

USING CRSP AND COMPUSTAT

AN INTRODUCTION

Prepared by:
Patricia Ledesma
Michael Faulkender

January 31, 2001

1. Kellogg Research Computing
2. Wharton Research Data Services
3. Compustat
4. CRSP
5. Skew3
6. Basic Unix commands
7. SAS
8. Using Fortran sample programs
9. Essential FTP Commands
10. WRDS Data files

1. Kellogg Research Computing

Web site www.kellogg.nwu.edu/kis/research/
New www.kellogg.nwu.edu/researchcomputing/
E-mail researchcomputing-help@kellogg.nwu.edu
 (messages reach Jian Guo and Patricia Ledesma)
Unix server: skew3.kellogg.nwu.edu (or skew3.kellogg.northwestern.edu)

2. Wharton Research Data Services (WRDS)

Web site: wrds.wharton.upenn.edu
Unix host: wrds.wharton.upenn.edu
E-mail support: wrds-support@wharton.upenn.edu
Account request: From the main WRDS web page, click on “Account Request”
 (second to last item on left menu)
Sample programs: Can be found in the samples directory for each dataset in wrds.
Online help: From the main page, click on HELP.
 The Help menu will appear on the left, including links to the
 Documentation and Manuals, as well as Frequently Asked
 Questions about WRDS and SAS

Searching for PERMNO and CUSIP in CRSP or for CNUM and DNUM in Compustat:

At the prompt in the WRDS Unix server, use the following commands:

For CRSP:

```
grep -i “company name” /wrds/crsp/seqdata/msf.names
```

For Compustat:

```
grep -i “company name” /wrds/compustat/seqdata/ina.names  
(or res.names or fca.names)
```

Examples:

```
grep -i "ibm" /wrdsx/crsp/seqdata/msf.names
```

```
12490 45920010 3571 IBM INTERNATIONAL BUSINESS MACHS COR 19251231-199911  
75139 03093810 6799 BZP AMERICUS TR FOR IBM SHS 19870731-199200  
75140 03093820 6799 BZS AMERICUS TR FOR IBM SHS 19870731-199209  
75141 03093830 6799 BZU AMERICUS TR FOR IBM SHS 19870731-199209
```

The columns, left to right, are: PERMNO, CUSIP, Header SIC code, ticker, company name, start to end date (end date is truncated on the screen).

```
grep -i "ibm" /wrdsx/compustat/seqdata/ina.names
```

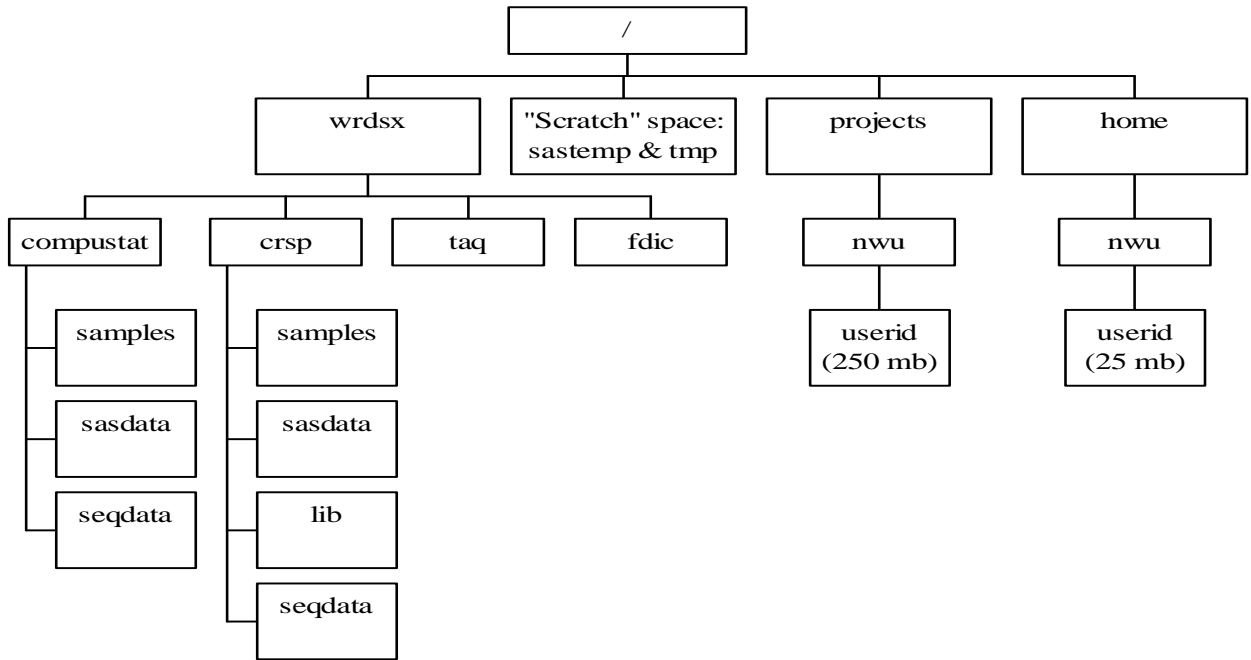
Kellogg Research Computing

7370 459200 101 IBM INTL BUSINESS MACHINES CORP

The columns, left to right, are: DNUM (industry classification code), CNUM (CUSIP issuer code), CIC (CUSIP issuer number and check digit), SMBL (ticker symbol) and company name.

WRDS directories

Each user gets two directories, a “home” (with 25MB of space), which is what the user sees by default on login, and a “projects” directory (250MB of space).



3. Compustat

Standard & Poor's Compustat provides the annual and quarterly Income Statement, Balance Sheet, Statement of Cash Flows, and supplemental data items on most publicly held companies in North America. Financial data items are collected from a wide variety of sources including news wire services, news releases, shareholder reports, direct company contacts, and quarterly and annual documents filed with the Securities and Exchange Commission. Compustat files also contain information on aggregates, industry segments, banks, market prices, dividends, and earnings. Depending upon the data set, coverage may extend as far back as 1950 through the most recent year-end. Kellogg's subscription to Compustat includes the files listed below.

- **Industrial files (ina, inq):** Includes data from balance statements, income statements and cash flows for the publicly held companies listed in NYSE and AMEX for the most recent 20 years
- **Full coverage files (fca, fcq):** Includes companies listed in NASDAQ, regional exchanges, publicly held companies trading common stock and wholly owned subsidiaries trading preferred stock or debt.
- **Research files (res, req):** Contain companies that have been deleted from the Industrial files [Primary, Supplementary, Tertiary] and the Full-Coverage files due to acquisition, merger, bankruptcy, liquidation, reverse acquisition, leveraged buyout, or because they became a private company. The "Current" data file covers the most recent 20 years and is updated annually. There are two additional research data files, one that covers the previous 20 years (currently, 1961-1980, updated annually, called "Backdata") and one that covers 1950 through 1969 ("Wayback data"). The last one is not updated.
- **Bank files (bna, bnq):** Data on about 600-700 banking institutions.
- **Business Information Industry Segment file (bif):** Contain up to seven fiscal years for each company. Within each year there are from one to ten records of industry segment data, depending on reports by companies.
- **Business Information Geographic Segment files (geo):** Contain up to seven fiscal years for each company. Within each year there are 5 records of geographic segment data.
- **Prices, Dividends and Earnings files (pde),** current and research: Contain market information and about 120 industry indexes and composites.
- **ExecuComp** (Executive Compensation)

Note: Many users frequently search three of Compustat's databases to find a specific company (industrial, research and full coverage files). Wharton has combined the three files into one in the SAS version of the data (compann.ssd01 and compqtr.ssd01).

4. CRSP

CRSP is the “nickname” for the datasets sold by the Center for Research in Security Prices, part of the School of Business at the University of Chicago.

CRSP is a collection of datasets with basic and derived information for securities traded in on U.S. exchanges (NYSE, AMEX, and NASDAQ). Monthly data starts generally in 1925, while daily data starts in 1962. Kellogg is subscribed to the following CRSP datasets:

- US Stock databases
- US Indices database and security portfolio assignment module
- US Treasury databases
- Survivor-Bias Free US Mutual Fund database
- CRSP/Compustat Merged database

The main security identifiers for CRSP are PERMNO and PERMCO (or CRSPID for the Treasury file), which are assigned by CRSP itself.

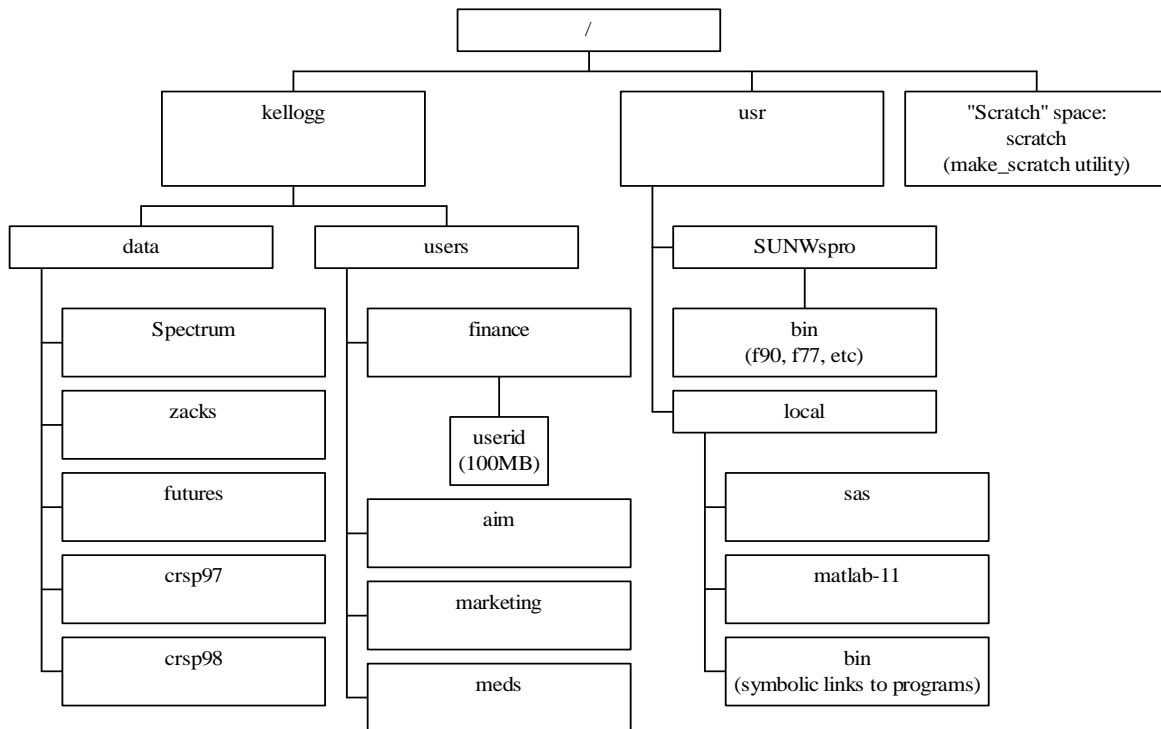
CRSP data files are split into two kinds of files: “header” and “data” files. Header files have descriptive information for each security: primary and secondary identifiers, most recent name information, exchange code, SIC code, counts of “array events” (number of name structures, number of distribution structures, etc) and the ranges for which there is a time series (begin and end to the series). The data files contain the primary identifiers and the time series.

Events files: For the stock databases, these files contain all the information about name history (effective dates and last date of names, ticker, etc), delisting (date, code, links to securities or companies that can be used to track the issue further, etc), distributions (codes describing the event, factors to adjust prices and shares, links to securities associated to the event, dates, etc), shares outstanding and group history.

5. Skew3

Skew3 is Kellogg's Unix server, a Sun ULTRASparc workstation with four CPUs, 4 GB of RAM memory and a variety of statistical and mathematical programs (as well as the standard compilers and editors): SAS, Stata, Matlab, Maple, SPSS, etc.

To work on your data, you will probably FTP your files from your WRDS account. Skew3 is generally much faster than the WRDS system. The following chart shows the main directories in skew3.



For more information about Skew3:

www.kellogg.nwu.edu/researchcomputing/skew3.htm

6. Basic Unix commands

- To log into a Unix host you need a terminal emulator, either telnet (recommended: HostExplorer), secure shell (ssh) or an X-Windows emulator (recommended: X-Win32). You also need the host IP name (example: wrds.wharton.upenn.edu or skew3.kellogg.nwu.edu) and a valid login name and password.
- Unix is case sensitive (commands and file names)
- Unix systems are hierarchical. Instead of multiple drives (as in a PC), multiple file systems are joined in a single hierarchy. In the charts for the WRDS and skew3 hosts, /kellogg/users might be on a separate physical hard drive than /kellogg/data.

<i>Commands</i>	<i>What they do</i>
quota -v	It will tell you how much space you have in your home directory and what percentage is left available. The quota is shown in KB.
man	Manual pages
pwd, whoami	Present working directory, ID of currently logged on user
ls	List space, gives a list of files in current directory
cd	Change directory
mkdir, rmdir	Create (make) a new directory, remove existing directory
cp, mv, rm	File operations: Copy, move and remove (always specify source and target)
more, head, tail	Displays contents of a file. “more” shows the entire file, screen by screen. “head” shows the first 10 lines of the file (unless the number of lines is specified with the -n option), while “tail” shows the last 10 lines of a file (-n also applies)
grep	Search a file for a given string. The -i option ignores upper and lower case distinctions
vi, pico, emacs	Text editors

The “pico” editor can be started by typing its name at the command prompt. If you type “pico filename”, where “filename” is the name of an existing file, pico will open the file for editing. If “filename” does not exist, pico will create it (if you quit without saving the new file, pico will delete it).

The following graphic shows you how pico looks when you have a file open:

```

1 - DARWIN2
File Edit Transfer Fonts Options Macro View Window Help
[Icons] [F] [F] [F] [F] [F1] [F2] [F3] [F4] [CLR] [Hand]
UW PICO(tm) 2.9 File: test.sas Modified
options pagesize=60 linesize=72 nodate;
data grades;
  input name $ exam1 exam2 exam3;
  avgscore=mean(exam1,exam2,exam3);
  if avgscore<51 then grade='f';
  else if 51<=avgscore<60 then grade='d';
  else if 61<=avgscore<70 then grade='c';
  else if 71<=avgscore<80 then grade='d';
  else grade='a';
cards;
studenta 89 95 70 80
studentb 55 66 58 79
studentc 42 33 50 40
studentd 100 40 66 89
studentf 89 95 75 85
;
proc means data=grades;
var exam1 exam2 exam3;
[Icons] Get Help WriteOut Read File Prev Pg Cut Text Cur Pos
Exit Justify Where is Next Pg UnCut Text To Spell
1 Sess-1 129.74.250.114 1 3/1

```

The last few lines show the commands available in pico. In UNIX, the caret (“^”) sign represents the control character. To execute any of the commands shown at the bottom of the pico screen, use the control (Ctrl) key followed by the character shown. For example, to read a file into pico, press “Ctrl-R”.

If “pico” requires a confirmation or further information from you to execute the command, it will prompt you in the bottom of the window. Further “control commands” may appear below that prompt. A few critical commands follow:

<i>To...</i>	<i>Press...</i>
Save and exit pico	^X
Save without quitting pico	^O
Cut one line	^K
Paste previously cut line	^U
Select a block of lines	^^ (CTRL-SHIFT-^)
Go one page down	^V
Go one page up	^Y
Go to the top of file	^W^Y
Go to bottom of file	^W^Y
Insert a file in file currently being edited	^R

If you ask for the help pages in pico (^G), you can learn about other commands.

More UNIX skills:

- **Wild cards:** you can use wild cards to refer to collections of files. The asterisk (“*”) replaces any number of characters (for example, “ls *.txt” would list all the files whose name ends with “.txt” in the current directory), while the question mark (“?”)

replaces only one character (e.g., “file?.txt” can stand for file1.txt or filea.txt, but not for file12.txt).

The tilde (“~”) helps you move quickly to someone’s home directory. By itself, it refers to your home directory (for example, “cd ~” would take you back to your home directory). Followed by a netid, it will take you to the specified netid’s home directory. In skew3, “cd ~ple531”, for example, will take you to the “ple531” home directory. This will work if the home directory in question allows other users to browse the contents.

- **Redirecting output:** Any command in Unix has a default device for output and a default device for input. For example, when you ask for a file list (“ls”), the output is the screen, while the command takes its input from the current directory (disk). Unix allows you to change the input and output devices of commands. For example, instead of showing the list of files on screen, you may want to save that list in a file. You could modify the command as follows: “ls > filelist.txt”.
- **Pipes:** You also may use the output of one command as the input for a subsequent one. This can be done using UNIX “pipes”. The pipe is denoted by the vertical bar. For example:

ls | more

Sends the result of “ls” (a list of files) to the “more” command. The “more” command shows a file (or in this case, the output of “ls”) screen by screen.

- **Repeating commands:** Your Unix session keeps a history of commands given, which you can see with the “history” command. Any of them can be repeated by typing “!#”. Where “#” is the command number in the command history

Also “!x” will repeat the last command that started with an “x”

“!!” will repeat the previous command

- **Running SAS syntax:** The command is:

sas mysyntax

If the extension of “mysyntax” is “sas” (i.e., mysyntax.sas), there is no need to type it. More on SAS in section 7.

- **Running Fortran programs:** See section.8.
- **Compression:** You can save considerable time when you FTP datasets by compressing them before transferring. Use either the “compress” or “gzip” commands.

7. SAS

Check: www.kellogg.nwu.edu/researchcomputing/sas.htm

- All SAS statements end with a semi-colon.
- SAS statements are not case sensitive; spacing does not matter. Do not include tabs in a program file.
- There are two basic types of statements: DATA ‘steps’ and PROC (procedures). The DATA step is where you manipulate the data (creating variables, recoding, subsetting, etc). PROCs are used to run statistics on existing datasets and, in turn, can generate datasets as output.
- SAS libraries are directories that contain SAS data files. They are referenced with “LIBNAME” statements, in which a library reference (“libref”) is assigned to a given directory. Think of the libref as a nickname for the directory.
- When you run a SAS program, SAS creates a log file (where it writes the commands as it reads them, and any warnings or errors) and a listing file, where it writes any output requested by the program. These files have the same names as the program file, but with extensions “log” and “lst”, respectively.
- In WRDS accounts, there is a file called “autoexec.sas” which assigns a number of librefs for different dataset collections: CRSP, Compustat, FDIC, etc.

WRDS runs SAS version 6.12, while SKEW3 runs SAS version 8.1. There is essentially no difference between the two (there are more options in 8.1), except that the SAS file format is different. Version 6.12 files have the extension “ssd01”, while version 8.1 files have extension “sas7bdat”. Version 8 can read 6.12 datasets when you specify the version (the SAS “engine”).

System options:

To make SAS output easier to read on screen, use the following options: “linesize=76” (76 is the number of characters per line; the option is abbreviated as “ls”) and “nocenter”.

Reading SAS data files:

All the files in WRDS for SAS are already in SAS format. To access SAS system files, you must issue a LIBNAME statement and use a two-level name for the dataset. The first level is the library name; the second level is the file name (without extension). One-level names are reserved for temporary data files, which exist only for the duration of the SAS session.

For example, to create a subset of “inames.ssd01” from /wrdsx/compustat/sasdata, you can use the following commands:

```
options ls=76 nocenter nodate;
libname cstat '/wrdsx/compustat/sasdata';
data test;
  set cstat.inames;
  where SMBL in ('IBM' 'BVA' 'AU' 'CBJ');
```

```
proc print data=test;
```

The library reference (“cstat” in the LIBNAME statement) is arbitrary, up to 8 character. The WHERE statement allows you to subset selected observations. In this case, it was done by ticker symbol.

Creating a SAS data file:

To write a SAS data file to your account, use the LIBNAME statement and then name the dataset. This is enough to create it. Example:

```
LIBNAME mywork '/projects/nwu/pledasma';  
data mywork.subset;  
set comp.inanames;  
  where SMBL in ('IBM' 'BVA' 'AU' 'CBJ');
```

This will create a dataset called “subset.ssd01” in /projects/nwu/pledasma.

Creating an ASCII output file:

After creating a subset and manipulating a data file ins SAS, you may want to output the resulting dataset to an ASCII file in order to use it in another package (e.g., Matlab). Use the FILE and PUT statements. The example below outputs four variables to a file called “spectruminst.dat”. The \$ after CUSIP indicates it is a character variable (string).

```
data temp;  
  set inststat;  
  file 'spectruminst.dat';  
  put cusip $ year totsh noinst;
```

Alternatively, you can create a “CSV” file (comma-separated-values), which is easy to read in Excel:

```
proc export data=temp  
  outfile="spectruminst.csv"  
  dbms=csv  
  replace;
```

Reading ASCII data:

SAS has an immense number of options to read ASCII data (“raw data”). Here is just a simple DATA step example to reading a data file in which the data is organized in columns:

```
filename raw '/kellogg/data/Spectrum/S3438112.DTA';  
data file3;  
  infile raw;  
  input cusip $ 1-8 mgrno 9-13 type 14 shares 15-25;
```

Reading data typed within the program:

If you only have a few observations to input in your program, you may do it within the program. Here is an example:

```
data oranges;  
  input variety $ flavor texture looks;  
  cards;  
navel 9 8 6  
temple 7 7 7  
valencia 8 9 9
```

```
mandarin 5 7 8  
;
```

Note that the semi-colon following the data is in the line after the last observation. This is one of the few cases where this matters (the column does not matter, but it must be in the next line).

Showing the contents of a SAS data file:

This command is extremely useful, since the files in WRDS already have variable labels. To show the list of variables included in the Compustat “inanames.ssd01” file, the following command is enough:

```
proc contents data=comp.inanames;
```

“Printing” a dataset:

PROC PRINT is another very useful command to check your data. It allows you to print a dataset (or a subset) to the output file. The following example shows only the first 3 observations for two variables of the “organs” data set:

```
proc print data=oranges (obs=3);  
var variety looks;
```

Appending datasets:

To “stack” datasets with the same (or very similar) structures, the SET statement is enough. For example, if you have extracted TAQ data for January and February 2000 and would like to have the data in a single file:

```
data all;  
set jan feb;
```

Merging datasets:

If you want to put datasets “side by side”, you can use the MERGE statement. There are several things to consider before you do this: if you are merging by a certain variable (for example, by PERMNO or by CUSIP), the datasets have to be sorted by this variable. You can also merge by several variables (example: by CUSIP and date).

Many of the sample programs in WRDS use PROC SQL. In many instances, it is a very efficient way of merging (if you are familiar with SQL), except that it does not perform certain types of merges. In SQL lingo, ‘outer joins’ are a problem (cases in which values in one dataset or ‘table’ do not have a counterpart in the other dataset).

```
proc sort data=tmp1; by gvkey;  
proc sort data=tmp2; by gvkey;  
data outhere.test;  
merge tmp1 tmp2;  
by gvkey;
```

Contents of TREASURY HEADER file 1

CONTENTS PROCEDURE

File Name: /wrdsx/crsp/sasdata/mbmhdr.ssd01

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Informat	Label
6	COUPRT	Num	8	40	7.3	7.3	Coupon Rate (percent per annum)
1	CRSPID	Char	15	0			CRSP Issue Identification Number
2	CUSIP	Char	8	15			CUSIP Number
15	IDTBNK	Num	8	98	DATE9.		Bank Eligibility Date at Time of Issue
14	IDTCP	Num	8	90	DATE9.		First Call Date at Time of Issue
13	IDTDTD	Num	8	82	DATE9.		Date Dated by Treasury
21	IDTFC	Num	8	118	DATE9.		Date of First Coupon Payment
22	IDTFCF	Char	1	126			Flag for Date of First Coupon
4	IDTMAT	Num	8	31	DATE9.	YYMMDD8.	Maturity Date at time of Issue
19	IFLWR	Char	1	116			Payment of Estate Tax Code
18	ITAX	Char	1	115			Taxability of Interest
5	ITYPE	Char	1	39			Type of Issue
7	IUNIQ	Char	1	48			Uniqueness Number
12	IWHY	Char	1	81			Reason for End of Data on File
16	IYMCN	Num	8	106	YYMMN6.		Year and Month of First Call Notice
9	MFINIS	Num	8	57	3.	3.	Month Number of Last Price in Data
11	MFINISD	Num	8	73	DATE9.		Date of Last Price in Data
8	MSTART	Num	8	49	3.	3.	Month Number of First Price in Data
10	MSTARTD	Num	8	65	DATE9.		Date of First Price in Data
3	NAME	Char	8	23			Name of Government Security
20	NIPPY	Char	1	117			Number of Interest Payments per Year
17	NOTICE	Char	1	114			Notice Required on Callable Issues
23	VALFC	Num	8	127	9.6	9.6	Amount of First Coupon Payment

Contents of TREASURY HEADER file 2

CONTENTS PROCEDURE

-----Alphabetic List of Indexes and Attributes-----

#	Index
1	CRSPID

-----Sort Information-----

Sortedby: CRSPID
Validated: YES
Character Set: ASCII

Contents of TREASURY DATA file 3

CONTENTS PROCEDURE

File Name: /wrdsx/crsp/sasdata/mbmdat.ssd01

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Informat	Label
9	ACCINT	Num	8	64	8.5	8.5	Accrued Interest as of Month End
1	CRSPID	Char	15	0			CRSP Issue Identification Number
6	IOUT1R	Num	8	47	5.	5.	Face Value Outstanding
7	IOUT2R	Num	8	55	5.	5.	Publicly Held Face Value Outstanding
3	NMON	Num	8	23	3.	3.	Number of the Month
14	PCYLD	Num	8	104	8.5	E13.6	Yld to Maturity Compounded Semiannually
10	PDINT	Num	8	72	8.5	8.5	Interest Payable During Month
4	PRIC1R	Num	8	31	10.5	10.5	First Price, usually Bid
5	PRIC2R	Num	8	39	10.5	10.5	Second Price, usually Ask
2	QDATE	Num	8	15	DATE9.	YYMMDD6.	Quote Date
12	RETNUA	Num	8	88		E13.6	Unadjusted Return
13	RETNXS	Num	8	96	9.4	E13.6	Adjusted Return
8	SOURCR	Char	1	63			Primary Data Source
11	YIELD	Num	8	80	9.4	E13.6	Promised Daily Yield

-----Alphabetic List of Indexes and Attributes-----

#	Index
1	CRSPID

Contents of TREASURY DATA file 4

CONTENTS PROCEDURE

-----Sort Information-----

Sortedby: CRSPID QDATE
Validated: YES
Character Set: ASCII

```

/* Program:  bonds.sas                               */
/* Author:   Patricia Ledesma                         */
/* Date :   12/18/2000                               */
/*                               */
/* Dataset:                                         */
/* CRSP US Treasuries                               */
/*                               */
/* Description:                                     */
/* Subsets the two most recently issued 30-year    */
/* bonds each month. Each line in resulting data file */
/* represents one month. A "1" at the end of a variable */
/* name indicates a variable for the most recently issued */
/* bond in the period, while a "2" indicates the second */
/* most recently issued bond.                       */
/*                               */
options ls=76 nocenter nodate;

libname crspdata '/wrdsx/crsp/sasdata';
libname out '/projects/nwu/pledesma';

options ls=76 nocenter nodate;

/* Step 1: Using the HEADER file, produce a list of 30-year bonds */
data bondlist;
  set crspdata.mbmhdr;
  where name='BOND';
  term = year(idtmat) - year(idtdtd);
  if term=30;

/* Step 2: Using list generated in step 1, subset monthly data for */
/* the 30 year bonds since 1960, keeping selected variables.      */
data subset1
  (keep=COUPRT CRSPID IDTDTD IDTMAT ITYPE NIPPY
   ACCINT IOUT1R IOUT2R PCYLD PRIC1R
   PRIC2R QDATE SOURCR);
  merge crspdata.mbmhdr bondlist (in=a);
  by crspid;
  if a;
  if year(qdate)>=1960;

/* List of variables for reference */
proc contents data=subset1;

/* Step 3; Sort data by quote date (QDATE) and date dated by */
/* Treasury (IDTDTD); create an index to recognize bonds (QOBS). */
/* Keeping only the two most recent bonds (QOBS=1,2)          */
proc sort data=subset1;
  by qdate descending idtdtd;

data subset2;
  set subset1;
  by qdate descending idtdtd;
  retain qobs;
  if first.qdate then qobs=0;
  qobs=qobs+1;
  if qobs in(1,2);

/* Step 4: Subset all the data for bond 1 and for bond 2, then */
/* merge the files by QDATE to get the desired data file with */
/* one line of data per month.                                  */
data bond1;
  set subset2;
  where qobs=1;

```

```

rename
  COUPRT = coup1rt1
  CRSPID = crspid1
  IDTDTD = idtdtd1
  IDTMAT = idtmat1
  ITYPE = itype1
  NIPPY = nippy1
  ACCINT = accint1
  IOUT1R = iout1r1
  IOUT2R = iout2r1
  PCYLD = pcyld1
  PRIC1R = pric1r1
  PRIC2R = pric2r1
  SOURCR = sourcr1;
drop qobs;

data bond2;
set subset2;
where qobs=2;
rename
  COUPRT = coup1rt2
  CRSPID = crspid2
  IDTDTD = idtdtd2
  IDTMAT = idtmat2
  ITYPE = itype2
  NIPPY = nippy2
  ACCINT = accint2
  IOUT1R = iout1r2
  IOUT2R = iout2r2
  PCYLD = pcyld2
  PRIC1R = pric1r2
  PRIC2R = pric2r2
  SOURCR = sourcr2;
drop qobs;

data out.bonds;
merge bond1 bond2;
by qdate;

/* End */

```

```
/* Program: ccm.sas */
/* Author: Patricia Ledesma */
/* Date : 01/09/2001 */
/* */
/* Dataset: */
/* CRSP/Compustat merged database */
/* */
/* Description: */
/* Subsets all firms for certain data items and years */

options ls=76 nodate nocenter;

/* Wharton automatically assigns the CRSP libref to the */
/* wrdsx/crsp directory */

libname outhere '/projects/nwu/pledasma';

*proc contents data=crsp.headcst;
*proc contents data=crsp.cheadcst;
*proc contents data=crsp.cstann;

*proc sort data=crsp.cheadcst;
*by gvkey;
*proc sort data=crsp.cstann;
*by gvkey;

data tmp1 (keep=gvkey permno);
set crsp.cheadcst;
proc print data=tmp1 (obs=50);

data tmp2 (keep=yeara gvkey data149 fyr);
set crsp.cstann;
where 1987 <= yeara <= 1993;
proc print data=tmp2 (obs=50);

proc sort data=tmp1;
by gvkey;

proc sort data=tmp2;
by gvkey;

data outhere.test;
merge tmp1 tmp2;
by gvkey;

proc print data=outhere.test (obs=50);
```

8. Using Fortran sample programs

Sample programs:

`/wrds/compustat/samples`

`/wrds/crsp/samples`

There are some interesting rules to be aware of when programming in Fortran:

1. A character in the first column makes the row a comment line.
2. Line numbers go in the first five spaces of a line
3. Anything in the 6th space makes the line a continuation of the previous line.
4. Commands are confined to the space between column 7 and 72.
5. All variables must be declared at the beginning of the program.
6. Input and Output lines with a format must have an associated line specifying that format.

All Fortran programs run on WRDS must be compiled and made executable as follows:

To compile them:

For Compustat:

```
f77 -o prog prog.f
```

For CRSP and CRSPLink:

```
f90 -o prog prog.f -I$CRSP_INCLUDE $CRSP_LIB/crsplib.a -lm
```

To make them executable:

```
chmod +x prog
```



```

C#####
C Program : ina.f
C#####
C Usage : To read binary Compustat Annual Industrial file
C (Time period: 1978 through 1998)
C Input : ina.bin Annual Industrial
C Output : ina.out Selected Securities
C Method : The program reads the data for each security.
C If the security meets the selection criteria,
C sample data is written to an ascii disk file.
C Hardware : HP J210
C OS : HP-UX 10.20
C Software : FORTRAN/9000
C Author : Steve Crisp
C Modified : 9/99
C#####
C YEARA - The Data Year represents a 12-month fiscal period
C for which data is collected. The data year
C does not necessarily correspond to the calendar
C year. Thus, data for a company whose fiscal year
C ends May 31, 1989 appears in DATA YEAR 1988.
C FYR - This code designates the ending month for each
C company's accounting year. Fiscal years ending
C January 1 through May 31 are treated as ending
C the prior year. Thus, data for a fiscal year
C beginning on June 1, 1988 and ending on May 31,
C 1989 is reported as data year 1988. A fiscal
C year beginning on July 1, 1988 and ending on
C June 30, 1989, is reported as data year 1989.
C#####
C VARIABLE DECLARATIONS
C#####
C CHARACTER*28 CONAME, INAME
C CHARACTER CNUM*6, SBI*8, EIN*10, CIC*3
C CHARACTER CSPIN*1, CSSPIN*2, CSSPIT*1, SPDRCF*2, SPDRCF*2
C CHARACTER SUBDET*2, SPCBRC*2, NAICS*6
C CHARACTER*2 APTINT(70,20)
C REAL DATA(350,20)
C INTEGER DNUM, REC, FILE, ZLIST, FYR(20), YEARA(20), XREL
C INTEGER STK,DUP,UCODE(20)
C INTEGER STATE,COUNTY,FINC,SOURCE(20)
C INTEGER NSECCIN,NSECCSEL,IUNIT,JUNIT
C DATA IUNIT,JUNIT /0,0/
C DATA IUNIT,JUNIT /21,22/
C#####

```

```

C OPEN FILES
C
C Note: When compiling with f99 FORTRAN, use the ACTION = 'READ',
C command instead of the READONLY commands in the section below.
C#####
C OPEN (IUNIT, ACCESS='SEQUENTIAL', STATUS='OLD', READONLY,
C OPEN (IUNIT, ACCESS='SEQUENTIAL', STATUS='OLD', ACTION='READ',
C FORM='UNFORMATTED',
C FILE='/'wds/compustat/segdata/ina.bin')
C OPEN (JUNIT, ACCESS='SEQUENTIAL', STATUS='NEW',
C FILE='ina.out')
C#####
C LOOP TO READ AND WRITE COMPANY DATA
C#####
C DO WHILE (.TRUE.)
C READ (IUNIT,END=900) DNUM,CNUM,CIC,REC,FILE,ZLIST,INAME,CONAME,
C SMEL,FYR,YEARA,XREL,STK,DUP,UCODE,STATE,COUNTY,FINC,
C SOURCE,CSPIN,CSSPIN,CSSPIT,SPDRCF,SUBDET,SPBRC,
C EIN,NAICS,APTINT,DATA
C IF (DNUM.EQ.0) GOTO 900
C NSECCIN = NSECCIN + 1
C#####
C Replace the following IF statement with your own criteria
C and add your own processing code.
C#####
C IF (.TRUE.) THEN
C NSECCSEL = NSECCSEL + 1
C#####
C WRITE AN ASCII OUTPUT RECORD TO DISK
C DO 800, K = 1, 20
C WRITE (JUNIT,110) DNUM,CNUM,CONAME,YEARA(K),DATA(6,K),
C DATA(12,K)
C 800 CONTINUE
C#####
C REMOVE THE FOLLOWING LINE IN ORDER TO PROCESS THE ENTIRE FILE
C ONLY AFTER YOU ARE SURE YOUR SELECTION CRITERIA ARE CORRECT
C IF (NSECCSEL.GE.10) GOTO 900
C END IF
C END DO
C#####
C END OF FILE PROCESSING
C#####
C 900 CLOSE (IUNIT)
C CLOSE (JUNIT)
C WRITE (6,910) NSECCIN,NSECCSEL
C#####
C FORMAT STATEMENTS
C 110 FORMAT (14,X,A6,X,A28,X,14.2(X,F10.3))
C 910 FORMAT (X,NUMBER OF SECURITIES PROCESSED WAS: ',IS/
C * X,NUMBER OF SECURITIES SELECTED WAS: ',IS)
C STOP
C END

```

```

C#####
C Program : dsf.f
C#####
C Usage : TO READ CRSP BINARY DAILY STOCK FILE
C (TIME PERIOD: JULY 1962 THROUGH DECEMBER 1998)
C
C Input : CRSP Daily Calendar File & Daily Stock File
C dsf.bin & ds1.bin
C
C Output : dsf.out
C
C Method : THE PROGRAM READS THE DAILY STOCK CALENDAR FILE FOR
C CALENDAR DATES, THEN THE DATA FOR EACH SECURITY
C IS READ. IT READS ALL DATA FOR A SECURITY INTO
C HEADER, INFO, DDATA AND ADATA COMMON BLOCKS. IF
C THE SECURITY MEETS THE SELECTION CRITERIA, SAMPLE
C DATA IS WRITTEN TO AN ASCII DISK FILE.
C
C Hardware : HP J210
C OS : HP-UX 10.20
C Software : FORTTRAN/9000
C
C Author : CRSP
C Modified : 6/99
C#####
C NOTE: This program reads the data sequentially. This may take time
C especially if you have to read the whole file before you
C come to your data.
C
C If you have a list of PERMNOs in a separate file PLEASE
C ensure that they are sorted. See instructions below for
C correct method of incorporating this feature.
C-----
C This program contains some processing options which have
C been included as comments. If you wish to make use of these
C options, simply remove the "C" in the first column of each
C relevant FORTRAN statement and follow any accompanying
C instructions.
C
C There are two types of options available within this program,
C date boundaries and processing style.
C
C The date boundary options are:
C
C (1) Use the date boundaries supplied by the file;
C (2) Use a single set of date boundaries of your own choosing;
C (3) Use a different set of date boundaries for each security.
C
C To control the file processing with a set of pre-selected
C PERMNOs read from an input file called PERMNO.DAT.

```

```

C-----
C USING THE DATA:
C
C THE CRSP STOCK FILE USER'S GUIDE HAS COMPLETE VARIABLE
C DESCRIPTIONS, INFORMATION ON ACCESSING THE DATA AND HAS
C DESCRIPTIONS OF AVAILABLE UTILITY ROUTINES AND OTHER SUGGESTIONS.
C
C THE INFO COMMON BLOCK CONSISTS OF FIVE RECTANGULAR ARRAYS: NAMES,
C DIST, SHARES, DELIST, AND NASDIN. THE FIRST SUBSCRIPT INDICATES
C THE COMPONENT OF AN EVENT, AND THE SECOND INDICATES THE NUMBER
C OF THE EVENT. FOR EXAMPLE, DISTS(6,2) SPECIFIES THE 6TH WORD OF
C THE SECOND DISTRIBUTION STRUCTURE, OR THE EX-DISTRIBUTION DATE.
C MEMORNIC SUBSCRIPT PARAMETERS ARE AVAILABLE FOR ACCESSING
C COMPONENTS OF INFO ARRAYS, SO DISTS(EXDT,2) CAN BE USED INSTEAD
C OF DISTS(6,2). A CORRESPONDING HEADER VARIABLE NOMXXX INDICATES
C THE MAXIMUM NUMBER OF EVENTS OF THAT TYPE AVAILABLE FOR THAT
C SECURITY.
C
C ALL INFO ARRAYS ARE INTEGER ARRAYS. REAL AND CHARACTER ARRAYS
C ARE EQUIVALENCED TO SOME OF THE INTEGER ARRAYS WHEN ALL COMPONENTS
C OF THE EVENT ARE NOT INTEGERS. THE EQUIVALENCED ARRAY MUST BE
C USED TO ACCESS THE REAL OR CHARACTER FIELDS. FOR EXAMPLE, THE
C EXDATE OF THE THIRD DISTRIBUTION IS DISTS(EXDT,3) WHILE THE
C DIVIDEND AMOUNT IS RDISTS(DIVAMT,3).
C-----
C THE ALLINCLS INCLUDE FILE INSERTS INCLUDE FILES CONTAINING ALL
C PARAMETERS AND DECLARATIONS OF CALENDAR AND DATA VARIABLES USED
C BY THE CRSP STOCK SAMPLE PROGRAMS.
C-----
C INCLUDE '/wrds/crsp/1lb/allincls'
C-----
C DECLARE VARIABLES
C
C IUNIT UNIT NUMBER OF THE CALENDAR/INDEX FILE
C JUNIT UNIT NUMBER OF THE DATA FILE
C KUNIT UNIT NUMBER OF THE OUTPUT FILE
C LUNIT UNIT NUMBER OF THE PERMNO LIST FILE
C NSECI NUMBER OF SECURITIES READ IN SO FAR
C NSECS NUMBER OF SECURITIES SELECTED SO FAR
C-----
C INTEGER USER_PERMNO
C INTEGER IUNIT, JUNIT, KUNIT, LUNIT, NSECI, NSECS
C INTEGER EDATE, EDATE
C INTEGER BEGPOS, ENDPDS, BPOS, EPOS
C-----
C DATA USER_PERMNO /000000/ !DEFAULT FOR SEQUENTIAL PROCESSING
C DATA IUNIT,JUNIT,KUNIT,LUNIT /10,11,12,13/, NSECI,NSECS /0,0/
C-----
C OPEN CALENDAR AND DATA FILES
C
C Note: When compiling with f99 FORTRAN, use the ACTION = 'READ'
C command instead of the READONLY commands in the section below.
C-----
C OPEN (IUNIT,
C . FILE = '/wrds/crsp/seqdata/ds1.bin',
C . STATUS = 'OLD', READONLY,
C . STATUS = 'OLD', ACTION = 'READ',
C . ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')

```

```

OPEN (JUNIT,
      FILE = '/wrds/crsp/segdata/dsf.bin',
      STATUS = 'OLD', READONLY,
      ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')
      .
      .
      .
OPEN (KUNIT,
      FILE = 'dsf.out', STATUS = 'NEW',
      ACCESS = 'SEQUENTIAL')
      .
      .
      .
C   Open the PERMNO file which contains the PERMNOs of the securities
C   If you wish to extract and optionally, their corresponding dates.
C   If you select this option, you must also remove the 'C' from
C   one of the READ (JUNIT,101) statements below and from the section
C   labeled PERMNO.DAT CODE.
C
C   OPEN (JUNIT, FILE= 'permno.dat', STATUS= 'OLD')
C
C-----
C   READ CALENDAR RECORDS AND RETURN THE NUMBER OF TRADING DATES
C   IN NDAYS.
C-----
CALL BICAL(JUNIT,NDAYS)
CLOSE (JUNIT, STATUS = 'KEEP')
C-----
C   Set up default date range - whole period
C-----
BEGPOS = 1
ENDPOS = NDAYS
C-----
C O P T I O N A L   C O D E :   R e l a t i n g   t o   d a t e   b o u n d a r i e s
C-----
C   TO SET UP DATE BOUNDARIES
C   Replace the dates after the '=' with your own dates in
C   YYYYMMDD format. Remove the 'C' from the first column of
C   each of the next four statements.
C-----
C   IDATE = 19700101           !Begin Date
C   BEGPOS = INDXPT (IDATE)
C   IDATE = 19891231
C   ENDPOS = INDXPT (IDATE)
C-----
C-----
C   READ ALL THE DATA FOR THE NEXT SECURITY.
C-----
90 DO WHILE (.TRUE.)
C
C   If you opened the PERMNO.DAT file above, you must select one
C   of the READ statements below. If your PERMNO file contains
C   dates, then select the second READ statement and by removing the
C   'C's in column 1. Choose your dates carefully so that you
C   are sure data exists for the period.
C
C 100 READ (JUNIT,101,END=900) USER_PERMNO

```

```

C 100 READ (JUNIT,101,END=900) USER_PERMNO, BDATE, EDATE
C-----
C   PERMNO.DAT - Check for blank lines in the PERMNO.DAT file
C-----
C   IF (USER_PERMNO .EQ. 000000) THEN
C     GOTO 100
C   ENDIF
C-----
C   Remove the C's from column 1 of the following two lines if the
C   PERMNO.DAT file contains dates.
C-----
C   BEGPOS = INDXPT(BDATE)
C   ENDPOS = INDXPT(EDATE)
C-----
102 CALL BIGHT(JUNIT,USER_PERMNO,*900,*850)
C-----
C   PERMNO.DAT CODE - Remove C's if using an input file of PERMNOs.
C-----
C 105 IF (PERMNO .LT. USER_PERMNO) THEN
C     GOTO 102
C   ELSE IF (PERMNO .GT. USER_PERMNO) THEN
C     GOTO 100
C   ENDIF
C   IF (USER_PERMNO .EQ. 000000) THEN
C     GOTO 100
C   ENDIF
C-----
NSBCT = NSBCT + 1
C-----
C   Remove the C's from column 1 of the following lines if the
C   PERMNO.DAT file contains dates.
C-----
C   IF ((BEGPOS .LT. ENDDAT) .AND. (ENDPOS .GT. BEGDAT)) THEN
C     BPOS = BEGPOS
C     EPOS = ENDPOS
C     IF (BEGPOS .LT. BEGDAT) BPOS = BEGDAT
C     IF (ENDPOS .GT. ENDDAT) EPOS = ENDDAT
C     N OBS = EPOS - BPOS + 1
C     ENDIF
C-----
NSBCT = NSBCT + 1
C *****
C
C   INSERT PROCESSING CODE HERE
C-----
C *****
C-----
C   Write ASCII sample data
C-----
C   Modify the following 2 statements as needed.
C-----
C   This example writes all the price data that
C   exists for the security.

```


7. Essential FTP commands

For more information, check: <http://www.kellogg.nwu.edu/researchcomputing/ftp.htm>

FTP command	What it does
? or help	Get a listing of available FTP commands
help <i>command</i>	Get help on a specific command. Example: help ascii
open <i>host</i>	Connect to specified host. Example: open wrds.wharton.upenn.edu
close	Close an established connection
quit	Exit FTP client (will close any established connection)
binary or bin	Set transfer type to 'binary'. The default transfer type is ASCII
ascii	Set transfer type to ASCII
put	Transfer a file from the client (local) to the host (remote)
get	Transfer a file from the host (remote) to the client (local)
dir or ls	Get a directory listing of the current directory in the host
cd <i>directory_name</i>	Change directory in the host to specified directory
lcd <i>directory_name</i>	Change directory in the client to specified directory
prompt	Toggles interactive prompting in multiple commands. By default, commands such as "mget" and "mput" prompt the user to confirm the transfer of each file. To turn this confirmation off, issue the "prompt" command.
mput <i>file_names</i>	Transfer specified files from client (local) to host (remote). Wild cards may be used. Examples: mput raw.do test.do mput *.sas mput s3* mput *
mget <i>file_names</i>	Transfer specified files from host (remote) to client (local). Wild cards may be used. Examples: mget raw.do test.do mget *.sas mget s3* mget *
! OR ! <i>command</i>	Escape to shell. Issued by itself, "!" will take you to the local operating system's prompt, allowing the execution of commands (for example, creating directories or renaming files). To return to the FTP session, type "exit". Followed by a command (from the local operating system will immediately execute the command on the local machine. For example: "!dir" would list the contents of the current directory in the client (local machine).

Kellogg Research Computing
8. WRDS Data Files

Wharton Code	Product Name	Description	Seqdata Files	Sasdata Files
COMPSTAT				
INA	PST Annual Current	Industrial Annual (1979-98)	ina.bin ina.names	Comp.ina Comp.inanames
INAB1	PST Annual Backdata	Industrial Annual (1960-79)	inab1.bin inab1.names	Comp.inab1 Comp.inab1nam
INAB2	PST Annual Wayback	Industrial Annual (1950-69)	inab2.bin inab2.names	Comp.inab2 Comp.inab2nam
FCA	FCOTC Annual Current	Full Coverage Annual (1979-98)	fca.bin fca.names	Comp.fca Comp.fcanames
FCAB1	FCOTC Annual Backdata	Full Coverage Annual (1960-79)	fcab1.bin fcab1.names	Comp.fcab1 Comp.fcab1nam
FCAB2	FCOTC Annual Wayback	Full Coverage Annual (1950-69)	fcab2.bin fcab2.names	Comp.fcab2 Comp.fcab2nam
RES	Merged Annual Research Current	Research Annual (1979-98)	res.bin res.names	Comp.res Comp.resnames
RESB1	Merged Annual Research Backdata	Research Annual (1960-79)	resb1.bin resb1.names	Comp.resb1 Comp.resb1nam
RESB2	Merged Annual Research Wayback	Research Annual (1950-69)	resb2.bin resb2.names	Comp.resb2 Comp.resb2nam
INQ	PST Qtrly Current	Industrial Quarterly (1989-99)	inq.bin inq.names	Comp.inq Comp.inqnames
INQB1	PST Qtrly Backdata	Industrial Quarterly (1978-89)	inqb1.bin inqb1.names	Comp.inqb1 Comp.inqb1nam
INQB2	PST Qtrly Wayback	Industrial Quarterly (1967-78)	inqb2.bin inqb2.names	Comp.inqb2 Comp.inqb2nam
INQB3	PST Qtrly Way Wayback	Industrial Quaretrly (1962-73)	inqb3.bin inqb3.names	Comp.inqb3 Comp.inqb3nam
FCQ	FCOTC Qtrly Current	Full Coverage Quarterly (1989-99)	fcq.bin fcq.names	Comp.fcq Comp.fcqnames
FCQB1	FCOTC Qtrly Backdata	Full Coverage Quarterly (1978-89)	fcqb1.bin fcqb1.names	Comp.fcqb1 Comp.fcqb1nam
FCQB2	FCOTC Qtrly Wayback	Full Coverage Quarterly (1967-78)	fcqb2.bin fcqb2.names	Comp.fcqb2 Comp.fcqb2nam
FCQB3	FCOTC Qtrly Way Wayback	Full Coverage Quarterly (1962-73)	fcqb3.bin fcqb3.names	Comp.fcqb3 Comp.fcqb3nam
REQ	Merged Qtrly Research Current	Research Quarterly (1989-99)	req.bin req.names	Comp.req Comp.reqnames
REQB1	Merged Qtrly Research Backdata	Research Quarterly (1978-89)	reqb1.bin reqb1.names	Comp.reqb1 Comp.reqb1nam
REQB2	Merged Qtrly Research Wayback	Research Quarterly (1967-78)	reqb2.bin reqb2.names	Comp.reqb2 Comp.reqb2nam
REQB3	Merged Qtrly Research Way Wayback	Research Quarterly (1962-73)	reqb3.bin reqb3.names	Comp.reqb3 Comp.reqb3nam
CIA	CDN \$ Annual Current	Canadian Industrial Annual (1979-98)	cia.bin cia.names	Comp.cia Comp.cianames
CIQ	CDN \$ Qtrly Current	Canadian Industrial Qtrly (1989-99)	ciq.bin ciq.names	Comp.ciq Comp.ciqnames
SIC	BIF Standard Indl Classification	SIC file	N/a	Comp.sic
BIF	BIF Industry Segment Current	Industry Segments (1992-98)	bif.bin bif.names	Comp.bif Comp.bifnames
BIFB1	BIF Industry Segment Backdata	Industry Segments (1986-92)	bifb1.bin bifb1.names	Comp.bifb1 Comp.bifb1n
BIFB2	BIF Industry Segment Wayback	Industry Segments (1980-86)	bifb2.bin bifb2.names	Comp.bifb2 Comp.bifb2nam

Kellogg Research Computing

Wharton Code	Product Name	Description	Seqdata Files	Sasdata Files
BII	BIF Industry Segment Research Current	Industry Segments Research (1992-98)	bii.bin bii.names	Comp.bii Comp.biinames
BIIB1	BIF Industry Segment Research Backdata	Industry Segments Research (1986-92)	biib1.bin biib1.names	Comp.biib1 Comp.biib1nam
BIIB2	BIF Industry Segment Research Wayback	Industry Segments Research (1980-86)	biib2.bin biib2.names	Comp.biib2 Comp.biib2nam
GEO	BIF Geographic Segment Current	Geographic Segments (1992-98)	geo.bin geo.names	Comp.geo Comp.geonames
GEOB1	BIF Geographic Segment Backdata	Geographic Segments (1986-92)	geob1.bin geob1.names	Comp.geob1 Comp.geob1nam
GEOB2	BIF Geographic Segment Wayback	Geographic Segments (1980-86)	geob2.bin geob2.names	Comp.geob2 Comp.geob2nam
GEI	BIF Geographic Segment Research Current	Geographic Segments Research (1992-98)	gei.bin gei.names	Comp.gei Comp.geinames
GEIB1	BIF Geographic Segment Research Backdata	Geographic Segments Research (1986-92)	geib1.bin geib1.names	Comp.geib1 Comp.geib1nam
GEIB2	BIF Geographic Segment Research Wayback	Geographic Segments Research (1980-86)	geib2.bin geib2.names	Comp.geib2 Comp.geib2nam
PDE	US PDE Adjusted	Price-Dividens-Earnings (1961-1997)	pde.bin pde.names	Comp.pde Comp.pdenames
BNA	Bank Annual Current	Bank Annual (1978-97)	bna.bin bna.names	Comp.bna Comp.bnanames
BNQ	Bank Qtrly Current	Bank Quarterly (1988-98)	bnq.bin bnq.names	Comp.bnq Comp.bnqnames
IFA	S&P Index Fundamentals – Annual	Index Fundamentals Annual	N/a	N/a
GV	Global Advantage	Industrial/Commerical Financial Services Issues Currency	gvic.bin gvic.names gvfs.bin gvfs.names gvi.bin gvi.names gvc.bin	Comp.gvic Comp.gvicname Comp.gvfs Comp.gvfsnam Comp.gvi Comp.gvinames Comp.gvc
EXECCOMP	ExecuComp	Executive Compensation	N/a	Comp.execcomp
COMPUSTAT DRI				
BASIC	Basic Economics	Basic Economics (Sample) - Annual Basic Economics (Sample) – Quarterly Basic Economics (Sample) – Monthly	N/a	Comp.basica Comp.basicq Comp.basicm
USCENW	Global Economics	Global Economics USCEN Weekly	N/a	Comp.uscenw
CRSP				
MSF	1998 MAX NYSE/AMEX/NASDAQ Monthly Stock	Calendar/Indices data Stock data Cross section information Events data Company identifiers CRSPAccess97 data files	msi.bin msf.bin msf.names ma199812/	Crsp.msi Crsp.msf Crsp.msfby Crsp.mse Crsp.msfnames
DSF	1998 DAX NYSE/AMEX/NASDAQ Daily Stock	Calendar/Indices data Stock data Cross section information Events data Events data merged by permno/date Company identifiers CRSPAccess97 data files NYSE/AMEX year-end cap/port/beta/sdev Nasdaq year-end cap/port/beta/sdev	dsi.bin dsf.bin dsf.names da199812/	Crsp.dsi Crsp.dsf Crsp.dsfby Crsp.dse Crsp.dseall Crsp.dsfnames Crsp.asf1 Crsp.asf2

Kellogg Research Computing

Wharton Code	Product Name	Description	Seqdata Files	Sasdata Files
DEC	1998 IXX Indices/Portfolio Assignments	Daily NYSE Indices/Cap. Deciles Monthly NYSE Indices/Cap. Deciles Quarterly NYSE Indices/Cap. Deciles Annual NYSE Indices/Cap. Deciles Daily AMEX Indices/Cap. Deciles Monthly AMEX Indices/Cap. Deciles Quarterly AMEX Indices/Cap. Deciles Annual AMEX Indices/Cap. Deciles Daily NYSE/AMEX Indices/Cap. Deciles Monthly NYSE/AMEX Indices/Cap. Dec Quarterly NYSE/AMEX Indices/Cap. Dec Annual NYSE/AMEX Indices/Cap. Dec Daily NASDAQ Indices/Cap. Deciles Monthly NASDAQ Indices/Cap. Deciles Quarterly NASDAQ Indices/Cap. Deciles Annual NASDAQ Indices/Cap. Deciles Daily NYSE/AMEX/NASDAQ Indices Monthly NYSE/AMEX/NASDAQ Indices Quarterly NYSE/AMEX/NASDAQ Ind Annual NYSE/AMEX/NASDAQ Indices Daily NYSE/AMEX Ind/Beta Deciles Daily NYSE/AMEX Ind/StDev Deciles Daily Nasdaq Indices/ Beta Deciles Daily Nasdaq Indices/StDev Deciles Monthly Cap-Based NYSE Results Monthly Cap-Based NYSE/AMEX Res. Mthly Cap-Based NYSE/AMEX/NAS Res Quarterly Cap-Based NYSE Results Quarterly Cap-Based NYSE/AMEX Res. Qtrly Cap-Based NYSE/AMEX/NAS Res Daily CRSP Index on the S&P500 Monthly CRSP Index on the S&P500 Monthly Treasury and Inflation Index Quarterly Treasury and Inflation Index Annual Treasury and Inflation Index CRSPAccess97 data files Year-end NY/AM/NASD Cap Deciles Year-end NYSE/AMEX Cap Deciles Year-end Nasdaq Capitalization Deciles Year-end NYSE Capitalization Deciles Year-end AMEX Capitalization Deciles Year-end NYSE/AMEX Beta Deciles Year-end NYSE/AMEX St Dev Deciles Year-end Nasdaq Beta Deciles Year-end Nasdaq St Dev Deciles Year-end NY/AM/NASD Cap Deciles Year-end NYSE/AMEX Cap Deciles Year-end Nasdaq Capitalization Deciles Year-end NYSE Capitalization Deciles Year-end AMEX Capitalization Deciles Daily ret with NY/AM/NASD Cap Deciles Daily ret with NYSE/AMEX Cap Deciles Daily returns with Nasdaq Cap Deciles Daily returns with NYSE Cap Deciles Daily returns with AMEX Cap Deciles Daily ret with NYSE/AMEX Beta Deciles Daily ret w/ NYSE/AMEX St Dev Deciles Daily returns with Nasdaq Beta Deciles Daily returns with Nasdaq St Dev Deciles Monthly ret w/ NY/AM/NAS Cap Deciles Monthly ret w/ NYSE/AMEX Cap Deciles Monthly returns with Nasdaq Cap Deciles	dsia.bin msia.bin qsia.bin asia.bin dsib.bin msib.bin qsib.bin asib.bin dsic.bin msic.bin qsic.bin asic.bin dsio.bin msio.bin qsio.bin asio.bin dsix.bin msix.bin qsix.bin asix.bin dsbc.bin dssc.bin dsbo.bin dsso.bin mhista.bin mhistn.bin mhistq.bin rebala.dat rebaln.dat rebalq.dat dsp500.bin msp500.bin mcti.bin qcti.bin acti.bin ix199812/	Crsp.dsia Crsp.msia Crsp.qsia Crsp.asia Crsp.dsib Crsp.msib Crsp.qsib Crsp.asib Crsp.dsic Crsp.msic Crsp.qsic Crsp.asic Crsp.dsio Crsp.msio Crsp.qsio Crsp.asio Crsp.dsix Crsp.msix Crsp.qsix Crsp.asix Crsp.dsbc Crsp.dssc Crsp.dsbo Crsp.dsso Crsp.mhistq Crsp.mhistn Crsp.mhistq Crsp.rebala Crsp.rebaln Crsp.rebalq Crsp.dsp500 Crsp.msp500 Crsp.mcti Crsp.qcti Crsp.acti Crsp.dport1 Crsp.dport2 Crsp.dport3 Crsp.dport4 Crsp.dport5 Crsp.dport6 Crsp.dport7 Crsp.dport8 Crsp.dport9 Crsp.mport1 Crsp.mport2 Crsp.mport3 Crsp.mport4 Crsp.mport5 Crsp.erdport1 Crsp.erdport2 Crsp.erdport3 Crsp.erdport4 Crsp.erdport5 Crsp.erdport6 Crsp.erdport7 Crsp.erdport8 Crsp.erdport9 Crsp.ermport1 Crsp.ermport2 Crsp.ermport3

Kellogg Research Computing

Wharton Code	Product Name	Description	Seqdata Files	Sasdata Files
GBM	1998 BMR Monthly US Government Bonds	Fixed Term Indices Fama Portfolio Returns – 6 mon intervals Fama Portfolio Returns – 12 mon intervals Fama-Bliss Discount Bond Prices Fama-Bliss Discount Bond Yields Calendar file Master file header Master file data Cross-Sectional file Cross-Sectional file – CRSPID order Risk Free Rates 12 month forward rates based on ask 6 month forward rates based on ask 12 month holding returns based on ask 6 month holding returns based on ask 12 month prices based on ask 6 month prices based on ask 12 month yields based on ask 6 month yields based on ask 12 month forward rates based on avg 6 month forward rates based on avg 12 month holding returns based on avg 6 month holding returns based on avg 12 month prices based on avg 6 month prices based on avg 12 month yields based on avg 6 month yields based on avg 12 month forward rates based on bid 6 month forward rates based on bid 12 month holding returns based on bid 6 month holding returns based on bid 12 month prices based on bid 6 month prices based on bid 12 month yields based on bid 6 month yields based on bid Fama Maturity Portfolios Fixed Term Indices Fama-Bliss Prices and Yields Yield Price Forward Rate File Holding Period Return File Price File Risk Free Rate File	bxmthind.dat bondport6.dat bondport12.dat famablispri.dat famablisyld.dat mbi.bin mbm.bin mbx.bin riskfree.dat fwdask12.dat fwdask06.dat hldask12.dat hldask06.dat priask12.dat priask06.dat yldask12.dat yldask06.dat fwdave12.dat fwdave06.dat hldave12.dat hldave06.dat priave12.dat priave06.dat yldave12.dat yldave06.dat fwdbid12.dat fwdbid06.dat hldb12.dat hldb06.dat pribid12.dat pribid06.dat yldb12.dat yldb06.dat bondport.xls bxmthind.xls famablis.xls famayield.xls ffwdrates.xls fhldrates.xls fprice.xls riskfreerate.xls	Crsp.bxmthind Crsp.bndprt06 Crsp.bndprt12 Crsp.fbpri Crsp.fbyld Crsp.mbi Crsp.mbmhdr Crsp.mbmdat Crsp.mbx Crsp.mbxid Crsp.riskfree Crsp.fwdask12 Crsp.fwdask06 Crsp.hldask12 Crsp.hldask06 Crsp.priask12 Crsp.priask06 Crsp.yldask12 Crsp.yldask06 Crsp.fwdave12 Crsp.fwdave06 Crsp.hldave12 Crsp.hldave06 Crsp.priave12 Crsp.priave06 Crsp.yldave12 Crsp.yldave06 Crsp.fwdbid12 Crsp.fwdbid06 Crsp.hldb12 Crsp.hldb06 Crsp.pribid12 Crsp.pribid06 Crsp.yldb12 Crsp.yldb06
MFDB	1996 MFR US Mutual Fund Database	Mutual fund data Mutual fund Identifiers Distributions Fund Averages	N/a	Crsp.mfdb Crsp.mfdbnam Crsp.mfdbd Crsp.mfdbavg
CCM	CRSP/COMPUSTAT Merged Database	CRSP/COMPUSTAT Merged Database	uh199907/	Crsp.cheadest
FDIC				
RIS	Research Information System	Complex Derived Items Financial Time Series Mergers and Aquisitions Ratios Structure	N/a	Fdic.cdiyymm Fdic.ftsyyymm Fdic.mergyyymm Fdic.ratyymm Fdic.struyymm
DOW JONES AVERAGES				
DJIA	Dow Jones Averages	Dow Jones Averages	N/a	Dj.djia
PHLX				
COPH	Currency Options	Currency Options	N/a	Phlx.coph

Kellogg Research Computing

Wharton Code	Product Name	Description	Seqdata Files	Sasdata Files
IV	Implied Volatility	Implied Volatility	N/a	Phlx.iv
TAQ				
CT/CQ	NYSE TAQ (Monthly CDs)	Master File Dividend File Consolidated Trades Consolidated Quotes	N/a	Taq.mastyymm Taq.divyymm Taq.ctyymm Taq.cqyymm
IBES				
DET	Daily Detail History	Detail Identifier Adjustments Excluded Estimates Sector/Industry/Group Codes Stopped Estimate Exchange Rate Report Currency Actuals	N/a	Ibes.det Ibes.id Ibes.adj Ibes.exc Ibes.sig Ibes.stop Ibes.canx Ibes.cur Ibes.act
HIOUT	Summary History	Summary Statistics Background Data Company Identification Adjustment Factors Sector/Industry/Group Codes Currency File EURO Exchange Rate	N/a	Ibes.hiout1 Ibes.hiout2 Ibes.hiout3 Ibes.hiout4 Ibes.hiout5 Ibes.hiout6 Ibes.hiout7