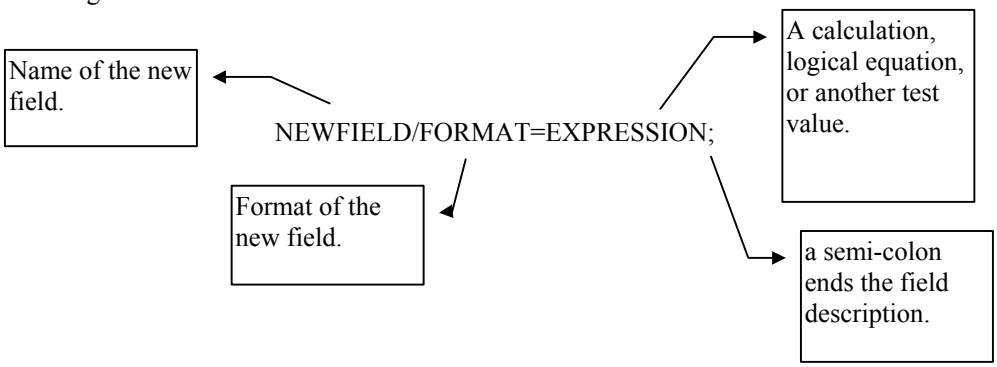
```
TABLE FILE RTFILE
PRINT RT020
BY RT010
IF RT005 EQ &SEM
ON TABLE HOLD AS CURRCRED
END

JOIN RT010 IN CURRCRED TO KEY IN RAFILE AS J1

DEFINE FILE CURRCRED
NEWTOT/I6=RT020 + RA320;
END

TABLE FILE CURRCRED
PRINT RT020 AS 'Curr,Credits' RA320 NEWTOT
BY RA005
IF RA305 EQ 'U'
END
```

An example of your resulting report:

Student Number	Curr Credits	Cum Earned Hours	NEWTOT
001501316	15.00	15.00	30
001509492	15.00	39.00	54
001560126	15.00	102.00	117

Your new defined field—NEWTOT is displayed in the above report. You can see how it correctly calculated the sum of current credits and cumulative credits.

The format for defining a new field is:



In the above example, I chose NEWTOT as the name of my new field. The name can be up to 8 characters long. I then typed the / and then the format for the field.

In this situation the field format I choose would depend on whether or not I was selecting undergraduates or graduates. Graduate students can get 1/2 credit for courses where undergrads cannot. This means, if I choose a format which strips the decimal places off when it displays and my population of students includes graduates, then my displayed numbers might be incorrect. If however, I know I am only working with undergraduates then I could take the decimal places off (or not). To leave the decimal points I could define the field format as D5.2. To take them off, I could use I3.

When you assign this number, make sure it's large enough to accommodate the resulting calculation. I then added an equal sign and the expression. The expression in the this case is the addition of two fields: RT020+RA320 and a semi-colon at the end of the line. You must END your define statement.

You can then use this newly defined field in your table request. Remember a defined field is available for the duration of your Focus session.

A new field which uses a previously defined field in it's expression

Marist determines a student's class year for you but in the following example, we'll take the result of our new field and create a new field called FROSH which will return a Y if a person will still be a freshman after the calculation of a N if the person will be an upperclassman after the calculation:

FROSH/A1=IF NEWTOT LT 30 THEN 'Y' ELSE 'N';

We can now constrain on this new field in our request:

```
TABLE FILE RTFILE
PRINT RT020
BY RT010
IF RT005 EQ &SEM
ON TABLE HOLD AS CURRCRED
END

JOIN RT010 IN CURRCRED TO KEY IN RAFILE AS J1

DEFINE FILE CURRCRED
NEWTOT/I6=RT020 + RA320;
FROSH/A1=IF NEWTOT LT 30 THEN 'Y' ELSE 'N';
END

TABLE FILE CURRCRED
PRINT RT020 AS 'Curr,Credits' RA320 NEWTOT
BY RA005
IF RA305 EQ 'U'
IF FROSH EQ 'Y'
END
```

The new report is selecting those students who will still be "Freshman" after the calculation. If I wanted to do just the reverse—eliminate freshmen, the IF statement could be rewritten as:

IF FROSH EQ 'N'

Note: you could have added a calculation to the end of the FROSH field:

FROSH/D6.2=IF NEWTOT LT 30 THEN 1 ELSE (FIELD1-2) * FIELD2;

Now FROSH does the following: If the value of NEWTOT is LT 30 then return a value of 1. If it is not, subtract 2 from Field 1 and multiply the result by field2. Notice the field format needed to be changed so that the new field could accommodate the result of the calculation.

Valid Arithmetic Expressions include:

+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation.

Below is a direct example from FOCUS for IBM Mainframe User's Manual—Volume I:

```
DEFINE FILE SALES
COUNT WITH PROD_CODE = 1'
RATIO = DELIVER_AMT/OPENING_AMT;
NEWVAL = (RATIO-1) ** 2;
END
```

1. All numeric values are converted to double-precision floating point decimal format before use in calculations. When a large number is converted to decimal format, a rounding error may occur, and should be taken into account in these types of calculations.
2. If you attempt to divide by 0, FOCUS sets the defined value to 0.
3. If a number is too large or too small, Focus displays an OVERFLOW or UNDERFLOW warning.

4. Focus evaluates the arithmetic operators in a certain order. This order of execution may affect the evaluation of an arithmetic expression. Arithmetic operations are preformed before logical operations, in the following order:

FIRST:	**	exponentiation
SECOND:	/ *	division and multiplication
LAST:	+ -	addition and subtraction

5. When operators are at the same level, the evaluation is performed from left to right. To change this order, you must use parentheses. Expressions in parentheses are evaluated before any other expression.[1]

The gist of the above quoted material from the Focus User's guide is that Focus processes mathematical expressions according to the standard sequence: first the contents of parentheses, then exponentiation, then multiplication and division, finally, addition and subtraction.

Using EDIT

Shorten length of field

If a field called NAME is 32 characters long but we KNOW it never exceed 20, we could shorten the length of the displayed field by defining a new field:

```
NAMESHRT/A20=EDIT (NAME,'99999999999999999999');
```

We are using nines (9999...) in the above example because we just want the first 20 characters from the NAME field.

Format a field

Let's format the student id number from 111223333 to 111-22-3333. The field name for student id is AA002 in this example. We would write:

```
STUID/A11=EDIT (AA002,'999-99-9999');
```

To accommodate the two dashes, we increase the field length to 11. Focus prints the first three characters of the id number, a dash, the next two characters, another dash, and then the last four characters. We now have a nicely formatted id number.

Displaying two fields as one—concatenation

What if first name and last name were two separate fields, each 16 characters long and we wanted them to display as one full name? The existing fields are FIRSTNAME and LASTNAME. To concatenate them, we would write:

```
FULLNAME/A34=LASTNAME||(',  '|FIRSTNAME);
```

We chose 34 characters for our new field. 32 is the sum of both fields plus two for the comma and the space which follow. Our new field—FULLNAME will print like:

```
Smith, Joseph
Smyth, Jane
etc.
```

We accomplish this by putting LASTNAME as the first field to pull, then we type two vertical bars: ||. Two vertical bars mean strong concatenation. It will eliminate trailing blanks after the last name. We then want to literally print a comma so we enclose it in apostrophes. We then want a weak concatenation - one vertical bar - and then the first name. The weak concatenation will preserve the space.

[1] Focus for IBM Mainframe Users Manual Volume I, Release 6.5 pp. 2-93 - 2-94

Sorting on new fields

You can only sort on newly created fields which have been DEFINEd. If you COMPUTE them, you cannot sort on them.

DECODE

DECODE can be used to translate a "code" to a full length description. For example ENSC to Environmental Science or FR to Freshman. At Marist we provide you with a subroutine to accomplish this translation. It is called XSFOCT and is described in the chapter entitled *Printing the Long Translation of a Field.* Let's look at the values of RT14C in SIS. The valid values are:

FR
SO
JR
SR

They could be translated to Freshman, Sophomore, etc. using the XSFOCT subroutine. However, when you sort them, they come out as Freshman, Junior, Senior, Sophomore in that order because they sort alphabetically. You could use DECODE to fix them so they sort properly:

```
DEFINE FILE RTFILE
CLSS/A12=DECODE RT14C('FR' 'A-FRESHMAN' 'SO' 'B-SOPHOMORE' 'JR' 'C-JUNIOR'
    'SR' 'D-SENIOR');
END

TABLE FILE RTFILE
PRINT RT010
BY RT120
BY CLSS
IF RT005 EQ &SEM
END
```

resulting report:

Prim Mjr 1	CLSS	Student Number
ACCT	A-FRESHMAN	010722551
		999886754
	B-SOPHOMORE	002604035
		008998998
AMST	C-JUNIOR	008568009
	D-SENIOR	016626733

You could have accomplished the same result by using EDIT but look how long it is:

```
DEFINE FILE RTFILE
CLSS/A12=IF EDIT (RT14C,'999') EQ 'FR' THEN
  EDIT (RT14C,'A-FRESHMAN') ELSE
  IF EDIT (RT14C,'999') EQ 'SO' THEN
  EDIT (RT14C,'B-SOPHOMORE') ELSE
  IF EDIT (RT14C,'999') EQ 'JR' THEN
  EDIT (RT14C,'C-JUNIOR') ELSE
  EDIT (RT14C,'D-SENIOR');
END
```

Computing a field

In the very first section of this chapter we DEFINEd a new field named FROSH. Since we did not sort on this field, nor did we intend to use it in any other reports, we could have COMPUTEd it instead:

```
TABLE FILE RTFILE
PRINT RT020
BY RT010IF RT005 EQ &SEM
ON TABLE HOLD AS CURRCRED
END

JOIN RT010 IN CURRCRED TO KEY IN RAFILE AS J1

TABLE FILE CURRCRED
PRINT RT020 AS 'Curr,Credits' RA320
COMPUTE NEWTOT/I6=RT020 + RA320;
BY RA005
IF RA305 EQ 'U'
END
```

If we had simply wanted to just print the result of this calculation in the report, then using COMPUTE is a much quicker method for accomplishing this. It's one line versus an entire DEFINE section. The difference with using the COMPUTE is that you cannot use this new field anywhere else but within this TABLE REQUEST as I did at the beginning of this example (Unless you put it in a hold file.) So, if I had wanted both, I would go back to using the DEFINE.

DEFINE vs COMPUTE

As you have seen, you can use DEFINE or COMPUTE to create new fields. Deciding which one to use depends on several factors:

- Whether you want to use the new field as a sortfield. If you do, then you must DEFINE it.
- Whether Focus requires detailed or aggregated values to calculate the values for the new field. If the value of the new field can only be reached by using aggregated values, then you must use a COMPUTE.
- Whether you plan to use the field in more than one report per reporting session. If the value of the field can be determined by using detailed values, you can DEFINE it.

Key Differences

There are two major differences between DEFINEd and COMPUTEd fields.

- The point at which Focus determines the value for the temporary fields, and
- The period for which the newly created field is available. DEFINEd fields are calculated only once and are available for the reporting session or until cleared. COMPUTEd fields are available only for the TABLE request that contains the COMPUTE.

The point at which Focus processes the calculations and determines the temporary field values has implications for how you can use the fields in a TABLE request. It also has implications for whether to COMPUTE or DEFINE a field. The following example illustrates the difference between DEFINEd and COMPUTEd field values:

An Example of DEFINE vs COMPUTE

Determining valid Dean's List students

Let's imagine that a student's CUM GPA is evaluated for two semesters—Fall and Spring to determine if they are eligible for Dean's List. So that's one Academic Year's worth of data. Let's also say that to be a candidate for Dean's List, the student must achieve a CUM GPA of 2.5 or better in EACH semester. To determine if a student has met this criteria, we would want to examine their CUM for each of the two semesters and see if it is 2.5 or above. Since a DEFINEd field can look at detailed values, we can look at each semester independently. Since a COMPUTEd field can only be used for aggregated values, we must first SUM the two GPAs and divide by 2. So this is what our Focexec looks like:

```
TABLE FILE RTFILE
PRINT RT040 RT005
BY RT010
IF RT020 GE 12
IF RT005 EQ &SEM
ON TABLE HOLD AS SEMGPAS
END

DEFINE FILE SEMGPAS
  DEANSLST/A3 = IF RT040 GT 2.5 THEN 'YES' ELSE 'NO';
END

TABLE FILE SEMGPAS
SUM RT040 NOPRINT
  COMPUTE YEARGPA/D5.2 = RT040/2; AND
  COMPUTE DEAN2/A3 = IF (RT040/2) GT 2.5 THEN 'YES' ELSE 'NO';
BY RT010
PRINT RT040 DEANSLST
BY SUBKEY_RT
PRINT RT040 DEANSLST
BY RT010
END
```

The newly defined field—DEANSLST—is examining the student's semester GPA (both of them) and returning a value of YES if it is over 2.5 and NO if it is not.

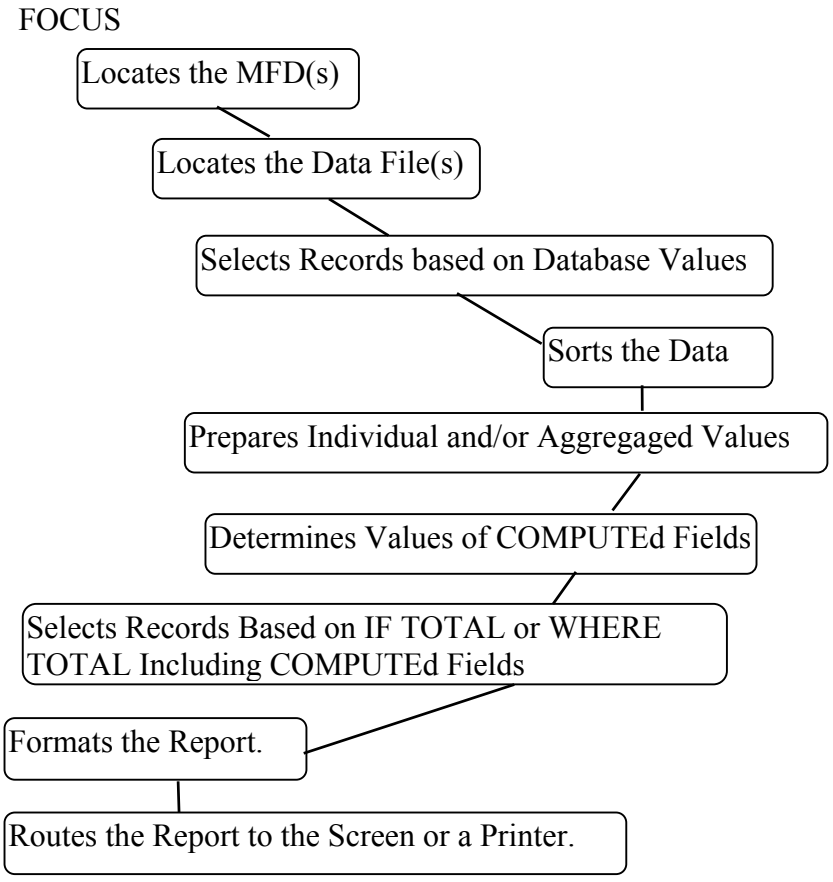
In comparison, the COMPUTEd field—DEAN2—is taking the value of the SUMmed semester GPA (both of them) and dividing the result by 2. So the COMPUTE is working with the aggregated value while the DEFINE is working with the detail value. The danger with using COMPUTE in this example, is that a student could have a high GPA in one semester, a very low one in another semester, and still look like he/she achieved Dean's List. See the following report:

RT010	YEARGPA	DEAN2	RT040	DEANSLST
MC999776565	2.77	YES	2.44	NO
			3.100	YES
MC122334444	1.72	NO	.000	NO
			3.438	YES

Look at the first person. The COMPUTEd field (DEAN2) has determined he has made Dean's List, because it is based on the YEARGPA. The DEFINEd field (DEANSLST) shows you that this student does not qualify for Dean's List because he did not achieve a 2.5 in one semester.

Please note: There is usually more than one way to create a report in Focus. The example was designed as an illustration of the difference between COMPUTE and DEFINE. There are other methods we could have used to achieve the result above.

The sequence in which Focus processes DEFINEd and COMPUTEd fields.



Selecting Records on SUMmed, COUNTed values or COMPUTEd fields

To screen on aggregated or COMPUTEd values, you must use IF TOTAL or WHERE TOTAL tests. For example:

```
TABLE FILE RTFILE
SUM RT330
BY RT010
IF RT005 EQ '2001S'
IF RT020 GE 1
END
```

The Focexec above SUMs the number of credits a student has earned in a past semester. It displays the student's id number and the total number of earned hours for the student.

To now select only those students who earned between 12 and 15 credits, we would use IF TOTAL or WHERE TOTAL:

```
TABLE FILE RTFILE
SUM RT330
BY RT010
IF RT005 EQ '2001S'
IF RT020 GE 1
IF TOTAL RT330 FROM '12' TO '15';
END
```

XXI. Addresses & Phone Numbers in SIS

There are two procedures for printing addresses in Focus. You can print an address and/or phone number in a list (report) or you can create labels. This section covers printing addresses in a list. For information on printing labels, see *Creating Labels* section.

To print addresses on a report, you have to use what's called a subroutine. This mini program looks at all the addresses a student has and provides you with the one you specify. The subroutine provides you with new "fields" you will use to print addresses. They are referred to as the *AP elements* because they all begin with AP.

Including the AP elements in your Focexec

There are three components you need to include in your Focexec to use the AP elements:

1. -SET &ADDRPREF = 'xxx';
2. a DEFINEd field entitled SSN/A09 = social security number (explained below)
3. -INCLUDE ADDRESS1

-SET

The -SET command tells Focus to assign a value for ADDRPREF. ADDRPREF is your way of telling Focus which address to include in your report. The PREF in the variable ADDRPREF says, what is your first preference for an address, if I can't find that one is there an alternative I can provide you with? For example, you would like to print addresses for full-time undergraduates. You would like their dorm address if they are a resident, their commuter address for an on-campus PO Box, or a permanent address if they do not have either of the previous ones. You would write:

```
-SET &ADDRPREF = 'MP';
```

Your -SET command can be at the very top of the Focexec or you can include it after your last JOIN.

DEFINEd SSN

You must define a field called SSN. Focus needs the institution and the ID number of the student to pull the correct address from the AAFILE for you. For most SIS users, you begin your Focexec by selecting records from the RTFILE. You do this with the intention of joining this smaller subset of data to either the RA or the AAFILE or both. In this case you have already used institution and id to join—RT010. You can use RT010 in your define:

```
DEFINE FILE HOLDFILE
   SSN/A09 = RT010;
   -INCLUDE ADDRESS1
END

;
```

-INCLUDE ADDRESS1

The -INCLUDE ADDRESS1 must be included in your DEFINE as shown above.

Available addresses:

At this time, you have the ability to print one address only. (You could print DORM and one other address discussed in Tips & Tricks at the end of this section, but you could not print BILLING and COMMUTER—only DORM and one other address.) The available addresses include:

1. Address Pref.

PLUS
P- Permanent
B- Billing
L- Local
M- Mailing (Marist)

The ones listed are the most common.

After you include the necessary components to include addresses, your Focexec might resemble:

```
TABLE FILE RTFILE
PRINT RT120
BY RT010
IF RT005 EQ &SEM
IF RT020 GE 12
IF RT14E EQ 'U'
ON TABLE HOLD AS UNDR
END

JOIN RT010 IN UNDR TO KEY IN RAFILE AS J1

-SET &ADDRPREF = 'MLP';
DEFINE FILE UNDR
  SSN/A09 = RT010;
-INCLUDE ADDRESS1
END

TABLE FILE UNDR
.....
```

You are now ready to include the addresses in your Focexec. The AP elements described at the beginning of this section can be considered as FIELDS in your HOLD FILE. They are listed below.

Name—If you join to the AAFILE, you can use the name from there. But if one of the following formats suits your needs and you plan on including addresses in your Focexec, then you can use one of these. Please Note: You DO NOT have access to AA005 (the name sort field) with this. So if you planned on using AA005, you still have to join to the AAFILE.

Field Name	Name Format	Length
AP100	Last, First	32
AP105	First Only	32
AP110	Last Only	32
AP120	Salutation Last	32
AP125	First Middle Last	32
AP130	Salutation First Last	32

Formatted Addresses—You would use this set of fields if you want to suppress the printing of a blank line if the student has a blank address line. For example, if a record does not have an address line 2, then it suppresses the printing of that line and brings address line 3 right up under line 1, so the result is a nicely formatted address.

FieldName	Name Format	Length
AP200_F	Formatted Address Line 1	32
AP205_F	Formatted Address Line 2	32
AP210_F	Formatted Address Line 3	32
AP215_F	Formatted Address Line 4	32
AP245_F	Formatted Address (County Long)	20
AP250_F	Formatted Address (Phone)	13

Unformatted Address—The unformatted address is most appropriately used if you are creating a Focexec with the intention of downloading the information in raw data format. It does not suppress the printing of blank address lines.

FieldName	Name Format	Length
AP220_U	Unformatted Address (Street 1)	32
AP225_U	Unformatted Address (Street 2)	32
AP230_U	Unformatted Address (City)	13
AP235_U	Unformatted Address (State)	02
AP240_U	Unformatted Address (Zip)	09
AP245_U	Unformatted Address (County Xx)	02
AP250_U	Unformatted Address (Phone)	10

An example of using the AP fields:

```
TABLE FILE RTFILE
PRINT RT120
BY RT010
IF RT005 EQ &SEM
IF RT020 GE 12
IF RT14E EQ 'U'
ON TABLE HOLD AS UNDR
END

JOIN RT010 IN UNDR TO KEY IN AAFILE AS J1

-SET &ADDRPREF = 'DCP';
DEFINE FILE UNDR
  SSN/A09 = RT010;
-INCLUDE ADDRESS1
END

TABLE FILE UNDR
PRINT AP100 AS ' ' RT120
   OVER AP205_F AS ' '
   OVER AP210_F AS ' '
   OVER AP215_F AS ' '
   OVER AP245_F
BY AA005 NOPRINT
END
```

This Focexec would create the following report:

```
Smith, Gina R                MAJOR     PSYS
Glenmont  NY
          12345

Cotter, Amy J                MAJOR     MEDT
Cumberland ME
          03584
```

Tips & Tricks

If you want to print more than one address for each student you can print their Local Address and one other address.LOcal and Mailing are available in the RT and AAFILE, respectively. So, if you want Local and Billing address for instance, pull Local from the RTFILE and Mailing from the AAFILE (by joining to it). Then use the three components necessary to include addresses in your Focexec. You can then print both—Local & Mailing and Billing Address.

XXII. Printing

There exist quite a few printing options at Marist. The type of Focexec you have created determines the options available to you.

Type & Options:

List less than 120 pages:

- Launch it to print when it is displayed on your screen – Hot Screen Printing
- Send it to your Lotus Notes Account (as a Word, PDF or HTML file)
- Sign on to MVS Batch and Batch it to your TSO Account
- Sign on to MVS Batch and Batch it to a Mainframe Printer

List longer than 120 pages:

- Sign on to MVS Batch and Batch it to your TSO Account
- Sign on to MVS Batch and Batch it to a Mainframe Printer

Non-Print File:

- Sign on to MVS Batch and Batch it to your TSO Account
- Sign on to MVS Batch and Batch it to a Mainframe printer

Labels:

- Sign on to MVS Batch and Batch it to a Mainframe Printer

Focus itself provides two methods of printing reports—one allows you to print your output after you've looked at it on-line. The other allows you to sign-on to IA Batch to submit your report. These two methods provide you with great flexibility. The first method has been even further expanded to allow you to send your printout to your Lotus Notes email account or to your TSO Account. Through either of these methods, you can first view your data and then either open in another application (such as Excel) or download to your PC.

This means you can work with this list data locally – either open in a PC Application or perhaps print to a Network Printer in your office. This section will cover:

- hot-screen printing vs. batch printing
- printing from the hot-screen including sending to your Lotus Notes account
- submitting your jobs through batch
- submitting labels through batch

Hot-screen printing is for lists only. (Labels and non-prints must be batched) Hot-screen printing refers to the capability of printing the report you see on your screen when you run a focexec on-line from the TED Editor or on-line focus. The report created and displayed is considered hot-output because you are viewing the results immediately. Hot-screen printing lets you send that report or a page of that report to the printer. (Note: You do have a limitation of 120 pages. If your report is longer than 120 pages, you must batch it.)

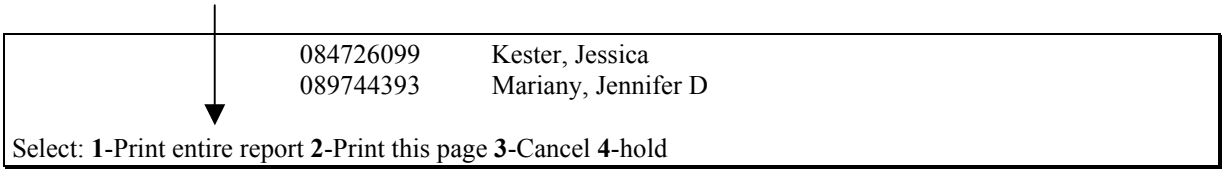
Batch printing refers to the capability of printing the report without having to use on-line focus to view and execute the focexec. The batch system is the method you use to submit for mailing labels and nonprints.

Picking up your output

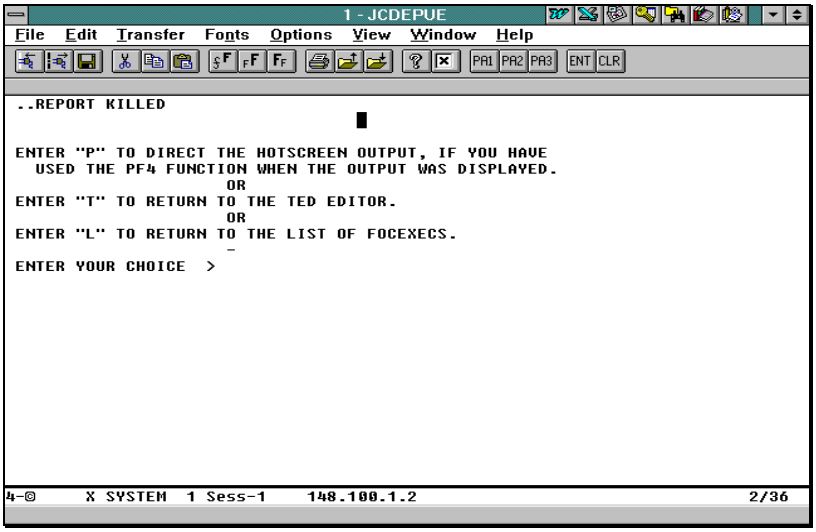
Whichever method you choose for printing, if you send your job to the Machine Room in Donnelly (LD150A or CENTRAL), your output will be placed in your Computer Mailbox located on the bottom floor of Donnelly nearest the Main Entrance of Donnelly. This is a combination box. If you are new to Marist or to printing, check with someone in your office to obtain your box number and combination.

Printing using Hot-screen

To print your focexec using the on-line function, you must first run the report. When you see the output on the screen and you are satisfied with the results you can print this report by pressing **PF4**.
You will see the following **four** options listed on the bottom of the screen:

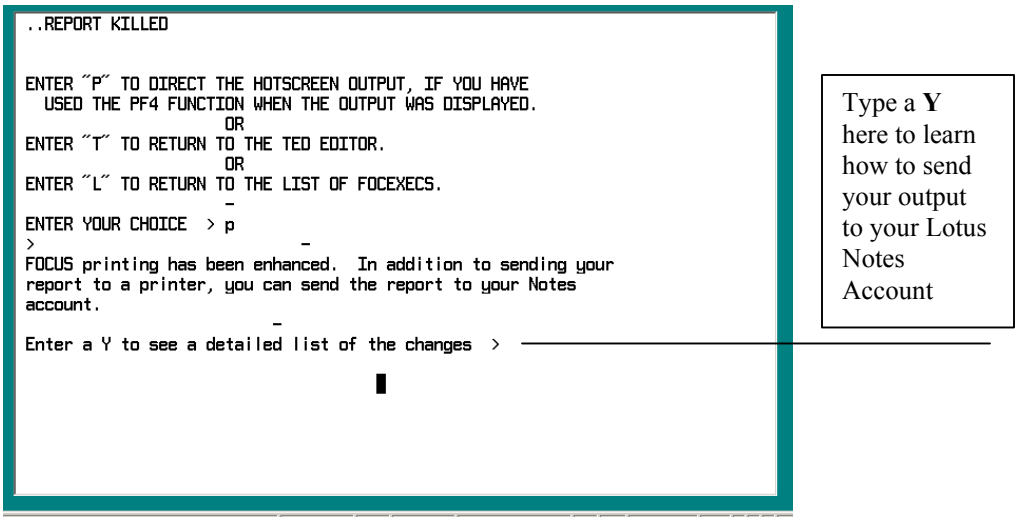


If you want to print the entire report press 1 and Enter. (2 for that PAGE ONLY). *Note*: The hold option does not work. Please ignore this option. The Focus system takes a snapshot of the report and holds it for you. To complete printing, press PF3 to get to the PRINT report prompt. The following screen is displayed:

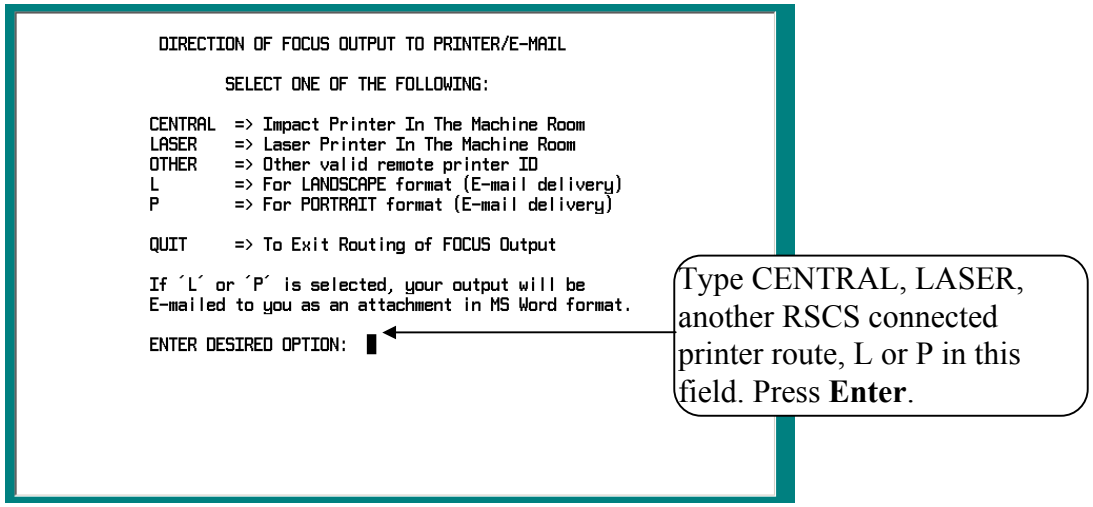


Type P and press **Enter**.

The following screen displays:



You can either type **Y** to view how to send your output back to Lotus Notes or type **N** to continue. If you choose Y, after viewing the Help File, the following will display. Accordingly, typing N displays this next screen:



Your options here are as follows:

Central: You can send your output to an impact printer in the Donnelly Machine Room. Your output will be printed on continuous form blue bar paper or white paper depending on the FORM number.

Laser: You can send your output to a laser printer in the Donnelly Machine Room. You can choose exactly what type of paper your output will be printed on by changing the FORM number described later. This includes having your output duplexed or perhaps printing more than on logical page on a physical piece of paper.

Other: Some larger offices have their own RSCS printer installed directly in their offices. An RSCS printer has a ROUTE and FORM # associated with it. This is usually designated on the printer in some way such as a label.

L: If you wish to send your output to your Lotus Notes Account and you have a list that extends beyond 80 characters, choose L – this is Landscape Mode.

P: If you wish to send your output to your Lotus Notes Account and you have a list that is less than 80 characters, choose P – this is Portrait Mode.

Batching your reports

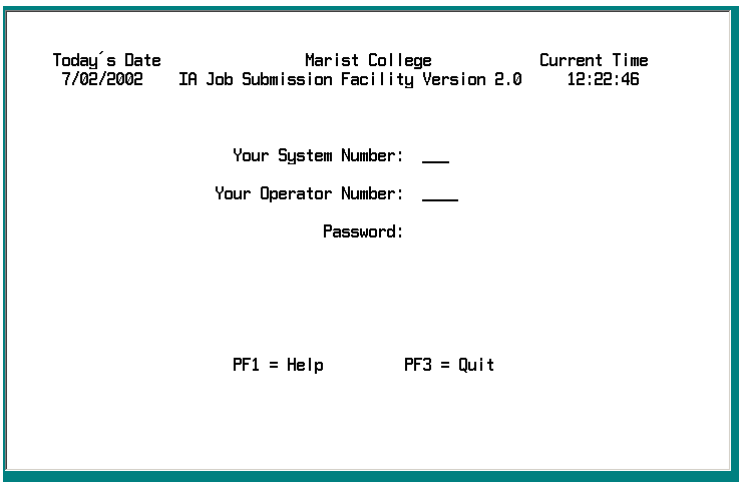
There are a few ways to access the batch system. The method you choose will depend on your personal preference.

Accessing Batch from On-line Focus

To access batch from on-line focus, go to the focus menu and move the cursor to *Access to Batch* and press Enter. The system will display the following screen:

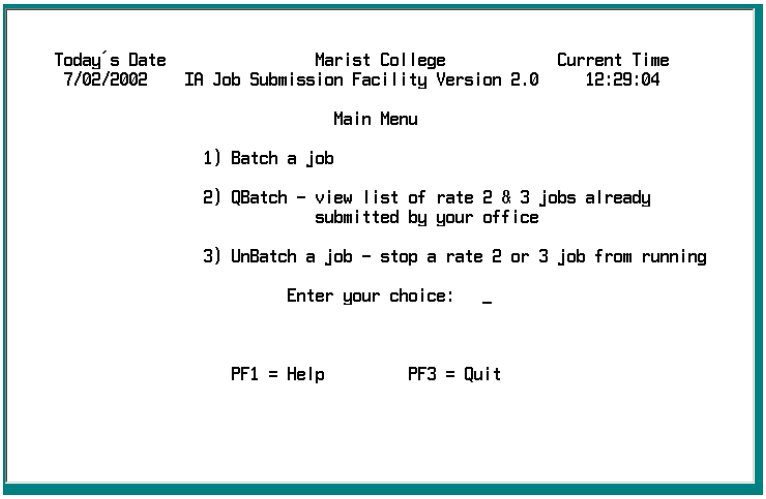
```
EZA8200I MVS TCP/IP TELNET CS V2R10
EZA8256I Connecting to MVS.MARIST.EDU 148.100.1.12, port 4023

EZA8270I Using Transparent Mode...
EZA8272I Notes on using Telnet when in Transparent Mode:
EZA8273I - To enter Telnet Command, Hit PA1
***
```

When you see the three asterisks, you must press **Enter**. The Batch Facility Sign-on Screen will display:

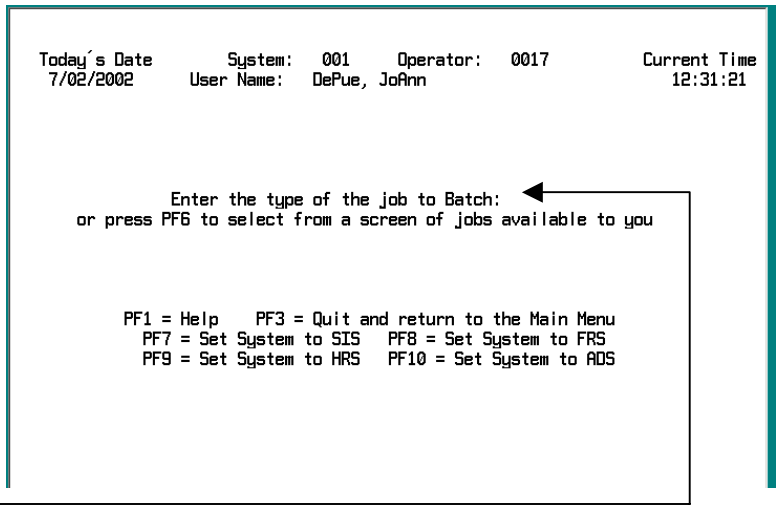


Fill in the fields next to the prompts with the correct information. Your operator number and password are the same as your IA sign on.

When you have correctly signed on, the following screen displays:



Type **1** at the prompt and press **Enter**. The following screen displays:

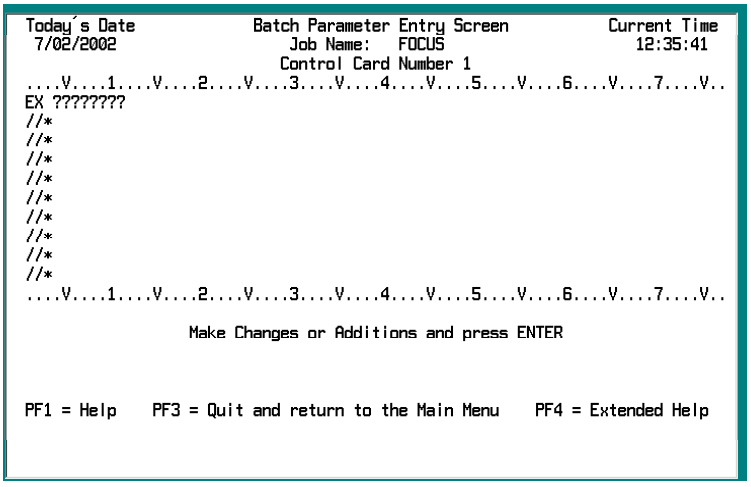


At this prompt, enter the name of the batch job you want to use to print your focexec. You options are:

FOCUS	Prints lists and other jobs on plain paper
FOCUST	Prints labels
FOCUSNP	Creates nonprints by shipping the data to your TSO account

After you have typed in one of these options, press **Enter**.

The Batch Job Parameters screen displays:



Replace the question marks next to the EX with the name of the focexec you want to print.
If **your** focexec uses variables to select data, you must type them on this screen following the focexec name. For example, if you have a variable defined to ask for the semesters you want the report to select from you would type,

EX EXAMPLE1 SEM=2002F

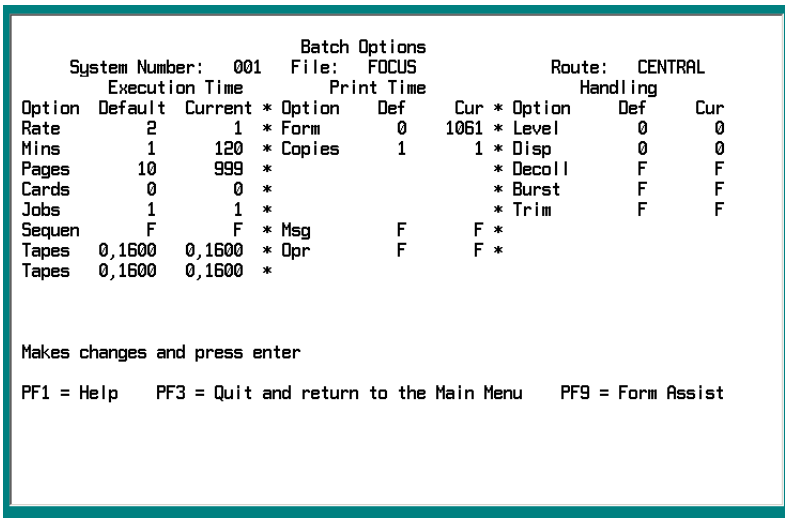
If you wanted to specify multiple semesters, you would type,

EX EXAMPLE1 SEM=2001F OR 2002S

If you have **multiple** variables in your focexec, you must specify their names and values in the same order that they appear in the focexec. (This is why it's very important to document the focexec so you don't have to hunt through the entire focexec to find the variables.) For example, if your focexec asks for semester, defined as SEM, and asks for class (RT14C), defined as CLASS, you would type the EX line as,

EX EXAMPLE1 SEM=2002F CLASS=FR

Once you have finished entering the variables, press Enter to move to the next screen, the *Batch Options Screen*.



```
                          Batch Options
     System Number:   001   File:   FOCUS              Route:   CENTRAL
         Execution Time          Print Time               Handling
Option  Default  Current * Option    Def     Cur * Option     Def     Cur
Rate          2        1  * Form        0    1061 * Level        0       0
Mins          1      120  * Copies      1       1 * Disp         0       0
Pages        10      999  *                       * Decoll       F       F
Cards         0        0  *                       * Burst        F       F
Jobs          1        1  *                       * Trim         F       F
Sequen        F        F  * Msg         F       F *
Tapes   0,1600   0,1600  * Opr         F       F *
Tapes   0,1600   0,1600  *


Makes changes and press enter

PF1 = Help    PF3 = Quit and return to the Main Menu    PF9 = Form Assist
```

There are several options, which you can change on this screen. The most important ones are, ROUTE, RATE, and FORM.

ROUTE

You have four options

1. **CENTRAL** The Central printer is the default. This means if you don't change CENTRAL to something else, your job will print on the high speed impact printer in the Donnelly Machine Room. It will print on normal 14 7/8 x 11 computer paper unless you change the FORM number. See FORM below. Your printout will not be letter quality.
2. **LD150A** Change CENTRAL to LD150A if you wish to print on 8½ x 11 white paper in letter quality format. If you do not change the form, your printout will be in landscape on letter size paper. The font will be slightly smaller than normal to fit all of the text on this size paper.
3. **OTHER** If you have an RSCS connected printer in your office, it should clearly have the ROUTE marked on it. This is not a PC printer. It is a mainframe printer with a ROUTE similar to PD250N or LD134A, etc. When sending to a local printer in your office/building, it is important to include the proper FORM number. This will also be clearly marked on the printer.
4. **TSO USERID**. You can send your job back to your TSO account. Enter your HM userid in the ROUTE field excluding your subcode, for example HMCC. You would then view your job in TSO by going to SDSF OUTPUT.

RATE

1. The rate determines what TIME your job will begin running. The default is 1. The turn around time for a Rate 1 job is up to three hours. Sometimes it is done sooner. Three hours is the high end.

FORM

FORM NUMBERS for sending to CENTRAL

1061 Standard 14 7/8 x 11 computer paper

1261 8½ X 11 perforated paper - portrait. If you use this form number make sure your line will not exceed 80 characters, because your the right side of your text will be cut off if it does.

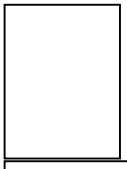
SPECIAL FORMS:

Includes specialty forms such as Official Transcripts, mailers, etc. Information regarding these forms can be viewed at: http://www.marist.edu/it/operations/form.html

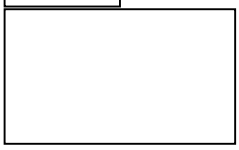
FORM NUMBERS for sending to LD150A (Laser Printer)

When sending to the Laser Printer, you are sending in PORTRAIT orientation or LANDSCAPE orientation.

PORTRAIT

This is portrait: 

It is a normal 8½ x 11 paper. If you send your job with a portrait job Number, each line of your printed output cannot exceed 80 characters. If it does, any characters after 80 will be cut off.

This is landscape: 

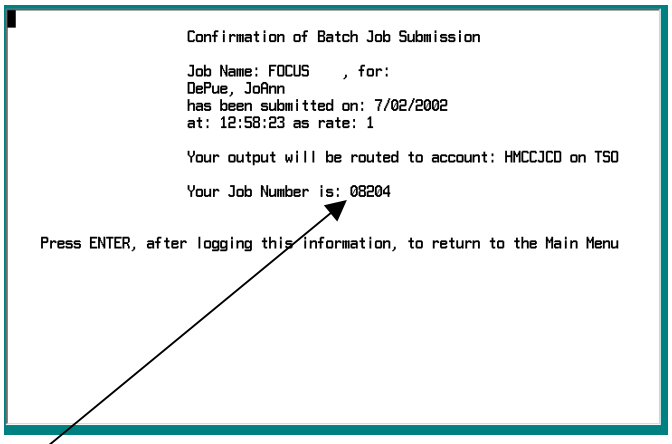
The paper size is still 8½ x 11, however, your output now prints longways on the paper. To accomplish this, a slightly smaller font is used. The result is a very neat, professional looking report.

PORTRAIT form numbers:	1261	Text prints on one side of 8½ x 11 paper.
	1351	Text will print on one side of 3-hole punch paper
	6251	Text will print on both side (duplex) of 8½ x 11 paper.
LANDSCAPE form numbers:	1061	Text will print long ways across one side of 8½ x 11 paper.
	1341	Text will print long ways across 3 hole punch 8½ x 11 paper
	6041	Text will print long ways across both sides of 8½ x 11 paper

You can change the form for your job by typing in a different form number next to FORM. To view a list of all available forms, see: http://www.marist.edu/it/operations/formdef.html

If you want to change any of the other available options and want more information about them, press PF1 while on the *Batch Options* screen. A description of each option is listed.

After you press Enter on the *Batch Options* screen, the system displays information about your job. The screen looks similar to this:



Write down your Job Number for future tracking purposes in the event that your job is missing. The Help Desk will ask for this number to track down the job.

Accessing Batch from IAPROD

To access the batch system from IAPROD, first sign on to IAPROD. In the screen field type BAT and press Enter. The *IA Job Submission Facility Version 2.0* screen displays. You must type in the system number, operator number and password. To continue, follow the instructions above starting from *Batching a Focexec*.

Batching Labels

Submitting for labels is similar to batching a list. There is a special batch job setup for printing labels. You should have already created the label focexec. For more information on creating labels see *Creating Labels* chapter.
To submit for labels follow the sign on procedures to get to batch. When the batch system asks for the name of the job to batch type FOCUST.

On the *Batch Parameters* screen, type the name of your label focexec next to the EX. Remember to delete any extra questions marks. Press the spacebar to skip a space and type ADDRPREF=xxxxxxxx, where xxxx is the letter representing the address type you want to print.

ADDRPREF gives you the option select which address is printed on the label. For example, you want to print students dorm address, but if the student doesn't have a dorm address you want the system to print their permanent address. To tell the system to print addresses in this order you would enter the preference as:

ADDRPREF=DP

There are several option available. The most common are:

L	Local
M	Mailing
P	Permanent address
B	Billing address

You also have the option to print TO THE PARENTS OF: as the first line of the label. To add this to the label you have to type TPO=Y after the ADDRPREF option. So if you want this title your batch EX line should look like this:

EX LABTEST ADDRPREF=LP,TPO=Y

If you don't want the TO THE PARENTS OF line, just leave that option off.

Continue batching the report as you would a list report. You will notice on the Batch Options screen that the FORM is listed as 1061. This is correct. The focexec statements and other header information will print on paper and only the address labels will print on labels.

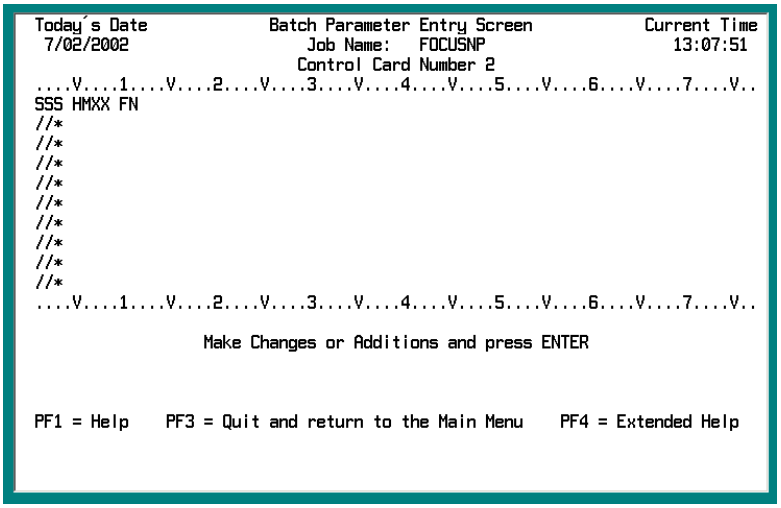
Note: If your request has **less** than 3/4 of a page of labels, the job is automatically printed on plain paper. This is done to save money since labels are very expensive.

Batching a Nonprint Focexec

If you are looking to get a subset of data from any of the systems to use with another programs such as MicroSoft Excel, you need to use the nonprint batch job. This batch job will take the output from a focexec, strip out headings, footings, and other titles, lay out the data in columns and ship the file to your TSO account.

To submit for a nonprint, follow the sign on procedures to get to batch. When the batch system asks for the name of the job to batch type FOCUSNP. The first screen displays the EX prompt, asking for the name of the focexec. Type the name of the focexec and any variables you may have added to the focexec.

The second screen to display looks like this:



You must replace the prompts with the appropriate information. SSS is replaced with the system you are generating the focexec from, either SIS, HRS, FRS. HMXX is your TSO userid, just the first 4 characters. FN is the name of the file you want to save the data to. If you were generating a file from a focexec the line on this screen should look something like this,

SIS HMCC FALL02FR

This line translates to, send a file with the name SIS.FALL02FR to HMCC.
Remember to include the correct system from where the data are coming **from**, otherwise the batch job will fail.

When you have typed the line correctly, press Enter. The system displays the *Batch Options* screen. You should change the route on this screen to your TSO account. The batch system will send the **job information** (the actual code of the Focexec, how many records it found, was it successful, etc.) to the held output queue in SDSF on your TSO account.

The actual data file (the file of information you wish to download to your PC) is put directly in your TSO library. To view it, from the main menu in TSO, choose option 0.3.4. At the "Dsname Level" prompt, enter your HMxx account code and press enter. (It is important that you type on the first FOUR characters of your TSO account at this prompt to display your data files). This will list all of your files. Type B next to the file to browse it. You can now download the data to your PC for further use.

Tips and Tricks

1. Check the EX statement for accuracy before pressing Enter. Once you press Enter on that screen the job will run and fail if the variables or the focexec name was wrong.
2. Always check your batch header information to determine why your focexec did not run. Most common mistakes include, misspelling the focexec name, leaving out variables and parameters that the focexec needs to run.
3. If you have on-line access, test the focexec on-line first. When you're happy with the format and the data, run it in batch.
4. Write down the batch job number displayed on the Confirmation screen. This will speed up the process if the Help Desk has to track down your output.

XXIII. Creating Labels

A "template" has been created by Information Services for you to use when you wish to create mailing labels in SIS.

1. Begin by naming (creating) a new Focexec.
2. Move your cursor to the Command Line or press F12 to move to the Command Line.
3. Type GET FOCEXEC(LABELEX) Press Enter.
4. This will copy the template into your new file. The very bottom of this file will be displayed. You want to move to the top of the file. Type TOP on the Command Line and press Enter to move to the top.
5. The first change you want to make is to the statement that reads -INCLUDE VIEW. Change VIEW to:

AAALABEL	Enter AAALABEL if you are using data from the AAFILE only.
MMMLABEL	Enter MMMLABEL if you are using data from the RTFILE, AAFILE, MMFILE, and/or RAFILE. It joins all of these files together.
RRTLABEL	Enter RRTLABEL if you are using data from the RTFILE, AAFILE, and/or RAFILE. It joins all of these files together for you.
SSALABEL	Enter SSALABEL if you are using data from the SWFILE, AAFILE, and/or SAFILE. It joins all of these together.

6. Enter your sort criteria (BY phrases) based on the VIEW you have chosen between the dashed lines in the template. Note: you must include NOPRINT after each BY phrase. For example:

   BY ZIP NOPRINT
   BY AA005 NOPRINT

   Please Note: A field named ZIP has been created for this label job. When you wish to sort by zipcode in SIS, include BY ZIP in this section of the template.

7. Enter your selection criteria (based on the view you have chosen) between the dashed lines in the template. For example:

   IF RT020 GE 12
   IF RT14C EQ 'FR'

Note: If the line IF RT005 EQ 'xxxxx' is displayed in your focexec, delete it. When you batch this job, it will automatically pull students from the current semester. If you wish to override the current semster, you will do this in BATCH by adding ,TERM=xx**xxx** after the ADDRPREF option. (Replacing **xxx** with YearYearSemester (2001S) with the semester you want).

8. Do not change any other lines in the template. SAVE your changes to the Focexec by typing FILE on the Command Line.
9. Refer to Printing section for instructions for batching your Focexec.

Important: You may not run your labels on-line. They must be batched.

XXIV. Printing the Long Translation of a Field

Sometimes, when you are printing a field on your report, you would like to print the LONG version of the value of the field instead of the CODE for the field. For instance, you may want to print CRIMINAL JUSTICE instead of CRJU for a student's major, or perhaps you would like to print Female/Male for a student's gender rather than F or M. You can do this by printing the long translation of a field's value. This chapter will cover printing long translations for SIS, FRS, HRS and ADS.

When you print the long translation for a field's value, you are using a Subroutine called XSFOCT. The correct syntax for using XSFOCT is:

XSFOCT (length,'element id', data value, default translation, output)

Parameter	Format	Description
Length	Numeric	What is the maximum length Focus should allow to print the long translation? (Maximum is 50.)
Element ID	Alpha	The DBD element ID. For example RT120 is the element to be translated. The Element ID has a maximum value of 5 characters. Enclose the Element ID in apostrophes.
Data Value	Alpha	The field name where the value is stored. This field has a maximum of 15 characters.
Default Translation	Alpha	Here you tell Focus what to print if Focus does not find a long translation for the field you have specified.
Output	Alpha	This is a new field which you create. It is where Focus puts the long translation. It is also what you use when printing. As with any new field, you can only specify 8 characters for this new field name. (It's very similar to a DEFINEd field.)

Important Note:

When you use XSFOCT and run your Focexec on-line, you may want to include a RECORDLIMIT or READLIMIT. This depends on how you constructed your Focexec. If you are defining this field on a smaller hold file, then you can probably run it on-line with only a slight difference in wait time. However, if you are not joining any files and are simply including the DEFINE for the entire database file, then you definitely want to just test the translation using READLIMIT or RECORDLIMIT, then batch your report when you are happy with the results (taking the limit off, of course).

SIS XSFOCT Example

The following Focexec will print the long translation of STATE and MAJOR.

```
TABLE FILE RTFILE
PRINT RT120
BY RT010
IF RT005 EQ &SEM
IF RT14E EQ 'U'
IF RT020 GE 1
ON TABLE HOLD AS CURRSTUD
END

JOIN RT010 IN CURRSTUD TO KEY IN AAFILE AS J1

DEFINE FILE CURRSTUD
TRANS_DEF/A50 = 'NO SIS TRANSLATION';
SIS_TRAN/A50 = XSFOCT (50,'AA016', AA016, TRANS_DEF, SIS_TRAN);
MAJ_TRAN/A50 = XSFOCT (50, 'RT120', RT120, TRANS_DEF, MAJ_TRAN);
END

TABLE FILE CURRSTUD
PRINT AA003 SIS_TRAN AS 'STATE' MAJ_TRAN AS 'MAJOR'
BY AA002 BY AA005 NOPRINT
END
```

An example of the report this exec would create:

Student Number	Student Name	STATE	MAJOR
444553333	Hamond, Jeffrey M	New Hampshire	Business Administration/Marketing
555446666	Ilian, Heather A	New Hampshire	Biology
666778888	Jurkin, Amy J	New Hampshire	Medical Technology

HRS XSFOCT Example

The following Focexec will print the long translation of GENDER and ETHNIC CODE.

```
DEFINE FILE EAFILE
TRANS_DEF/A50 = 'NO HRS TRANSLATION';
SEX_TRAN/A50 = XSFOCT (50, 'EA016', SEX_EA, TRANS_DEF, SEX_TRAN);
ETH_TRAN/A50 = XSFOCT (50, 'EA018', ETHNIC_CD, TRANS_DEF, ETH_TRAN);
END

TABLE FILE EAFILE
PRINT SEX_TRAN AS 'Gender' ETH_TRAN AS 'Ethnicity'
BY EMP_ID_100
END
```

An example of the report this exec would create:

Employee ID	GENDER	ETHNICITY
111223333	FEMALE	WHITE - NON-HISPANIC
222334444	NO HRS TRANSLATION	UNSPECIFIED
333445555	MALE	UNSPECIFIED
444556666	NO HRS TRANSLATION	UNSPECIFIED

ADS XSFOCT Example

The following Focexec will print the long translation of CFAE TYPE.

```
DEFINE FILE LCFILE
TRANS_DEF/A50 = 'NO ADS TRANSLATION';
TYP_TRAN/A50 = XSFOCT(50, 'LC024', CFAE_TYP_LC, TRANS_DEF,
TYP_TRAN);
END

TABLE FILE LCFILE
PRINT TYP_TRAN AS 'CFAE TYPE TRANSLATION'
BY ORG_NAME AS 'ORGANIZATION NAME'
END
```

An example of the report this Focexec would create:

ORGANIZATION NAME	CFAE TYPE TRANSLATION
A Company	Corporation
B Company	Corporation
C Corp.	Corporation
D Inc.	Corporation

FRS XSFOCT Example

The following Focexec will print the long translation of DEPARTMENT and FUND GROUP.

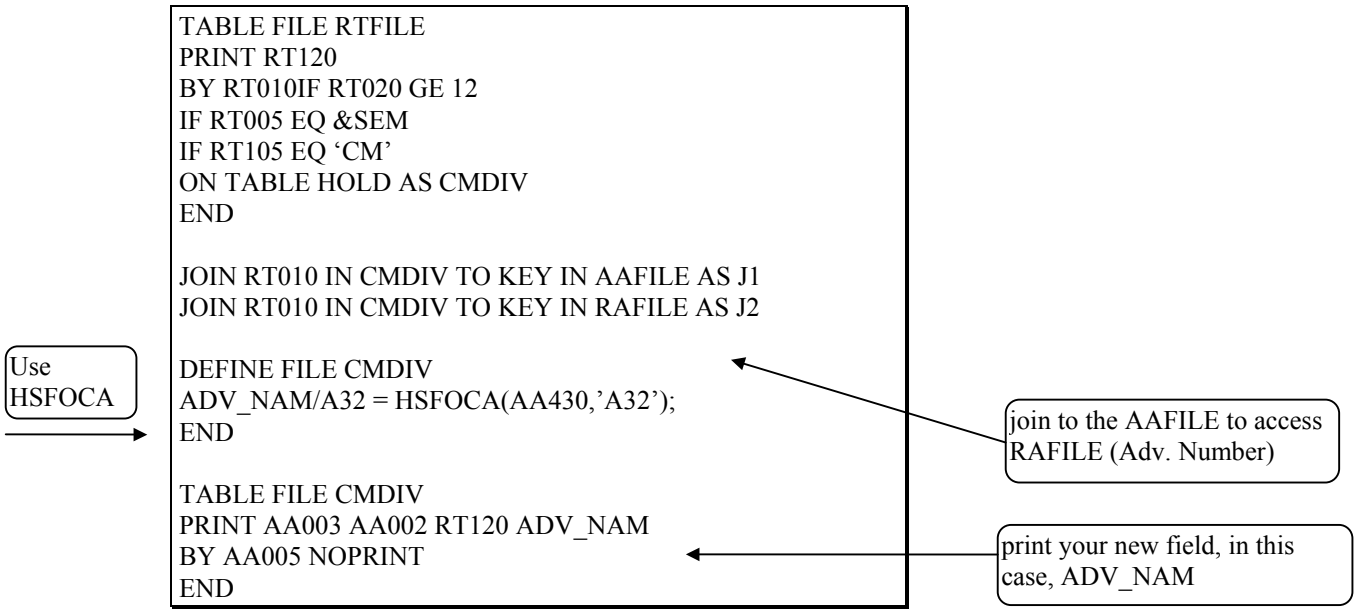
```
DEFINE FILE FGFILE
TRANS_DEF/A50 = 'NO FRS TRANSLATION';
DEP_TRAN/A50 = XSFOCT (50, 'FG054', FG054, TRANS_DEF, DEP_TRAN);
FDG_TRAN/A50 = XSFOCT (50, 'FG066', FG066, TRANS_DEF, FDG_TRAN);
END

TABLE FILE FGFILE
PRINT FG042 DEP_TRAN AS 'DEPARTMENT' FDG_TRAN AS 'FUND GROUP'
BY FG005
IF FG005 GT 000099
END
```

An example of the report this Focexec would create:

Account Digits 1-6	Account Description	DEPARTMENT
xx4433	REV & EXP SUMMARY	Business Office
xxx543	GOVT APPROPRIATIONS	Business Office
xx7896	GOVT GRANTS/CONTRACT	Business Office

XXV. Advisor Name in SIS

To include the Advisor Name in your Report, you must first have joined the AAFILE because the AAFILE contains the Advisor Number (AA430). Once you have joined to the AAFILE, define a new field which will print the Advisor Name BEFORE including the TABLE FILE... section of your Focexec as follows:



XXVI. Appendix A—MFD Examples

AAFILE

Description

You should use this MFD only as an example of how to read the MFD. The most up to date MFDs will be distributed in class. (Spring 2001)

The AAFile contains biographic and demograghic information about a student. FOCUS users at Marist access the AAFile primarily to print out student names. One of the goals of designing an efficient database is to cut down on redundancy of data. In SIS, this is why the AAFILE is where a student first is entered. His/her ID, Name, other biographic and demographic information is entered. From that point onward only the student's id number is carried along to the other files. It is the link between their static information (id, name) and their variable information such as the courses they will take during their career, their declared major(s), etc. To pull all of this information together, many times you will be joining different segments of the student's record together. You will pull in the AAFILE most often to simply obtain the student's name.

The initial segment in the AAFILE is called the ROOT. All information in this segment is unique to the student. The ROOT segment contains the KEY to the file called the Primary Key. The key is made up of one field: AA002. AA002 is the student's id number. This field uniquely identifies each record in the AAFILE.

You can determine that AA002 makes up the key by looking at the description of the Group: ROOT_KEY. It's an alphanumeric field, 9 characters long. The key is usually at the top of the file. Look at the descriptions of the first field: The field is Alphanumeric and 9 characters long. It makes up the key.

Segment Name:

Description of Key:

SEGNAME=ROOT	SEGTYPE=S0			
GROUP=ROOT_KEY	ALIAS=KEY	A09	A09	
	Usage	**Actual**	**Long Field Name**	**Title (which Prints)**
AA002	A09	A09	STU_ID	"Student,Number"
AA003	A32	A32	STU_NAME	"Student Name"
AA004	A52	A52	STU_NAME_LONG	"Long Name"
AA005	A16	A16		"Sort Name"
AA007	A01	A01	SPCL_NAME_FLAG	"Spcl,Name"
AA008	A04	A04	STU_SALU	"Salut"
AA009	A09	A09	STU_ID_OLD_SYS	"Old SID"
AA01A	A09	A09	STU_ID_PREV	"Previous, SID"
AA01B	A32	A32	PREV_NAME_1	"Previous, Name"
AA01C	A32	A32	PREV_NAME_2	"Previous, Name 2"
AA01D	A32	A32	PREV_NAME_3	"Previous, Name 3"
AA01E	A20	A20	BIRTH_CITY	"Birth City"

AA01F	A02	A02	BIRTH_STATE	“Birth, State”
AA01G	A02	A02	BIRTH_COUNTRY	“Birth, Cntry”
AA010	P08YYMD	P05	BIRTH_DT	"Birth Dt"
AA011	A01	A01	SEX	"Sex"
AA012	A01	A01	MARITAL_STAT	"Mrtl,Stat"
AA013	A01	A01	ETHNIC_ORIGIN	"Ethnic,Origin"
Field Name	**Usage**	**Actual**	**Long Field Name**	**Title (which Prints)**
AA014	A02	A02	SPECIAL_TAG	"SPTag"
AA015	A02	A02	CITIZENSHIP	"Citzn"
AA016	A02	A02	STATE_ORIG	"State, Origin"
AA017	A03	A03	COUNTY_ORIGIN	"County"
AA018	A02	A02	FILLER	
AA019	A01	A01	INFO_RLSE_FLAG	"Info,Rlse"
AA02A	P08YYMD	P05	RESID_QUEST_DT	“Resid,Quest,Date”
AA02B	A01	A01	RESID_QUEST_RCVD	“Resid,Quest,Rcvd”
AA02C	A11	A11	RESID_NUM	“Residency,Number”
AA02D	A02	A02	COUNTRY_RESID	“Cntry,Resid”
AA02E	A05	A05	HANDICAP_PERMIT	“Hndcp,Permit”
AA02F	A08	A08	HANDICAP_LIC	“Handicap,License”
AA02G	A02	A02	HANDICAP_LIC_STATE	“Hncp,Lic,St”
AA02H	A32	A32	HANDICAP_ATTNDNT	“Handicap Attendant”
AA020	A01	A01	HANDICAP_TYPE	" Hncp,Type"
AA021	A02	A02	EMPLOYER	"Employer"
AA022	P08YYMD	P05	VISA_EXPR_DT	"Expire,Date"
AA023	A01	A01	FRGN_STU	"Frgn,Stu"
AA024	A01	A01	COOP_STU	"Coop,Stu"
AA025	A02	A02	VET_CD	"Student,Type"
AA026	A02	A02	VET_BEN	"Location"
AA027	A09	A09	VET_NUM	"Claim Num"
AA028	P08YYMD	P05	VET_EXP	"Benefits, Expire,
AA029	A03	A03	VET_TR	“Beg,Sem”
AA030	A01	A01	VET_TYP_AA	"Vet Flag"
AA031	A32	A32	NOK_NAME_AA	"NOK Name"
AA032	A02	A02	NOK_RLTN_AA	"NOK,Rltn”
AA033	A10	A10	NOK_PHONE_AA	“NOK Phone”
AA034	P05.1C	Z04.1	ENTR_DT_AAA	“Step”
AA035	A01	A01	TONGUE_AA	“Cert,Code”
AA036	A01	A01	T2202A_AA	“T2202A,Override”

AA037	A02	A02	CNDA_STAT_AA	“Curr,Stat”
AA038	A02	A02	CNDA_YR_AA	“Last,Year,Rptd”
AA039	A01	A01	CNDA_RPT_AA	“Last Rpt,Full/Part”
AA040	A02	A02	CNDA_LVL_AA	"Last,Lvl,Rptd"
AA041	A03	A03	CNDA_DEG_AA	"Last Grad,Deg"
AA042	A04	A04	CNDA_MJR_AA	"Last Grad,Maj"
AA043	A01	A01	CNDA_FEE_AA	"Fee,Cat"
AA044	A03	A03	CNDA_FRML_AA	"Term"
AA045	A01	A01	CNDA_IMM_AA	"Immig,Status"
AA046	A03	A03	ROSTER	"Roster No."
AA047	A04	A04	COLL_CODE	"College Code"
AA048	A04	A04	PERS_ID_AA	"Pers, ID"
AA049	A02	A02	NCAA_ATHL_AA	"NCAA,Code"
Field Name	**Usage**	**Actual**	**Long Field Name**	**Title (which Prints)**
AA050	P07.3C	Z06.3	NCAA_GPA_AA	"NCAA,GPA"
AA051	A06YMD	A06	PASPRT_DT_AA	"Matric,Date"
AA052	A10	A10	WRK_PHON_AA	"Student,Work Phone"
AA053	A15	A15	NICKNAME_AA	"Nickname"
AA054	A01	A01	DRAFT_REG_AA	"Draft, Reg"
AA056	A02	A02	HOLD_FLG1	"Bursar"
AA057	A02	A02	HOLD_FLG2	"Perkins Default"
AA058	A02	A02	HOLD_FLG3	"Fin Aid Trans"
AA059	A02	A02	HOLD_FLG4	"Academic Dis"
AA060	A02	A02	HOLD_FLG5	"MMR"
AA061	A02	A02	HOLD_FLG6	"Contact ACA off"
AA062	A02	A02	HOLD_FLG7	"Commencement"
AA063	A02	A02	HOLD_FLG8	"Hold,Flag,8"
AA064	A02	A02	HOLD_FLG9	"Hold,Flag,9"
AA065	A02	A02	HOLD_FLG10	"Bad SS#"
AA066	A02	A02	HOLD_FLG11	"Acad Prob Flag"
AA067	A02	A02	HOLD_FLG12	"Hold,Flag,12"
AA068	A02	A02	HOLD_FLG13	"Hold,Flag,13"
AA069	A02	A02	HOLD_FLG14	"Hold,Flag,14"
AA070	A02	A02	HOLD_FLG15	"Hold,Flag,15"
AA071	A02	A02	HOLD_FLG16	"Hold,Flag,16"
AA072	A02	A02	HOLD_FLG17	"Hold,Flag,17"
AA073	A02	A02	HOLD_FLG18	"Hold,Flag,18"
AA074	A02	A02	HOLD_FLG19	"Hold,Flag,19"

AA075	A02	A02	HOLD_FLG20	"Hold,Flag,20"
AA076	A02	A02	HOLD_FLG21	"Hold,Flag,21"
AA077	A02	A02	HOLD_FLG22	"Hold,Flag,22"
AA078	A02	A02	HOLD_FLG23	"Hold,Flag,23"
AA079	A02	A02	HOLD_FLG24	"Hold,Flag,24"
AA080	A02	A02	HOLD_FLG25	"Hold,Flag,25"
AA081	A02	A02	HOLD_FLG26	"Hold,Flag,26"
AA082	A02	A02	HOLD_FLG27	"Hold,Flag,27"
AA083	A02	A02	HOLD_FLG28	"Hold,Flag,28"
AA084	A02	A02	HOLD_FLG29	"Hold,Flag,29"
AA085	A02	A02	HOLD_FLG30	"Hold,Flag,30"
AA086	A01	A01	ADMIS_IND_AA	"Adms,Ind"
Field Name	**Usage**	**Actual**	**Long Field Name**	**Title (which Prints)**
AA087	A01	A01	BR_IND_AA	"Billing,Ind"
AA088	A01	A01	ACAD_AUX_AA	"Acad,Aux,Ind"
AA089	A01	A01	BAS_ADMIS_AA	Basis, of,Adms
AA090	A01	A01	CRER_STAT_AA	"Career,Stats, Ind"
AA091	A01	A01	FAM_LOC_AA	"FAM,Stu, Locr"
AA092	A02	A02	STU_SITE_AA	"Stu,Site"
AA095	A02	A02	ACTV1_AA	"Actv, 1"
AA096	A04YM	A04	ACTV1_DT_AA	"Actv 1, Date"
AA097	A02	A02	ACTV2_AA	"Actv, 2"
AA098	A04YM	A04	ACTV2_DT_AA	"Actv 2, Date"
AA099	A02	A02	ACTV3_AA	"Actv, 3"
AA100	A04YM	A04	ACTV3_DT_AA	"Actv 3, Date"
AA101	A02	A02	ACTV4_AA	"Actv, 4"
AA102	A04YM	A04	ACTV4_DT_AA	"Actv 4, Date"
AA103	A02	A02	ACTV5_AA	"Actv, 5"
AA104	A04YM	A04	ACTV5_DT_AA	"Actv 5, Date"
AA105	A02	A02	ACTV6_AA	"Actv, 6"
AA106	A04YM	A04	ACTV6_DT_AA	"Actv 6, Date"
AA107	A02	A02	ACTV7_AA	"Actv, 7"
AA108	A04YM	A04	ACTV7_DT_AA	"Actv 7, Date"
AA109	A02	A02	ACTV8_AA	"Actv, 8"
AA110	A04YM	A04	ACTV8_DT_AA	"Actv 8, Date"
AA111	A02	A02	ACTV9_AA	"Actv, 9"
AA112	A04YM	A04	ACTV9_DT_AA	"Actv 9, Date"
AA113	A02	A02	ACTV10_AA	"Actv, 10"

AA114	A04YM	A04	ACTV10_DT_AA	ExtraCurrDate
AA115	A09	A09	SPS_ID_AA	"Spouse ID"
AA116	A01	A01	ADS_PASS_AA	" ADS,Passed, Ind"
AA156	A06	A06	MAR_PO_BOX	"Marist,PO Box"
AA157	A09	A09	HMSD_ID_NO	"HMSD,ID No"
AA158	A08	A08	COMP_ACCT	"Computer,Acc"
AA159	A03	A03	LOTT_SLOT	"Lottery,Slot"
AA160	A04	A04	LOTT_NUMB	"Lottery,Number"
AA161	A03	A03	FRESH_COHRT	"Freshman,Cohort"
AA162	A03	A03	TRANS_COHRT	"Transfer,Cohort"
AA163	A01	A01	WTHDRW_TYPE	"Withdraw,Type"
Field Name	**Usage**	**Actual**	**Long Field Name**	**Title (which Prints)**
AA164	A01	A01	WTHDRW_REAS	"Withdraw,Reason"
AA165	A02	A02	WTHDRW_DORM	"Withdraw,Dorm"
AA166	A06	A06	WTHDRW_DATE	"Withdraw,Date"
AA167	A01	A01	IMMUNITY	"Immunity Status"
AA168	A01	A01	MMR_IMMNTY	"MMR Immunity"
AA169	A01	A01	MSLS_IMMTY	Measles Immunity
AA170	A01	A01	MMPS_IMMTY	"Mumps Immunity"
AA171	A01	A01	RBLL_IMMTY	Rubella Immunity
AA172	A01	A01	HEPB_IMMTY	"HepB Immunity"
AA173	A01	A01	DIR_DIS_FLG	Dir Release Flag
AA180	P03	A03	ADDR_CTR_AA	
AA199	A06YMD	A06	SIS_MNTDT_AA	"SIS Maint, Date"

This segment is no longer in the AAFILE. This section is only shown for an example of what segments are and how to understand them. (Spring 2001)

The second segment in the AAFILE is called the ADD_SEG. This is the Address Segment (ADD_SEG). Each student may have 10 address. Up to five address types may be inputted for each address. Dorm Local, Permanent, Billing, and User. Each address type can only be used once, therefore a student can only have one permanent address, one dorm, etc. The number of addresses that exist for each student are stored in AA180. (OCCURS =AA180 in table below) Therefore, if a student has two addresses (one dorm one billing), then AA180 will be equal to 2. Information Services has provided you with PSEUDO fields, however, to handle addresses. Instead of this segment and it's fields, you will be using PSEUDO fields (ones we have created) which begin with AP. Therefore, the following information is provided for example of reading an MFD (Master File Description) only.

SEGNAME=ADD_SEG	SEGTYPE=S0	PARENT=ROOT	OCCURS=AA180	
Field Name	**Usage**	**Actual**	**Long Field Name**	**Title (which Prints)**
AA205	A01	A01	ADDR_DEL_AA	"Addr,Del"
AA210	P02	A02	ADDR_NUM_AA	"Addr,Num"
AA215	A01	A01	ADDR_TYP1_AA	"Addr,Type,1"
AA220	A01	A01	ADDR_TYP2_AA	"Addr,Type,2"
AA225	A01	A01	ADDR_TYP3_AA	"Addr,Type,3"
AA230	A01	A01	ADDR_TYP4_AA	"Addr,Type,4"
AA235	A01	A01	ADDR_TYP5_AA	"Addr,Type,5"
AA240	A32	A32	STREET1_AA	"Street 1"
AA245	A32	A32	STREET2_AA	"Street 2"
AA250	A13	A13	CITY_AA	"City"
AA255	A02	A02	STATE_AA	"State"
AA260	A09	A09	ZIP_AA	"Zip Code"
AA263	A02	A02	CNTRY_AA	"Country"
AA265	A10	A10	PHONE_AA	"Phone"

GROUP_01 is a segment used for HOLD FLAGS. Hold Flags begin at field AA056 and continue for 30 fields—to AA085. Instead of searching every hold flag by field name (because you are not sure what hold flags a student might have) you can use HOLD_FLG_ORD (or the Alias ORDER) to sweep all hold flags searching for anyone who has a hold flag.

AA_020_ORDER	I03	I04		
SEGNAME=GROUP_01	SEGTYPE=S0	PARENT=ROOT	OCCURS=30	POSITION=AA056
AA056	A02	A02	HOLD_FLG_AA	"Hold,Flag "
HOLD_FLG_ORD	ALIAS=ORDER	I03	I04	

GROUP_02 stores activity information about the student. For every activity in the SIS system for this student, there exists two parts: The activity number and the activity date. This segment is not used at this time.

SEGNAME=GROUP_02	SEGTYPE=S0	PARENT=ROOT	OCCURS=10	POSITION=AA095
AA095	A02	A02	ACTV_AA	"Actv"
AA096	A04YM	A04	ACTV_DT_AA	" Actv, Date"
ACTV_ORDR	I03	I04		

GROUP_03 allows you to sweep the Address Segment—ADD_SEG looking at the five address type fields looking for a specific value. For instance, to determine if a student has a dorm address value use AA215.

SEGNAME=GROUP_03	SEGTYPE=S0	PARENT=ADD_SEG	OCCURS=05	POSITION=AA215
AA215	A01	A01	ADDR_TYP_AA	"Addr,Type"

XXVII. Appendix B

Common Fields used in SIS

Field Description	Field Name	Field Description	Field Name
Student ID Number	Different in every file	**Degree Code, Awarded**	RA205
Student Name	AA003	**Degree GPA (not the same as CUM GPA, see RP460)**	RA260
Name Sort	AA005	**High Sch. Code (CEEB)**	RB111
Birth Date	AA010	**High Sch. Grad. Date**	RB120
Sex, M or F	AA011	**High Sch. Rank**	RB140
Ethnic Origin	AA013	**Test Score Code 1-5**	RB205
Special Tag	AA014	**Advisor Soc.Sec. No.**	AY005
Citizenship	AA015	**Degree Expected Term**	AA661
County of Origin	AA017	**Cum. Earned Hours**	RA320
Info. Release Flag	AA019	**Cum. GPA**	RA380
Student Type	AA025	**Term (Semester)**	RT005
Location	AA026	**# of credits student is taking for a specific semester**	RT020
Beginning Semester	AA03A	**# of credits student earned for a specific semester**	RT025
NCAA Athletic code	AA049	**GPA, for a specific Semester**	RT040
Matric Date	AA051	**Academic Action**	RT080
Work Phone	AA052	**Withdraw Code**	RT095
Academic Dismissal	AA059	**Career (U or G)**	RT14E
		College (Entire Division)	RT105
On Campus PO Box	AA156	**Class Yr. (FR SO JR SR)**	RT14C
Computer Account	AA158	**Primary Major**	RT120
Admit Type	MM520	**Primary Minor**	RT130
Applied for Level (Transfers, Grad, etc.)	MM466	**Certificate**	RT135
		Graduation Code	RT165
		Dorm Code	RT185
		Residency, Term	RT205

Common Fields used in ADS

Field Description	Field Name	Field Description	Field Name
Constituent/Alumni ID Number	**CID_LC, CID_LG, CID_LP, CID_LH, (Add filename to CID_)**		
Constituent Name	**Various fields are in LCfile**		
Constituent Type	**PREF_CONST**		
Activity Status	**ACTV_STAT_LC**		
Occupation Type	**OCC_TYP_LC**		
Gender	**SEX_LP**		
Alumni Graduation Year	**REUN_YR_1**		
Campaign	**CMPGN_CD_LH**		

Index