Center for Research in Security Prices
Graduate School of Business
University of Chicago

STOCK

CRSPAccess97

1997 Stock File Users Guide

CENTER FOR RESEARCH IN SECURITY PRICES

Graduate School of Business

The University of Chicago

CRSPAccess97 Stock File Users Guide

Data ending December 31, 1997

Covering securities traded on The New York Stock Exchange (NYSE),
The American Stock Exchange (AMEX), and
The Nasdaq Stock Market[SM] (NASDAQ).

Daily and Monthly Files

Center for Research in Security Prices
Graduate School of Business
University of Chicago
725 South Wells Street, Suite 800
Chicago, IL 60607

Telephone: 773.702.7467
FAX: 773.702.3036

E-Mail mail@crsp.uchicago.edu

Internet Address: http://www.crsp.com

To install, follow the instructions in the appendices of the documentation.

IMPORTANT – READ CAREFULLY BEFORE OPENING OR USING DATA PACKETS

COPYRIGHT NOTICE

PROPRIETARY RIGHTS

CRSP DATA LICENSE

DISCLAIMER

CRSP will endeavor to obtain information appearing on its Data Files from sources it considers reliable, but disclaims any and all liability for the truth, accuracy or completeness of the information conveyed. THE UNIVERSITY AND CRSP MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH RESPECT TO THE MERCHANTABILITY, FITNESS, CONDITION, USE OR APPROPRIATENESS FOR SUBSCRIBER'S PURPOSES OF THE DATA FILES AND DATA FURNISHED TO THE SUBSCRIBER UNDER THIS SUBSCRIPTION, OR ANY OTHER MATTER AND ALL SUCH DATA FILES WILL BE SUPPLIED ON AN "AS IS" BASIS. CRSP will endeavor to meet the projected dates for updates, but makes no guarantee thereof, and shall not have any liability for delays, breakdowns or interruption of the subscription. In no event shall the University or CRSP be liable for any consequential damages (even if they have been advised of the possibility of such damages), for damages arising out of third party suppliers terminating agreements to supply information to CRSP, or for other causes beyond its reasonable control.

In the event that the Subscriber discovers an error in the Data Files, Subscriber's sole remedy shall be to notify CRSP and CRSP will use its best efforts to correct same and deliver corrections with the next update.

Version CAU-1997-1.0

NOTICE CONCERNING USE OF THE CRSP DATA

The CRSP data files are proprietary and should be used only for research purposes by the faculty, students, or employees of the subscribing institution**.** The Subscription Agreement, signed by each subscribing institution states:

1. Subscriber acknowledges that the data files to which it is subscribing contain factual material selected, arranged and processed by CRSP and others through research applications and methods involving much time, study, and expense.

2. The Subscriber agrees that it will not transfer, sell, publish, or release in any way any of the data files or the data contained therein to any individual or third party who is not an employee or student of the Subscriber, and that the data provided to the Subscriber by CRSP is solely for the Subscriber's use.

3. Should the Subscriber or any of the employees or students of the Subscriber (collectively referred to herein as "users") wish to utilize CRSP data for any non-academic or profit-making endeavor, said user shall first obtain CRSP's express written consent and agree to pay an applicable royalty fee to CRSP.

4. The Subscriber may not copy the data or documentation in any form onto any device or medium without the express written consent of CRSP, except solely to create back-up copies of the CRSP data files for its internal use, subject to the terms of this Agreement.

5. Subscriber agrees and warrants that it will take all necessary and appropriate steps to protect CRSP's proprietary rights and copyright in the data supplied (including, but not limited to, any and all specific steps which may be expressly required by CRSP), and that the Subscriber will protect the data in no less than the manner in which it would protect its own confidential or proprietary information.

6. The Subscriber will inform all users and potential users of CRSP data of CRSP's proprietary rights in its files and data by giving each user a copy of this paragraph and any other specific requirements CRSP may mandate under this paragraph and by requiring each such user to comply with this paragraph and all such additional requirements.

7. This subscription applies to only one campus or location of a multi-campus or multi-location system; any additional campus or location desiring access to CRSP data files must apply for a separate subscription.

8. The Subscriber agrees that its obligation under the Subscription Agreement shall survive the termination of this Agreement for any reason.

IN RESPONSE TO REQUESTS FROM THE ACADEMIC COMMUNITY, WE HAVE PERMITTED SUBSCRIBERS TO USE THE CRSP DATA FILES ON A FEE BASIS FOR CERTAIN CONSULTING, REGULATORY, AND JUDICIAL APPLICATIONS. FEES ARE BASED ON THE FILES USED AND THE LENGTH OF USAGE. PLEASE CONTACT THE CRSP PRODUCTS AND SERVICES OFFICE (773.702.7467) FOR INFORMATION REGARDING THESE FEES.

The CRSP CDs are intended to be used as a backup only. They should be copied onto your system as soon as you receive them.

See reverse side for Data Replacement Policy ➔

DATA REPLACEMENT POLICY

If any item is missing or damaged, report it immediately. All CRSP files are supplied on certified mediums to assure readability. If you encounter problems reading your CD due to shipping damage or a hard I/O error, CRSP WILL REPLACE YOUR CD FREE OF CHARGE, PROVIDED YOU HAVE FOLLOWED THE FOUR STEPS OUTLINED IN THE YELLOW DELIVERY ACKNOWLEDGMENT CARD WITHIN 30 DAYS OF RECEIPT. To obtain a replacement CD, contact the Products and Services Office at **773.702.7467**. You will be given an RGA Number and asked to return the defective CD along with a letter and/or documentation supporting the problem.

Should you desire a replacement CD for reasons other than shipping damage or hard I/O errors, the fee will be $100.00 per CD. You are entitled to one hardcopy of documentation for each product type. Additional hardcopies will be billed at $20.00 per product per database format. Documentation is available, and updated, on-line through the file guide link at http://www.crsp.com. As a subscriber, you are entitled to make copies of the documentation and to distribute it for internal use only.

Note: Replacement CDs are subject to availability. 1997 Replacement CDs will not be available after January 31, 1999.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 How to Use This Guide

Please Read This Documentation Thoroughly before Attempting to Access the Data

Who Should Use This Guide

This guide is for all users of the CRSP Stock Files in CRSPAccess97 format. It includes file and data item organization and descriptions, data utility programs to access the data and installation instructions. Programmers should consult the **CRSPAccess97 Stock File Programmers Guide**, which contains details of CRSP libraries and sample programs that can be used to create FORTRAN 77 or C programs to access the data.

Document Organization

The Introduction contains a brief description of the stock database, an overview on the development of the CRSP stock database files, and highlights recent changes to these files.

The Data Description contains definitions and descriptions of all data items (variables) on the files. These are presented first in FORTRAN programming structures, then in structures organized for the C programming language. Indices variables, for use with the accompanying indices files, are built into the FORTRAN programming structures, and appended to the end of the C structures.

The Utilities section contains descriptions and instructions for use of CRSPAccess97 Data Utility Tools provided with the data. These tools include the Report Writer Utilities, Browse Programs and Portfolio Building Programs.

The Appendices contain supplemental information as follows:
A. Details of expanded coding systems and structures,
B. Data limitations and lists of issues affected by continuing research on the file,
C. Installation instructions for supported systems and programming basics, and
D. Data availability by product type.

The Index contains references to the variable mnemonics and names and to the interactive programs.

A companion guide, the **CRSPAccess97 Stock File Programmers Guide**, contains information and specifics on FORTRAN sample programs, data structures and usage, and access and utility functions, C sample programs, data structures and usage, and access and utility functions, and converted binary file specifications.

A companion product, the CRSPAccess97 Indices File/Portfolio Assignments File, supports supplemental stock and indices data integrated with the Stock Files. If using this product, see the **CRSPAccess97 Indices File Guide** for descriptions of the additional data.

Notational Conventions

- All names that occur within CRSP's FORTRAN or C sample programs and include files are printed using a `constant-width, courier,` font. These names include variable names, parameter names, subroutine names, subprogram names, function names, library names, and keywords. For example, CUSIP refers to the CUSIP Agency's identifier, while `CUSIP` refers to the variable that the programs use to store this identifier. CRSP's variable mnemonics, used as names and in the descriptions, are displayed capitalized using a `CONSTANT-WIDTH` font. C is typically displayed in `lower case,` excepting constants, which are displayed in `UPPER CASE`, while FORTRAN is displayed in `UPPER CASE`.

- All names that refer to the data utilities or include file titles are printed using an *italic helvetica* font (*`Italic Lucinda Console`* in the html version).

- Names of FORTRAN common blocks are delimited by slashes(/ /).

- Names with a similar format are sometimes referenced collectively, using three X's where the names differ. For example, the FORTRAN variables Beginning of Volume Data (`BEGVOL`), Beginning of Return Data (`BEGRET`), Beginning of Price Data (`BEGPRC`), etc. are sometimes referred to collectively as `BEGXXX`.

- In the variable definitions section, the variable `I` is sometimes used in referencing a variable in a FORTRAN array. In this case, `I` refers to a possible range of valid data in this array for this company, where the valid range is determined by the number of header variables. For example, the names date is referred to as `NAMES(NAMEDT,I)`. Here `I` is an integer between 1 and the header variable `NUMNAM`, which represents the number of name structures that exist for any specified issue in the CRSP Stock Database.

- All CRSP-defined data types have names in all capitals beginning with `CRSP_`.

- The text of this document is in Times New Roman. *Italics* and **bold** styles are used to emphasize headings, names, definitions and related functions.

- The term `CRSPDB` refers to a CRSPAccess97 database.

1.2 Preface

The Center for Research in Security Prices (CRSP) compiles the CRSP Stock Database Files which provide comprehensive security price data on the New York Stock Exchange (NYSE), the American Stock Exchange (AMEX) and The Nasdaq Stock Market℠ (Nasdaq). These files contain price, return and distribution data and are currently available in the following combinations: NYSE/AMEX and NYSE/AMEX/Nasdaq. The CRSP Stock Files are further differentiated by the frequency, monthly and daily, at which prices and returns are reported.

The Stock Files Provide Seven Types of Information on Individual Securities:

1. Identifying information including; complete name histories, all historical CUSIPs, share classes, tickers, Standard Industrial Classification (SIC) codes and other identifiers,
2. Price histories and trading volumes for the period a security is traded on the NYSE, the AMEX, or the Nasdaq Stock Market,
3. Security delisting information,
4. Complete distribution histories including all dividends, stock splits, and special distributions,
5. Shares outstanding values and year-end market capitalization,
6. CRSP's Total return series and capital appreciation holding period return series, and
7. Portfolio capitalization assignment data for the CRSP Indices.

The Stock Files are Accompanied by Basic Market Indices:

1. Equal-Weighted Market Returns, calculated with and without dividends,
2. Value-Weighted Market Returns, calculated with and without dividends,
3. Standard & Poor's 500 and/or Nasdaq Composite (dependent upon Product).

The CRSP Market Return Indices Series are based exclusively on the securities appearing in the corresponding Stock File. A separate indices product, the CRSPAccess97 Indices File/Portfolio Assignments File, can be integrated with the CRSPAccess97 Stock Database to provide additional market and portfolio series, as well as additional security level portfolio statistic and assignment data. Indices are available for different exchange combinations and methodologies, and can be found listed in the CRSPAccess97 Indices File Guide.

1.3 Development of the CRSP Stock Files

CRSP Stock File Data Dates By Exchange

	Monthly Stock Files		Daily Stock Files	
Exchange	**Beginning Date**	**End Date**	**Beginning Date**	**End Date**
NYSE	12/31/25	12/31/97	07/02/62	12/31/97
AMEX	07/02/62	12/31/97	07/02/62	12/31/97
Nasdaq	12/29/72	12/31/97	12/14/72	12/31/97

The CRSP Data Files were developed by the Center for Research in Security Prices (CRSP), Graduate School of Business, University of Chicago. Lawrence Fisher, currently at Rutgers University, built the CRSP Stock Master File and originated the basic design and content of the Master File. For a more complete discussion of the original files, see Lawrence Fisher and James H. Lorie, *A Half Century of Returns on Stocks and Bonds*, Chicago: The University of Chicago, Graduate School of Business, 1977, Appendices A and B.

The original CRSP Stock File contained month-end prices and returns from the New York Stock Exchange (NYSE) dating from December, 1925. The monthly American Stock Exchange (AMEX) data beginning in July, 1962 was combined with the NYSE Monthly File to create the current NYSE/AMEX Monthly File. This file contains information on approximately 8,800 common stock securities to date. The NYSE/AMEX Daily File is comprised of the daily prices and returns dating from July 2, 1962, and contains historical data on roughly 8,250 securities.

The Nasdaq Stock File contains information for over 14,800 domestic common stocks and ADRs traded on the Nasdaq Stock Market since December 14, 1972. The Nasdaq Stock File was originally released in 1987. Nasdaq security data is now merged with NYSE and AMEX security data in the CRSP NYSE/AMEX/Nasdaq Daily and Monthly Stock Files.

Data Sources for the CRSP NYSE/AMEX Files

The data used to construct the original Master File was hand collected by CRSP. Standard & Poor's Price Tape and Punched Card Dividend Service provided the daily price and dividend data between July, 1962 and September 1, 1972. On September 1, 1972, these services were acquired by Interactive Data Corporation (IDC) of Waltham, Massachusetts. IDC continued to provide the data between September, 1972 and April, 1987. Interactive Data Services, Inc. (IDSI), a subsidiary of IDC, has supplied this data since April, 1987. IDSI has additionally provided back data to include high, low, and volume data between July, 1962 and March, 1987.

Data Sources for the CRSP Nasdaq Historical Data File

CRSP collected machine-readable data from three sources to build the Nasdaq File. Interactive Data Corporation (IDC) of Waltham, Massachusetts, provided data on the daily price quotes and information about capitalization and distributions to shareholders between December 12, 1972 and August 31, 1984. The National Association of Securities Dealers (NASD) provided the data from November 1, 1982 to the present, with the exception of February, 1986 IDSI provided daily price and volume data for February, 1986. IDC was used as a secondary source to NASD between November 1, 1982 and August 31, 1984.

Nasdaq Markets

The Nasdaq Stock Market[SM] is comprised of two subsets of securities, The Nasdaq National Market and The Nasdaq Small Cap Market. Currently, for a security to be designated a Nasdaq National Market Security, it must meet criteria setting minimum levels for: annual income, numbers of publicly traded shares, market capitalization, share price, and number of market makers. The requirements for maintaining The Nasdaq National Market status are less stringent than the Stock Market's listing requirements. All other securities belong to The Nasdaq Small Cap Market. A security may move between The Nasdaq National Market and The Nasdaq Small Cap Market over time as its status changes.

The Nasdaq National Market was initiated in April 1982 for larger and generally more actively traded Nasdaq securities. The Nasdaq National Market Securities must meet higher financial and non-financial criteria than other

Nasdaq stocks, and are subject to last-sale reporting. In June of 1992 the regular Nasdaq segment of The Nasdaq Stock Market[SM] was renamed The Nasdaq Small Cap Market and for the first time these became subject to real-time price and volume reporting.

The CRSPAccess97 Nasdaq security data includes the closing bid, closing ask, and the number of trades, formerly included in the CRSP Supplemental Nasdaq Data File. The latter data items have been reported for issues listed on The Nasdaq National Market since November 1, 1982. Issues listed on The Nasdaq Small Cap Market have had these data reported since June 15, 1992.

For a more detailed description of how to identify The Nasdaq National Market and The Nasdaq Small Cap Market securities, see The Nasdaq National Market indicator definition for `NMSIND` in the Nasdaq information history array `NASDIN` described in the data definitions section.

Data Accuracy of the CRSP Stock Files

CRSP Stock Files are designed for research and educational use and have proven to be highly accurate. Considerable resources are expended to improve and to check the quality of the data. The CRSP Stock Files contain over one hundred million prices, distributions and derived data. Errors are not common. Some of the errors found in checking the data are the results of inaccuracies in the initial data source. The inaccuracies are corrected as soon as possible. Other errors are CRSP coding errors; over time these coding errors are found and corrected. Historical corrections account for differences in the data from update to update. The Annual CRSP Stock Files contain updated data through the end of the previous calendar year.

Machine-readable data are checked for internal consistency. Secondary sources including the *CUSIP Directory*, *Moody's Dividend Record*, Commerce Clearing House's *Capital Changes Reporter*, *The Directory of Obsolete Securities*, *Moody's Manuals*, the New York Stock Exchange *Weekly Bulletin*, the American Stock Exchange *Weekly Bulletin*, *Bank and Quotation Record*, *The Wall Street Journal*, and *The Commercial and Financial Chronicle* are used to check suspect information. Information not available in machine-readable form is hand-coded and verified.

1.4 Changes to the 1997 CRSP Access Stock Files

CRSPAccess97

CRSPAccess97 provides a fundamental change from the access formerly provided with CRSP Stock and Indices Data. CRSPAccess97 provides the following features:

- Binary data, programming libraries, and utility programs are provided for target machines.
- Precompiled and executable utilities are provided to dump data and perform namelist searches without programming.
- FORTRAN77 sample programs previously provided by CRSP are supported. Additional data items and random access are now available.
- C sample programs and libraries are available. C access supports random access and read/write capability.
- There are now two CRSPAccess97 Stock File Guides; the **CRSPAccess97 Users File Guide** and the **CRSPAccess97 Programmers File Guide**. The **CRSPAccess97 Stock File Users Guide** contains the data definitions, multiple data utility programs and installations instructions. The **CRSPAccess97 Programmers Guid**e contains programming specifics to modify and write FORTRAN and C programs, dump the data into the old binary format for use with the user' old programs, and a copy of the CRSP sample programs.

Database Changes (Changes Between the Old Database Format and CRSPAccess97)

CRSPAccess97 provides the same basic structure and data items as previous CRSP files. There are additional data items, structure components, and convention changes. FORTRAN programmers continue to use the same data organization as in previous CRSP files, but there are minor changes in organization for C programmers and users of utility programs. The Data Description section contains diagrams of the FORTRAN and C data item structures.

Year 2000-Compliant Date Formats

All dates in calendars and event fields are stored with full four-digit years, in YYYYMMDD format. FORTRAN programs convert back to YYMMDD format by default for backward compatibility, but can be set to use the YYYYMMDD format.

Daily Returns Without Dividends

The `RETX` variable is included in daily files as well as monthly files.

Daily Supplemental Nasdaq Data Items

Items previously stored in the Supplemental Nasdaq File, `NMSBEG`, `NMSEND`, `NMSBID`, `NMSASK`, and `NMSTRD`, are built into the standard stock databases.

Monthly Closing Bid and Ask for Nasdaq Issues

The last closing bid and closing ask each month from the daily Nasdaq Supplemental fields are included in monthly databases.

Monthly Alternate Price and Date of Alternate Price

Monthly databases between July 1962 and the present have two additional time series based on derivations of prices from daily databases. The alternate price series matches the standard price series with additional information for three additional cases: the first and last daily trading date in a partial month and missing month-end prices. The date of the alternate price is the date of each price in the alternate price series. Alternate price is only available to C programs.

Additional Portfolio Support

New C portfolio structures allow multiple portfolio statistic and assignment types, each with its own rebalancing calendar. Each portfolio type is linked to a group of portfolio index results. Nine daily and eight monthly portfolio types are available with the CRSPAccess97 Indices File/Portfolio Assignments product.

Additional Indices Items

A new common C index structure is based on indices organized by individual series or portfolio group. Each index is assigned a permanent identifier called `INDNO` by CRSP. Each index contains header information, and groups include rebalancing breakpoint statistics. Stock files contain a subset of market index series. Additional series and portfolio group indices are available on the CRSPAccess97 Indices File/Portfolio Assignments product.

Excess Returns Changes

FORTRAN support for excess returns and portfolio variables is now available for users with access to the CRSPAccess97 Indices and Portfolio Assignments Product. Options are available to duplicate the information in the CRSP Excess Return product as well as supporting excess returns against new index types. See the CRSPAccess97 Stock File Programmers Guide for Details. Nine daily and eight monthly portfolio series are available with the CRSPAccess97 Indices File/Portfolio Assignments product. The *ts_print* application can generate selected excess returns using available securities and index returns series.

Changes to the 1997 CRSPAccess97 Data

- **Excess Returns** FORTRAN support for excess returns and portfolio variables , `BXRET()` and `SXRET()`, compatible with those in the old format stock file is now available for users who also subscribe to the CRSPAccess97 Indices and Portfolio Assignments Product. Options are available to duplicate the information in the CRSP Excess Return product as well as supporting excess returns against new index types. See the CRSPAccess97 Stock File Programmers Guide for Details. Nine daily and eight monthly portfolio series are available with the CRSPAccess97 Indices File/Portfolio Assignments product. The `ts_print` application can generate selected excess returns using available securities and index returns series.

- **Delisting Codes:** Research was done on approximately 250 delistings coded as liquidations in the CRSP Stock Files. Changes include recoding, addition of missing distributions and adjustment of delisting returns.

  Liquidations occurring after 1962 that have a delisting code of 400 have been changed:

  > All post-1962 code 400 liquidations were researched to update the delisting codes and to include additional distributions and/or additional announcement information. Five new delisting codes for liquidations have been created. (See Appendix A.3, Delisting Codes Table.) Three new distribution codes for announcement information have been created. (See Appendix A.2, Distribution Codes and Events Tables.)

  A new set of CRSP guidelines for closing an issue to further delisting research has been developed. (See Delisting Structure array in the data definition section.)

  In the current file, 260 liquidation delistings have been closed to further research and only 16 are pending further research.

- **Added Securities:** The following securities with data prior to 1997 were added to the CRSP databases in the 1997 file.

Added Securities

PERMNO	Share Code	Name	Trading Range
76870	70	BANYAN MORTGAGE INVESTORS LP II	19861008-19920925
76897	70	BANYAN MORTGAGE INVESTORS LP III	19880929-19941031
77012	70	POPE RESOURCES	19910716-19971231
78552	70	CANAL RANDOLPH LTD PARTNERSHIP	19850503-19891207
78553	40	CONSOLIDATED CAP INCOME OPP TR	19850307-19860909
78582	70	CITY INVESTING CO LIQUIDAT TRUST	19851213-19971231
78588	70	AM 1ST FEDERALLY GUAR MTG F 2 LP	19860318-19920130
78592	70	VISTA ORG PARTNERSHIP L P	19860513-19901005
78605	40	CONSOLIDAT CAP INCOME OPP TR 2	19861105-19870519
78622	71	AMERICA FIRST APT INV LP	19870213-19971231
78668	71	AM 1ST PARTICIP PREF EQ MTG L P	19880601-19971231
78691	70	CALLON CONSOLIDATED PARTNERS LP	19890112-19940916
78746	71	GENETICS INSTITUTE INC	19920129-19961231
81225	11	SCHMITT INDUSTRIES INC	19960220-19971231
81628	70	M F S COMMUNICATIONS INC	19950516-19961231
84377	11	UNISOURCE WORLWIDE INC	19970102-19971231
84382	11	CHEMFIRST INC	19970107-19971231
85409	11	CHANCELLOR BROADCASTING CO DEL	19960209-19970905
85668	11	C V THERAPEUTICS INC	19961119-19971231

- ***ts_print*:** Several improvements and corrections were made to *ts_print*, including performance for large datasets.

  A weekly calendar option for the output file has been added to *ts_print's* available date selections.

  A `NOFILL` option has been added to the `OPTION` Specifications to allow restriction of out of range data.

- ***stk_print*:** Several improvements and corrections were made to *stk_print*, including better support of adjusted prices and index level options for exceptional data.

- ***dsxlook*** and ***msxlook***, interactive browse programs, are no longer provided.

2. DATA DESCRIPTION

This section contains the following information:

1. An Overview of Stock and Index Data Organization,
2. FORTRAN Structure and Data Items,
3. C Stock Structure and Data Items, and
4. C Indices Structure and Data Items.

2.1 Data Overview

There are four basic types of information stored in CRSP databases:

1. Header information. These are identifiers with no implied time component.
2. Event arrays. Arrays can represent status changes, random events, or observations. The time of the event and relevant information is stored for each observation. There is a count of the number of observations for each type of event data.
3. Time series arrays. An observation is available for each period in an associated calendar. A beginning and ending point of valid data are available for each type of time series data. Data is stored for each period in the range – missing values are stored as placeholders if information is not available for a period.
4. Calendar arrays. Each time series is tied to an array of relevant time periods. This calendar is used in conjunction with the time series arrays to attach times to the observations.

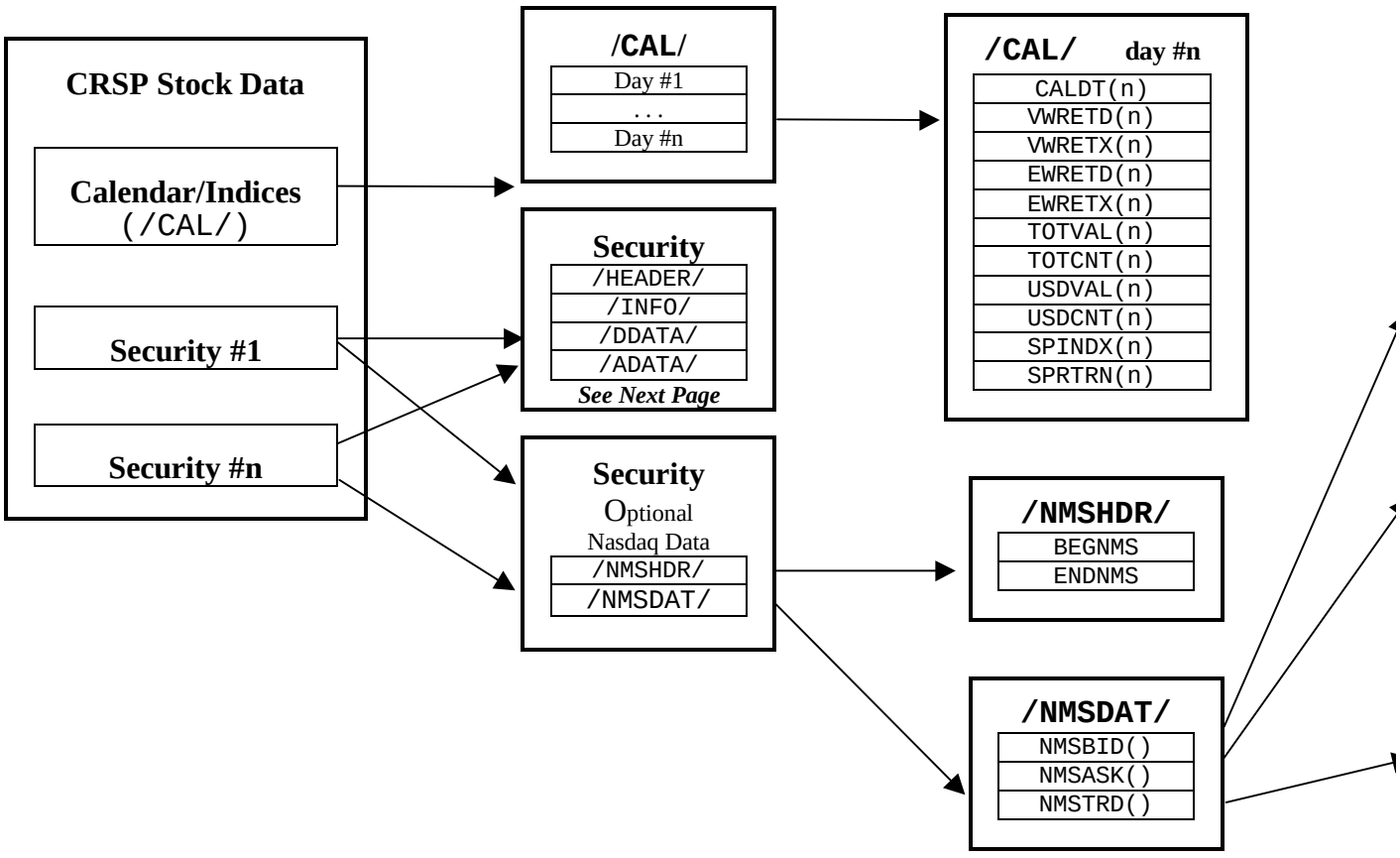
An observation can be a simple value or contain multiple components such as codes and amounts. Time series, except Portfolios, are based on calendars with the frequency of the database. In a monthly database the time series are based on a month-end trading date calendar and in a daily database the time series are based on a daily trading date calendar. Portfolio calendars are dependent on the rebalancing methodology of the specific portfolio type.

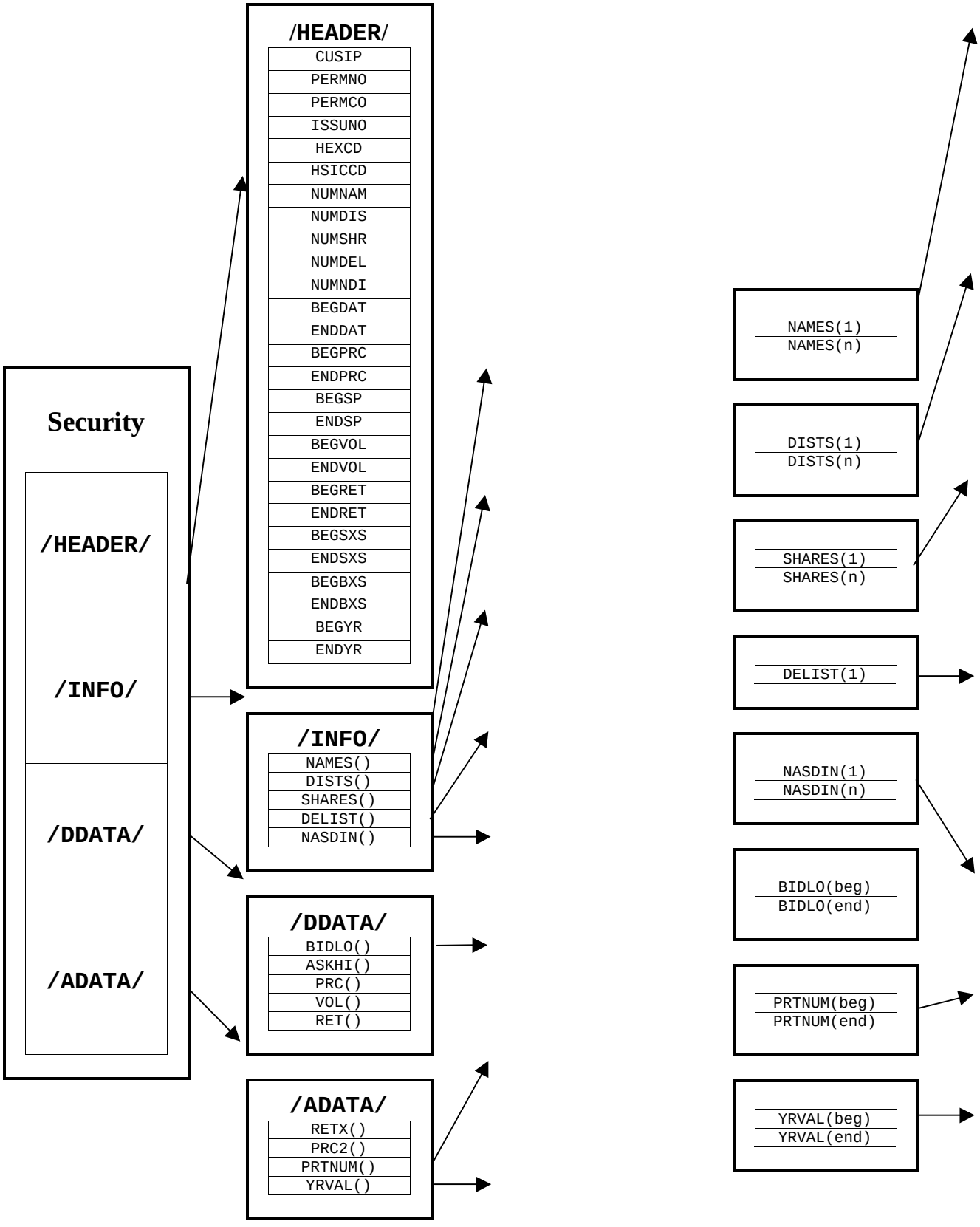
The following FORTRAN and C diagrams give a graphical overview of how the data are organized. The fundamental differences between the organization of the two are:

1. In FORTRAN there is one calendar used for all time series and information for three indices loaded with the calendar. In C, each time series can have its own calendar and all individual index series are interchangeable and identified with a permanent identifier.
2. In FORTRAN, range variables for number of observations and beginning and ending ranges are stored with the header information. In C, counts and ranges are stored with the array or time series.
3. In FORTRAN there are three standard portfolio groups, all with an implied annual calendar. In C there can be any number of portfolio groups, each with its own rebalancing calendar.

Diagram of FORTRAN Stock and Index Structure

The following diagram describes the breakdown of the FORTRAN File Structure for the CRSPAccess97 Stock Files. The index
the file.




Security
/HEADER/
/INFO/
/DDATA/
/ADATA/
/HEADER/
CUSIP
PERMNO
PERMCO
ISSUNO
HEXCD
HSICCD
NUMNAM
NUMDIS
NUMSHR
NUMDEL
NUMNDI
BEGDAT
ENDDAT
BEGPRC
ENDPRC
BEGSP
ENDSP
BEGVOL
ENDVOL
BEGRET
ENDRET
BEGSXS
ENDSXS
BEGBXS
ENDBXS
BEGYR
ENDYR
/INFO/
NAMES()
DISTS()
SHARES()
DELIST()
NASDIN()
/DDATA/
BIDLO()
ASKHI()
PRC()
VOL()
RET()
/ADATA/
RETX()
PRC2()
PRTNUM()
YRVAL()
NAMES(1)
NAMES(n)
DISTS(1)
DISTS(n)
SHARES(1)
SHARES(n)
DELIST(1)
NASDIN(1)
NASDIN(n)
BIDLO(beg)
BIDLO(end)
PRTNUM(beg)
PRTNUM(end)
YRVAL(beg)
YRVAL(end)

NAMES(n)

NAMEDT
NCUSIP/CUSP
TICKER/TICK
COMNAM/COMPNM
SHRCLS
SHRCD
EXCHCD
SICCD

DISTS(n)

DISTCD
DIVAMT
FACPR
FACSHR
DCLRDT
EXDT
RCRDDT
PAYDT
ACPERM*
ACCOMP*

SHARES(N)

SHROUT
SHRSDT
SHRFLG

DELIST(n)

DLSTDT
DLSTCD
NWPERM
NEXTDT
DLAMT*
DLRETX*
DLPRC
DLPDT*
DLRET
NWCOMP*

NASDIN(n)

TRTSDT
TRTSCD
NMSIND
NMCNT
NSINDX

PRTNUM(n)

CAP
CAPN
CAPQ

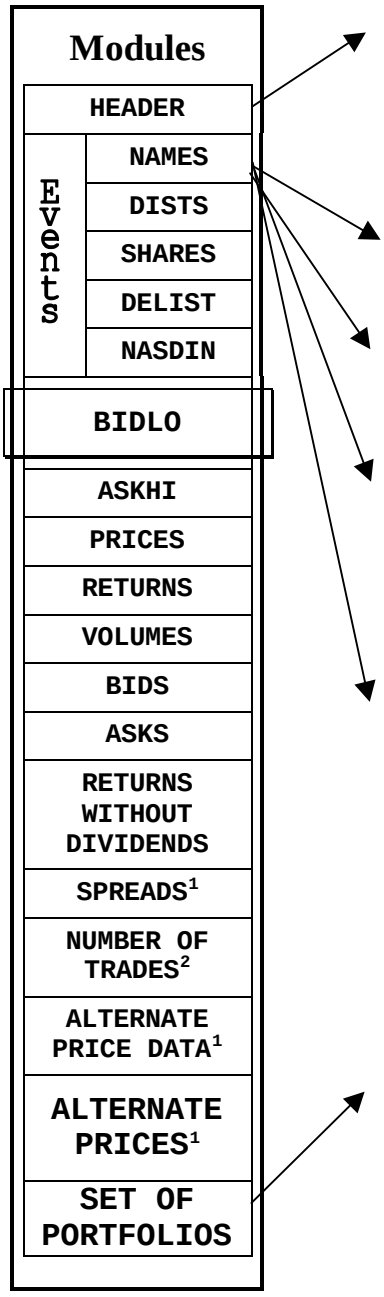
YRVAL(n)

CAP
CAPN
CAPQ

* New Variable

Diagram of Stock C Structure

The following diagram describes the breakdown of the C File Structure for the CRSPAccess97 Stock Files. A set of Calendars is common to all timeseries modules.



[1] Monthly Data Only

[2] Daily Data Only

HEADER[]

hcusip	permno	permco	compno
issuno	hexcd	hshrcd	hsiccd
begdt	enddt	dlstcd	htick
	hcomnam		

NAMES[n]

namedt	nameenddt	ncusip	ticker	comnam	shrcls
		shrcd	exchcd	siccd	

DISTS[n]

distcd	divamt	facpr	facshr	dclrdt
exdt	rcrddt	paydt	acperm	accomp

SHARES[n]

shrout	shrsdt	shrsenddt	shrflg

DELIST[n]

dlstdt	dlstcd	nwperm	nwcomp	nextdt
dlamt	dlretx	dlprc	dlpdt	dlret

NASDIN[n]

trtsdt	trtsenddt	trtscd	nmsind
	mmcnt	nsdinx	

PORT[#][n]

port	stat

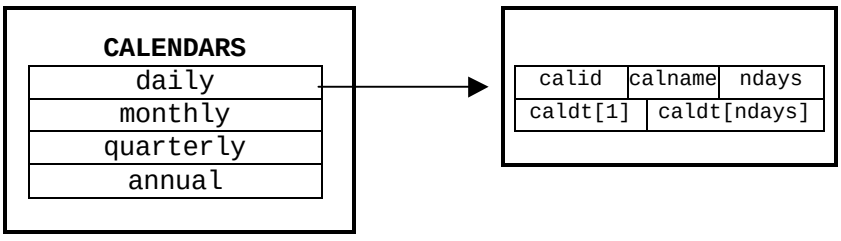
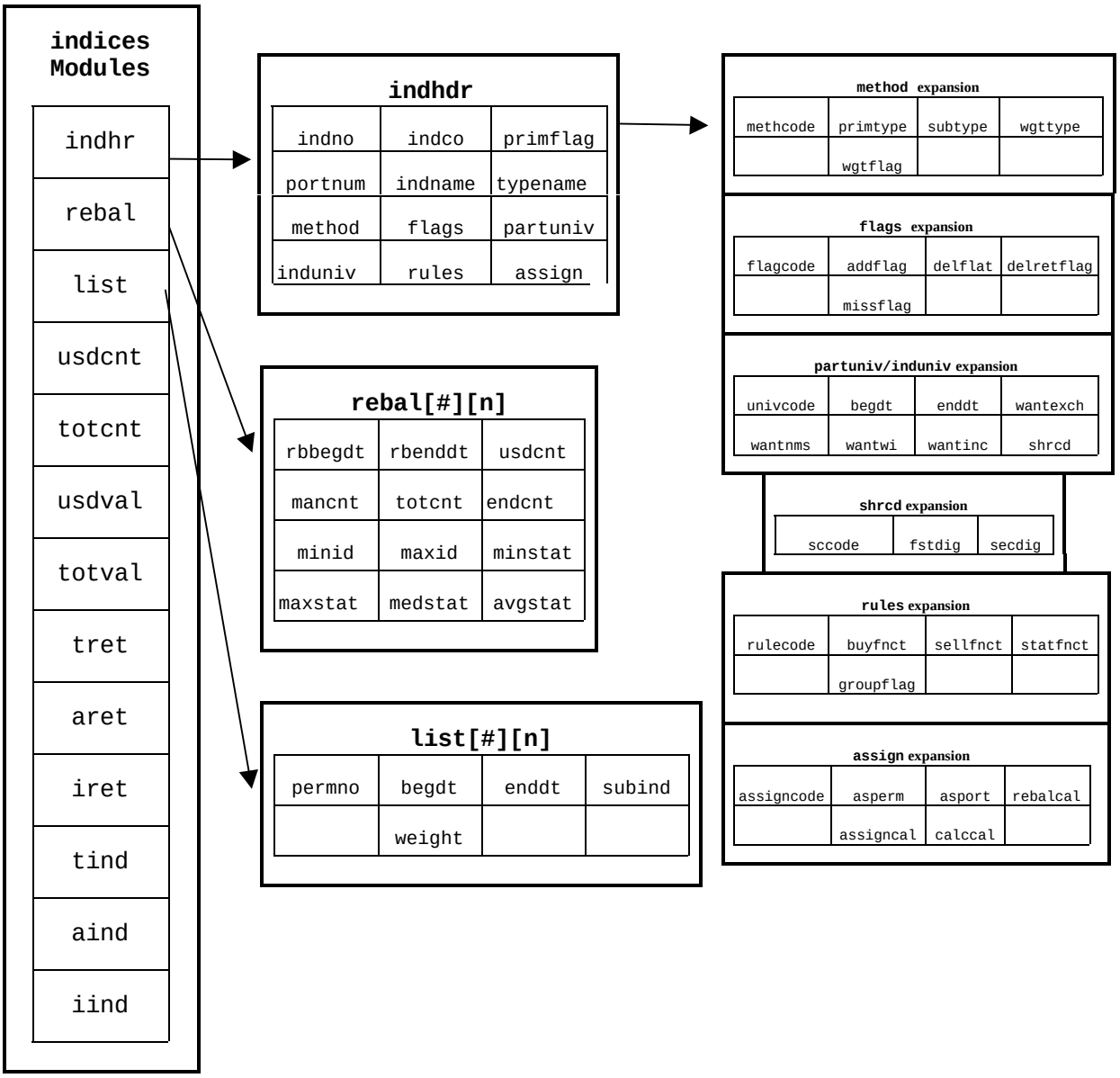


Diagram of C Index Structure

The following diagram describes the organization of the index data for CRSP Access97 Stock and Indices Files. Index data are organized by individual series or group under a CRSP-assigned identifier `INDNO`. All modules except `indhdr` represent a set of one or more individual series. The same calendars available to the stock structure are also used for the index data.



2.2 FORTRAN Variable Definitions

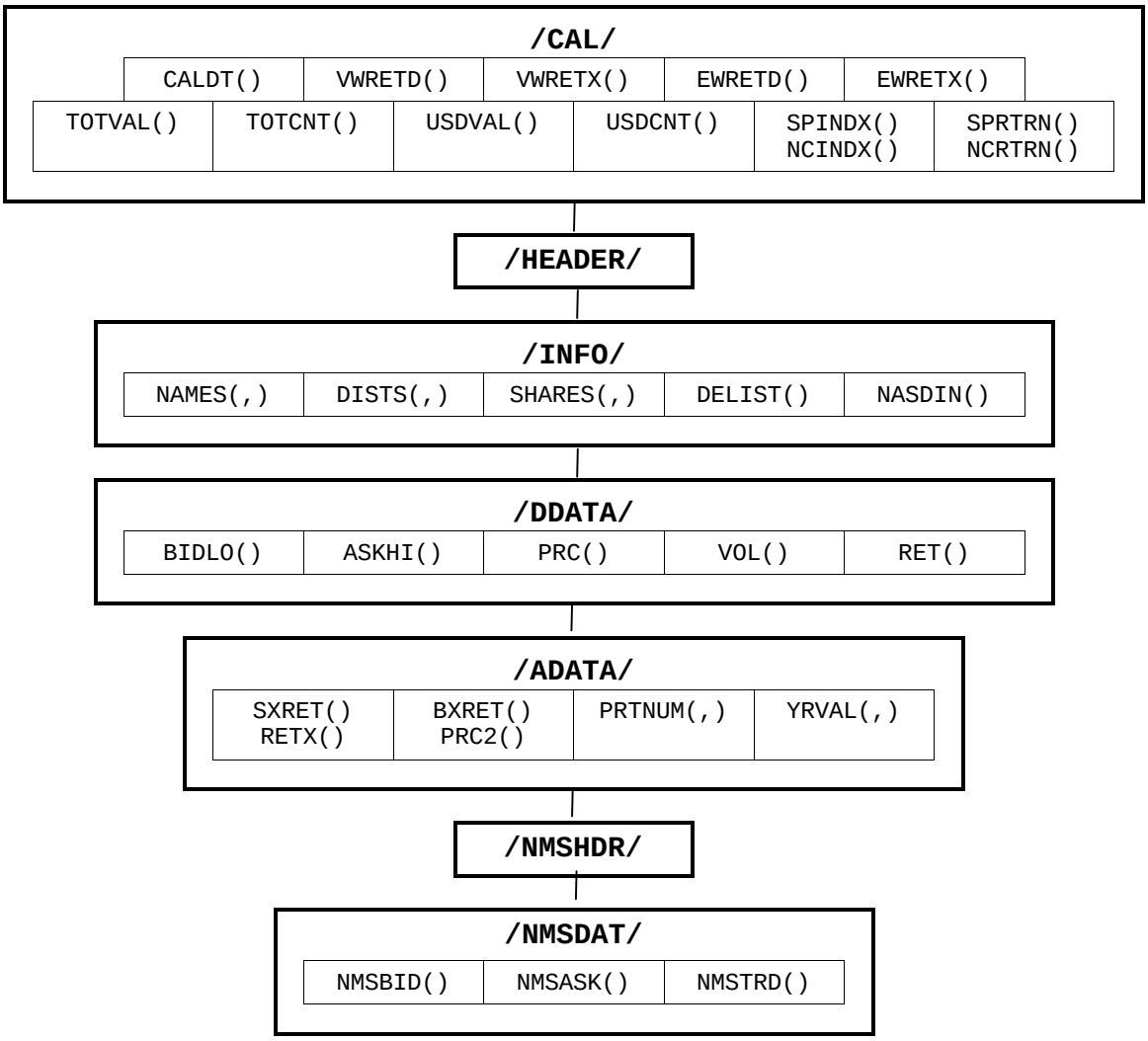
This section gives detailed descriptions of the variables provided in the file. Each description is preceded with a line containing three items:

1. The variable name - delimited by slashes ("/") if it is a common block or, if it is an array, followed by parentheses ("()") and enclosed commas to indicate the number of dimensions,
2. A short description of the variable,
3. The FORTRAN type for the variable.

The interpretation of these items is slightly different in the FORTRAN /INFO/ section. The data are grouped logically, rather than by FORTRAN type. The descriptions are divided by common block and indented within each array.

Dates in this section are described as YYMMDD format. The underlying data is stored as YYYYMMDD. See CRSPAccess97 Stock File Programmers Guide for switching formats in FORTRAN.

The following diagram shows the FORTRAN common blocks and the arrays contained within them.



/CAL/ — Calendar/Indices Arrays

The NYSE/AMEX and the combined NYSE/AMEX/Nasdaq Index files are contained in the /CAL/ common block and furnish three types of data:

Trading calendar dates;

Returns and other summaries on market portfolios;

Standard and Poor's 500 Composite and Nasdaq Composite indices data.

On the daily files, the calendar lists all NYSE/AMEX and Nasdaq trading dates from July 2, 1962 through end of the file. The beginning date for Nasdaq data is day 2610, which corresponds to December 14, 1972. This calendar includes only active market days and thus excludes weekends and market holidays. The monthly index file has all month-end trading days on the NYSE from December 1925 through the last month of the file. The AMEX and Nasdaq monthly data begins in month 440, July 1962, and in month 565, December 1972, respectively.

CRSP derives four indices for the market provided on each file, where the market refers to the exchanges covered by the files opened, NYSE/AMEX/Nasdaq, in combination or individually.

1. Value-weighted returns including all distributions (VWRETD);
2. Value-weighted returns excluding dividends (VWRETX);
3. Equally-weighted returns including all distributions (EWRETD);
4. Equally-weighted returns excluding dividends (EWRETX).

For a security n on a trading day I, the return $(r_n(I))$ is defined as the change in the total dollar value of an investment in that security, over some period of time, per dollar of initial investment. The description of array RET in common block /DDATA/ contains details of how returns are calculated.

The return on a portfolio $(R(I))$ is calculated as the weighted average of the returns for the individual stocks in the portfolio:

$$R(I) = \frac{\sum_n w_n(I) r_n(I)}{\sum_n w_n(I)}$$

In a value-weighted portfolio, the weight $(w_n(I))$ assigned to security *n's* return is its total market value $v_n(I)$. CRSP defines the market value of a security $(v_n(I))$ as the product of its price $(p_n(I-1))$ and its number of shares outstanding $(s_n(I-1))$,[3] i.e.,

$$w_n(I) \equiv p_n(I-1)s_n(I-1)$$

The total market value of the portfolio $V(I) = \sum_n v_n(I) = \sum_n w_n(I)$ is given in the USDVAL array. The value-weighted returns form the VWRETD and VWRETX arrays.

In an equally-weighted portfolio, $w_n(I)=1$ for every stock. Such a portfolio would consist of N stocks, with the same number of shares purchased for each stock. The number of stocks used in the equally-weighted portfolio

3 The number of shares outstanding for a security on a given day $(s_n(I))$ is derived from the SHARES array.

$V(I)=\sum_{v} w_n(I)=\sum_{n} 1=N(I)$ is given in the USDCNT array. The equally-weighted returns form the EWRETD and EWRETX arrays.

CRSP also furnishes closing levels of the Standard & Poor's Composite Index (SPINDX) for use with the NYSE/AMEX file, the combined NYSE/AMEX/Nasdaq file and the Nasdaq Composite Index (NCINDX) for use with the Nasdaq file and the combined NYSE/AMEX/Nasdaq file, and calculates returns for the index selected (SPRTRN or NCRTRN).

CALDT() **Calendar Date** **INTEGER**

CALDT contains the trading dates for the New York Stock Exchange, the American Stock Exchange and The Nasdaq Stock Market℠ files. These dates are stored in the form YYMMDD (year, month, date). Examples from the daily file and monthly file are listed below.

Daily File Examples:
CALDT(1) = 620702 (July 2, 1962)
CALDT(2610) = 721214 (December 14, 1972)
CALDT(8939) = 971231 (December 31, 1997)
CALDT(8939) = CALDT (NDAYS) = 971231

Monthly File Examples:
CALDT(1) = 251231
CALDT(440) = 620731
CALDT(565) = 721229
CALDT(865) = CALDT (NDAYS) = 971231

VWRETD() **Value-Weighted Return (including all distributions) Indices** **REAL**

VWRETD indices contain either the daily or monthly returns, including all distributions, on a value-weighted market portfolio (excluding American Depository Receipts (ADRs)).

VWRETX() **Value-Weighted Return (excluding dividends) Indices** **REAL**

VWRETX indices contain, depending on your subscription, daily or monthly returns, excluding all dividends, on a value-weighted market portfolio (excluding ADRs).

EWRETD() **Equal-Weighted Return (including all distributions) Indices** **REAL**

EWRETD indices contain, depending on your subscription, daily or monthly returns, including all distributions, on an equally-weighted market portfolio (including ADRs).

EWRETX() **Equal-Weighted Return (excluding dividends) Indices** **REAL**

EWRETX indices contain, depending on your subscription, daily or monthly returns, excluding all dividends, on an equally-weighted market portfolio (including ADRs).

TOTVAL() **Total Market Value** **REAL**

TOTVAL(I) contains the total market value for a given market, in $1000's, for all non-ADR securities with valid prices[4] on day I.

4 A valid price in the file is either a negative bid/ask average or a positive closing price.

TOTCNT() **Total Market Count** **INTEGER**

TOTCNT(I) is the number of stocks in the current file with a valid price[2] on day I.

USDVAL() **Market Value of Securities Used** **REAL**

USDVAL(I) is the total market value, in $1000's, of all securities that are used in the value-weighted indices on day I. To be used for a value-weighing, a security cannot be an ADR and must have valid prices[2] on the current day and the previous trading day.

USDCNT() **Count of Securities Used** **INTEGER**

USDCNT(I) is the number of stocks in the current file used in the equal-weighted indices on day I. To be used for equal-weighting, a security must have valid prices[2] on the current day and the previous trading period I-1.

SPINDX() **Level on the Standard & Poor's Composite Index**[5] **REAL**

SPINDX(I) is the level of the Standard & Poor's 500 Composite Index (prior to March 1957, 90-stock index) at the end of the trading day or month. This data is collected from publicly available sources such as the Dow Jones News Service, *The Wall Street Journal* or *Standard & Poor's Statistical Service.* SPINDX(I) does not include dividends. The index indicates the change in price of the component securities.

SPRTRN() **Return on the Standard & Poor's Composite Index**[3] **REAL**

SPRTRN(I) is the return on the Standard & Poor's Composite Index for period I as defined as:

$$\frac{SPINDX\ \ (I)}{SPINDX\ \ (I-1)} - 1$$

NCINDX() **Nasdaq Composite Index**[3] **REAL**

NCINDX(I) is the level of the Nasdaq Composite Index at the end of the trading day or month. This data is collected from Nasdaq and does not include dividends. The index indicates the change in price of the component securities.

NCRTRN() **Nasdaq Composite Return**[3] **REAL**

NCRTRN(I) is the return on the Nasdaq Composite Index for period I as defined as:

$$\frac{NCINDX\ \ (I)}{NCINDX\ \ (I-1)} - 1$$

5 One will receive either the NCINDX and NCRTRN and/or the SPINDX and SPRTRN. This is determined by the product subscription.

/HEADER/ — Identification And Date Range Variables

The header structure contains fields that identify an issue and specify the date range for which data is available. The identifying fields in this structure are the unique issue identifiers assigned by CRSP and CUSIP. The remaining fields in /HEADER/ describe the individual security's ranges for arrays in /INFO/, /DDATA/, and /ADATA/.

CUSIP **CUSIP Identifier** **CHARACTER*8**

CUSIP is the latest eight-character CUSIP identifier for the security through the end of the file. CUSIP identifiers are supplied to CRSP by the CUSIP Service Bureau, Standard & Poor's, a division of McGraw-Hill, Inc., American Bankers Associate database, Copyright 1997. See Appendix A.6 for more CUSIP copyright information.

CUSIP identifiers were first assigned in 1968 as integers and expanded in 1984 to include alphabetic characters. The first six characters (including leading zeroes) identify the issuer, while the last two characters identify the issue. CUSIP issuer identifiers are assigned to maintain an approximately alphabetical sequence. The CUSIP identifier may change for a security if its name or capital structure changes.

No header or historical CUSIPs are reused on our files. For securities no longer in existence or that were never assigned an official CUSIP identifier, CRSP has assigned a dummy CUSIP identifier for use in this field in accordance with the rules published in the CUSIP Directory. There are two potential dummy CUSIPs which are assigned by CRSP. One, ***99*9*, (containing a 9 in the 4th, 5th and 7th character positions) represents a CRSP assigned CUSIP with a dummy issuer number (the first 6 character positions) and a dummy issue number (the last 2 character positions). The other, ******9*, (containing a 9 in the 7 th character position) represents a CRSP-assigned CUSIP with a real issuer number but a dummy issue number. For example:

A CUSIP such as 12399099 or 12345699 is assigned by CRSP, and an identifier such as 12345610 is assigned by the CUSIP Agency.

Securities actively traded on an international basis, domiciled outside the United States and Canada, will be identified by a CINS (CUSIP International Numbering System) number. CINS numbers employ the same Issuer (6 characters)/Issue (2 characters) 8-character identifier system set by the CUSIP Numbering System. It is important to note that the first portion of a CINS code is always represented by an alpha character, signifying the issuer's country code (domicile) or geographic region. See the current CUSIP Directory for more information. See Appendix A.1 for a list of CINS country codes.

PERMNO **CRSP Permanent Number** **INTEGER**

PERMNO is a unique five-digit permanent identifier assigned by CRSP to each security in the file. Unlike CUSIP, TICKER, and COMNAM, the PERMNO neither changes during an issue's trading history, nor is reassigned after an issue ceases trading. The user may track a security through its entire trading history in CRSP's files with one PERMNO, regardless of name changes or capital structure changes. A security that is included on both CRSP's Nasdaq and NYSE/AMEX files will have the same CRSP permanent number in both files. The file is sorted by this field.

PERMCO **CRSP Permanent Company Number** **INTEGER**

PERMCO is a unique permanent identifier assigned by CRSP to all companies with issues on a CRSP file. This number is permanent for all securities issued by this company regardless of name changes. The mnemonics PERMCO and COMPNO are interchangeable.

If the value in this field is less than 20,000, CRSP has used the Nasdaq-assigned Company Number, for when the company had an issue trading on The Nasdaq Stock Market SM. A value over 20,000 indicates that the number was assigned by CRSP.

ISSUNO **Nasdaq Issue Number** **INTEGER**

ISSUNO is a unique integer assigned by the National Association of Securities Dealers (NASD) to each listed security on the Nasdaq Stock Market SM. It is this issue-specific identifier which differentiates securities issued by the same company. If the issue number is unknown, ISSUNO is set to zero. If an NYSE/AMEX security was ever traded on Nasdaq, this number is set to the latest issue number assigned when it was trading on Nasdaq. The ISSUNO in the CRSP file may change if Nasdaq assigns a new number to an issue CRSP considers to be a continuation of an existing issue.

HEXCD **Header Exchange Code** **INTEGER**

HEXCD displays the current exchange code listed for a specific security. Valid HEXCD's are 1, 2, or 3, which correspond to the NYSE, AMEX, and Nasdaq respectively. Other exchange codes are not included in HEXCD.

HSICCD **Header Standard Industrial Classification (SIC) Code** **INTEGER**

HSICCD is the last non-zero Standard Industrial Classification (SIC) Code found in a specific security's name structure. HSICCD will be zero for companies where CRSP's data source did not provide SIC Codes.

NUMNAM **Number of Name Structures** **INTEGER**

NUMNAM is the highest number of name structures for a given security in the file. Every name structure contains information for each of the NAMES array's 8 variables; 1. Name Date (NAMEDT), 2. Cusip Name (NCUSIP & CUSP), 3. Exchange Ticker Symbol (TICKER & TICK), 4. Company Name (COMNAM & COMPNM), 5. Share Class (SHRCLS), 6. Share Code (SHRCD), 7. Exchange Code (EXCHCD), and 8. Standard Industrial Classification Code, (SICCD).

The Number of Name Structures increases by one every time any of the 8 NAMES variables changes. NUMNAM is greater than or equal to one. In NAMES(*,NUMNAM), the * refers to the most recent name structure available in the file for the security.

NUMDIS **Number of Distribution Structures** **INTEGER**

NUMDIS is the highest number of distribution structures for a given security in the file. Every distribution structure contains information for each of the distribution (DISTS & RDISTS) array's 8 variables; 1. Distribution Code (DISTCD), 2. Dividend Cash Amount (DIVAMT), 3. Factor to adjust Price (FACPR), 4. Factor to Adjust Shares Outstanding (FACSHR), 5. Declaration Date (DCLRDT), 6. Ex-Distribution Date (EXDT), 7. Record Date (RCRDDT), and 8. Payment date (PAYDT).

The number of distribution structures increases by one for every distribution event. NUMDIS is greater than or equal to zero. NUMDIS can be zero if the security has no distribution history. In DISTS(*,NUMDIS) or RDISTS(*,NUMDIS), the * refers to the most recent distribution structure available in the file for the security.

NUMSHR **Number of Shares Structures** **INTEGER**

NUMSHR is the highest number of share structures for a given security in the file. Every share structure contains information for each of the share's (SHARES) array's 3 variables; 1. Number of

Shares Outstanding (SHROUT), 2. Date of Shares Structure (SHRSDT), and 3. Share Flag (SHRFLG).

The number of shares structures increases by one every time any of the 3 SHARES variables changes. NUMSHR is greater than or equal to zero. NUMSHR can be zero if no share information is available. In SHARES(*,NUMSHR), the * refers to the most recent share structure available in the file for the security.

NUMDEL **Number of Delisting Structures** **INTEGER**

NUMDEL is the highest delisting structure for a given security in the file. In this version of the file, there is only one delisting structure, and therefore NUMDEL always equals one. Every delisting structure contains information for each of the delistings (DELIST/RDELIS) array's 10 variables; 1. Delisting Date (DLSTDT), 2. Delisting Code (DLSTCD), 3. New CRSP Permanent Number (NWPERM), 4. Date of Next Available Information (NEXTDT), 5. Delisting Amount (DLAMT), 6. Delisting Return without Dividends (DLRETX), 7. Delisting Price (DLPRC), 8. Delisting Payment Date (DLPDT), 9. Delisting Return (DLRET) and 10. New CRSP Permanent Company Number (NWCOMP).

In DELIST(*,NUMDEL) or RDELIS(*,NUMDEL) * refers to the most recent delisting structure for the security on the file. NUMDEL is always equal to one.

NUMNDI **Number of Nasdaq Information Structures** **INTEGER**

NUMNDI is the highest number of Nasdaq structures for a given security in the file. In this version of the file, there is only one delisting structure. Every delisting structure contains information for each of the Nasdaq information structure's (NASDIN) array's 4 variables; 1. Traits Date (TRTSDT), 2. Traits Code (TRTSCD), 3. Nasdaq National Market Indicator (NMSIND) and 4. Market Maker Count (MMCNT).

In NASDIN(*,NUMNDI) * refers to the most recent Nasdaq structure for the security on the file. NUMNDI is zero for all Nasdaq securities that stopped trading prior to November 1, 1982 and for all NYSE/AMEX securities.

BEGDAT **Beginning of Data** **INTEGER**

BEGDAT is the first valid value of a security's non-empty daily or monthly data arrays. BEGDAT is always greater than zero. CALDT(BEGDAT) is the date that daily or monthly data begins for the security.

ENDDAT **End of Data** **INTEGER**

ENDDAT is the last valid value of a security's non-empty daily or monthly data arrays. ENDDAT is always greater than zero. CALDT(ENDDAT) is the date that daily or monthly data begins for the security.

BEGPRC **Beginning of Price Data** **INTEGER**

BEGPRC is the first valid value of a security's price array PRC(). If there is no price data in the file, BEGPRC is zero. CALDT(BEGPRC) is the date of a security's first price.

ENDPRC **End of Price Data** **INTEGER**

ENDPRC is the last valid value of a security's price array PRC(). If there is no price data in the file, then ENDPRC is zero. CALDT(ENDPRC) is the date of a security's last price.

BEGSP **First Secondary Price** **INTEGER**

BEGSP is the first valid value of a security's bid/low (BIDLO()) and ask/high (ASKHI()) arrays. If there is no price data in the file, then BDGSP is zero. CALDT(BEGSP) is the date of a securities first bid/low or ask/high data.

ENDSP **Last Secondary Price** **INTEGER**

ENDSP is the last valid value of a security's bid/low BIDLO()and ask/high (ASKHI()) arrays. If there is no price data in the file, then ENDSP is zero. CALDT(ENDSP) is the date of a security's last price.

BEGVOL **Beginning of Volume Data** **INTEGER**

BEGVOL is the first valid value of a security's share volume (VOL())array. If there is no share volume data in the file, then BEGVOL is zero. CALDT(BEGVOL) is the date of a securities first share volume data.

ENDVOL **End of Volume Data** **INTEGER**

ENDVOL is the last valid value of a security's share volume (VOL())array. If there is no share volume data in the file, then ENDVOL is zero. CALDT(ENDVOL) is the date of a security's last share volume data.

BEGRET **Beginning of Return Data** **INTEGER**

BEGRET is the first valid value of a security's returns (RET())array. If there is no returns data in the file, then BEGRET is zero. CALDT(BEGRET) is the date of a securities first returns data.

ENDRET **End of Return Data** **INTEGER**

ENDRET is the last valid value of a security's returns with dividends (RET())array. If there is no returns with dividends data in the file, then ENDRET is zero. CALDT(ENDRET) is the date of a security's last returns data.

BEGSXS **Beginning of Excess Return Data** **INTEGER**

BEGSXS is analogous to BEGPRC, except that it is used with the standard deviation excess returns array (SXRET()). In monthly or daily files with no excess returns loaded, BEGSXS should be interpreted as BEGRTX.

ENDSXS **End of Excess Return Data** **INTEGER**

ENDSXS is analogous to ENDPRC, except that it is used with the standard deviation excess returns array (SXRET()). In monthly or daily files with no excess returns loaded, ENDSXS should be interpreted as ENDRTX.

BEGBXS **Beginning of Beta Excess Return Data** **INTEGER**

BEGBXS is analogous to BEGPRC, except that it is used with the beta excess returns array (BXRET()). In monthly or daily files with no excess returns loaded, BEGBXS should be interpreted as BEGPR2.

ENDBXS **End of Beta Excess Return Data** **INTEGER**

ENDBXS is analogous to ENDPRC, except that it is used with the beta excess returns array (BXRET()). In monthly or daily files with no excess returns loaded, ENDBXS should be interpreted as ENDPR2.

BEGYR **First Year of Yearly Data** **INTEGER**

BEGYR is an index containing the first yearly data , values, and portfolio assignments available for a specific security. BEGYR plus 24 is the actual year, "YY", of data corresponding to this index.

ENDYR **Last Year of Yearly Data** **INTEGER**

ENDYR is in index containing the most recent yearly data, values, and portfolio assignments available for a specific security. ENDYR plus 24 is the actual year, "YY", of data corresponding to this index.

BEGRTX **Beginning of the Return Data Without Dividends** **INTEGER**

BEGRTX is the first valid value of a security's returns without dividends (RETX())array. If there are no returns without dividends data in the file, then BEGRTX is zero. CALDT(BEGRTX) is the date of a securities first returns without dividends data.

ENDRTX **End of the Return Data Without Dividends** **INTEGER**

ENDRTX is the last valid value of a security's returns without dividends (RETX())array. If there are no returns without dividends data in the file, then ENDRTX is zero. CALDT(ENDRTX) is the date of a security's last returns without dividends data.

BEGPR2 **Beginning of the Monthly Secondary Price Data** **INTEGER**

BEGPR2 is the first valid value of a security's secondary price (PRC2())array. If there is no secondary price data in the file, then BEGPR2 is zero. CALDT(BEGPR2) is the date of a securities first secondary price data.

ENDPR2 **End of Monthly Secondary Price Data** **INTEGER**

ENDPR2 is the last valid value of a security's secondary price (PRC2())array. If there is no secondary price data in the file, then ENDPR2 is zero. CALDT(ENDPR2) is the date of a security's last secondary price data.

/INFO/ — Name, Distribution, Share, Delisting, and Nasdaq Information Arrays

The /INFO/ common block consists of five structure arrays, containing name (NAMES, CNAMES), distribution (DISTS, RDISTS), share (SHARES), delisting (DELIST, RDELIS), and Nasdaq (NASDIN) information. The Information Structure (/INFO/) contains the complete history of all the observations of the pertinent events (names, shares, etc.) for that security. Each array has two levels. The first level references individual structure arrays above, and the second level references the individual observations within each of the 5 structure arrays.

Each individual observation in this section is followed by the parameters used to access its fields; a field should be accessed with its parameter whenever possible. For example, the "distribution code" field in the second level of the distribution history and can be accessed as DISTS(DISTCD,2). The field names are declared as constants. They do not hold the data themselves, and cannot be assigned new values.

Fields that are to be interpreted as INTEGER can be accessed through these INTEGER arrays. Non-INTEGER fields usually must be accessed differently. REAL data, found in the distribution and delisting histories, can be accessed through the RDISTS(,) and RDELIS(,) arrays: REAL arrays such as RDISTS() and RDELIS() are comparable in CRSP programs to the INTEGER arrays, DISTS(,) and DELIST(,) respectively. A distribution structure, for instance, contains the "dividend amount", a REAL value that can be accessed as RDISTS(DIVAMT,*). CHARACTER data, found in the name history, can be accessed through the CNAMES array in four-character groups, such as the CUSIP or CUSP, or through utility functions.

Dates in this section are described as YYMMDD format. The underlying data is stored as YYYYMMDD. See CRSPAccess97 Stock File Programmers Guide for switching formats in FORTRAN.

NAMES(,) **Security Name** **INTEGER**
CNAMES(,) **Security Name** **CHARACTER*4**

Each security has at least one name structure, which contains the following eight fields. The variables in these fields describe the security's name history. If the CUSIP, name, exchange, ticker symbol, or share class changes during the security's trading history, a new name structure is added, with the date of the change (NAMEDT).

Name histories may include periods, possibly outside the data range, when the security is trading on a different exchange or is not trading at all. The exchange code (EXCHCD) contains more detailed information on trading status and location for a given date range.

NAMEDT **Name Date** **INTEGER**

NAMES(NAMEDT,I) is the starting date of name structure I, stored in YYMMDD format. If the CUSIP, name, exchange or share class changes, CRSP adds a new name structure that records the change and the date the change became effective. NAMEDT is the date associated with a specific name structure. For example:

If the company had changed its name on April 28, 1976, and had two existing name structures, the new name and other information would appear in a new name structure I (where I=3) with NAMES(NAMEDT,3) having the value 760428.

NCUSIP **CUSIP Name** **2 CHARACTER*4**
CUSP **CUSIP Name** **CHARACTER*8 FUNCTION**

CUSP(I) and NAMES(NCUSIP,I) both refer to the CUSIP identifier of the I^{th} name structure. NAMES(NCUSIP,I) contains its first four characters and NAMES(NCUSIP+1,I) contains its last four characters. They can be printed using the FORTRAN format "2A4". The CUSP function returns the name CUSIP as one

CHARACTER*8 value, rather than as two halves. The eight character function makes it easier to compare the names CUSIP with /HEADER/CUSIP.

The CUSIP Agency will often change an issue's CUSIP identifier to reflect a change of name or capital structure. CRSP has preserved all CUSIPs that have been assigned to a given issue. NCUSIP may be a blank string if the name structure predates the CUSIP Bureau.

For more details of the CUSIP identifier see the /HEADER/CUSIP variable description.

TICKER **Exchange Ticker Symbol** **2 CHARACTER*4**
TICK **Exchange Ticker Symbol** **CHARACTER*8 FUNCTION**

The combination of ticker, exchange, and date uniquely identifies a security. A ticker may be one to three characters for NYSE and AMEX securities or four or five characters for Nasdaq securities. NAMES(TICKER,I) and TICK(I) both refer to the ticker of the I^{th} name structure.

Nasdaq trading tickers have four base characters and a fifth character suffix that provides information about an issue's type or temporary information about an issue's status. CRSP only includes the suffix when it provides permanent descriptive information. This table describes the suffixes appearing on the CRSP file.

Nasdaq 5th Character Suffixes

Suffix	Definition
A	Class A
B	Class B
F	Companies incorporated outside the US
S	Shares of Beneficial Interest
U	Unit
V	When-issued
Y	ADR
Z	Miscellaneous common issues

Occasionally Nasdaq will add two additional suffixes to the base ticker to identify certain issues. However, because the Nasdaq ticker field only allows for five characters, one letter of the base ticker will be dropped. For example:

If a foreign company with a class A stock has a base ticker symbol ABCD, Nasdaq adds two additional characters, A and F. Due to Nasdaq's limited fields, they will delete a letter from the base ticker, so ABCDAF would be truncated to ABCAF.

Nasdaq tickers in an issue's name history are presumed to represent legitimate trading symbols for that issue at some point in time, although these symbols may be listed out of proper chronological sequence. In addition, the Nasdaq file ticker symbols provided do not necessarily comprise a definitive list of all symbols used throughout an issue's trading history. Due to source limitations, the TICKER field may be blank in name histories of Nasdaq securities that stopped trading from the early 1970's through the early 1980's. NYSE tickers prior to July 1962 are blank.

COMNAM **Company Name** **8 CHARACTER*4**
COMPNM **Company Name** **CHARACTER*32 FUNCTION**

NAMES(COMNAM,I) is the name of the company as of the date specified in NAMES(NAMEDT,I). CRSP allocates a 32 character name description field for all securities. Preference is given to the spellings and abbreviations provided in Standard & Poor's CUSIP Directory. In cases where all name sources provide descriptions in excess of 32 characters, CRSP furnishes its own abbreviations. The I^{th} company name is stored in four character groups in NAMES(COMNAM,I) through NAMES(COMNAM+7,I). The COMPNM function returns the company name as one CHARACTER*32 value, rather than as eight parts.

SHRCLS **Share Class** **CHARACTER*4**

NAMES(SHRCLS,I) describes the class of share and is generally blank. Any letter that identifies the class of stock (e.g., "A" for class A common) is contained in this field, left justified, and padded with three blank spaces.

SHRCD **Share Code** **INTEGER**

NAMES(SHRCD,I) is a two-digit code describing the type of shares traded. The first digit describes the type of security traded.

First Digit - Share Code - Security Type

Code	Definition
1	Ordinary Common Shares
2	Certificates
3	ADRs (American Depository Receipts)
4	SBIs (Shares of Beneficial Interest)
7	Units (Depository Units, Units of Beneficial Interest, Units of Limited Partnership Interest, Depository Receipts, etc.)

Note: "Units" (code 7) does not represent combinations of common stock and anything else, such as warrants.

The second digit gives more detailed information about the type of security traded.

Second Digit - Share Code - Security Type

Code	Definition
0	Securities which have not been further defined.
1	Securities which need not be further defined.
2	Companies incorporated outside the US
3	Americus Trust Components (Primes and Scores).
4	Closed-end funds.
5	Closed-end fund companies incorporated outside the US
8	REIT's (Real Estate Investment Trusts).

For example: A SHRCD of 14 would represent ordinary common shares of a closed-end fund.

EXCHCD **Exchange Code** **INTEGER**

NAMES(EXCHCD,I) is a code indicating the exchange on which a security is listed. Normal exchange codes are respectively 1,2, and 3 for NYSE, AMEX and the Nasdaq Stock Market[SM]. An exchange code of zero indicates that a security is either trading on an unknown exchange, or is temporarily not trading at all. Adding 30 to the normal exchange codes (31, 32 & 33) delineates when-issued trading, such as during a reorganization. See Appendix A.1 for additional Exchange Code information.

Exchange Codes

Code	Definition
-2	suspended by the NYSE or AMEX
-1	halted by the NYSE or AMEX
0	not listed on exchange of current file
1	New York Stock Exchange
2	American Stock Exchange
3	The Nasdaq Stock Market[SM]
31	When-issued trading on the New York Stock Exchange
32	When-issued trading on the American Stock Exchange
33	When-issued trading on The Nasdaq Stock Market[SM]

SICCD **Standard Industrial Classification Code** **INTEGER**

NAMES(SICCD,I) is the Standard Industrial Classification code. SIC Code is used to group companies with similar products or services.

The Standard Industrial Classification Manual contains descriptions of categories recognized by the US Government. SIC Code is an integer between 100 and 9999. The first two digits refer to a major group. The first three digits refer to an industry group. All four digits indicate an industry. Missing SIC Codes are set to 0. Most Nasdaq SIC codes have only the first three digits; CRSP has added a fourth digit of zero.

DISTS(,) **Distribution Structure** **INTEGER**
RDISTS(,) **Distribution Structure** **REAL**

The distribution structure array has entries describing cash dividends, capital adjustments, and other distributions to shareholders. Each distribution structure includes:

- The type of distribution,
- The amount of distribution,
- A factor to adjust price,
- A factor to adjust shares outstanding, and
- The dates associated with the distribution (i.e., declare, ex-dividend, record, and payment).

If a distribution event has more than one component, CRSP codes each component of the event separately. All components of a distribution event share the same EXDT.

DISTS(*,*) should be used to access all the integer values in the distribution structure, while RDISTS(*,*) should be used to access the real values.

DISTCD **Distribution Code** **INTEGER**

CRSP describes distributions with a four-digit code. The first digit describes the type of distribution. The second digit describes the payment method. The third digit augments the type denoted by the first digit. The fourth digit provides information regarding the tax status of the distribution. The coding scheme is described in detail in Appendix A.2. In addition, Appendix A.2 contains a brief definition of some of the DISTCDs found in the CRSP stock files.

CRSP has not verified the tax status of ordinary cash dividends since 1987. Instead, CRSP assigned the most common tax code, taxable as dividend, to ordinary dividends. CRSP does verify the tax status of stock distributions and distributions associated with rights offerings, spin-offs, liquidations, mergers, reorganizations, and exchanges.

DIVAMT **Dividend Cash Amount** **REAL**

RDISTS(DIVAMT,I) is the US dollar value per share of distributions resulting from cash dividends, spin-offs, mergers, exchanges, reorganizations, liquidations, and rights issues.

If the Distribution code is 6225, a nonzero amount represents an offer price given to a certain amount of shares. For these cases, the dollar value per share is actually RDISTS(DIVAMT,I) multiplied by the percent of shares accepted by the offer, where the percent of shares accepted can be derived by multiplying RDISTS(FACPR,I) by negative one. See FACPR.

FACPR **Factor to Adjust Price** **REAL**

RDISTS(FACPR,I) is used to adjust stock prices after a distribution so that a comparison can be made on an equivalent basis between prices before and after the distribution.

RDISTS(FACPR,I) equals RDISTS(FACSHR,I) for most distribution events. There are three types of distributions where this is the case:

Cash dividends, RDISTS(FACPR,I) is set to zero;

Cases of mergers, total liquidations, or exchanges where the security disappeared, RDISTS(FACPR,I) is set to negative one by convention;

For stock dividends and splits, RDISTS(FACPR,I) is the number of additional shares per old share issued:

$$FACPR = \frac{s(t) - s(t')}{s(t')} = \frac{s(t)}{s(t')} - 1$$

where *s(t)* is the number of shares outstanding, *t* is a date after or on the

$$FACPR = \frac{DIVAMT}{P(t)}$$

DISTS(EXDT,I) for the split, and *t'* is a date before the split.
In other less common distribution events, spin-offs and rights, RDIST(FACPR,I) is not equal to RDISTS(FACSHR,I). RDISTS(FACPR,I) is defined as the RDISTS(DIVAMT,I) divided by the stock price on the DISTS(EXDT,I), *(P(t))*.

If there is no available price on the DISTS(EXDT,I), and there is a price within ten days after *(P(t))*, CRSP substitutes that price for *(P(t))*.

Since only month-end prices are available on the monthly stock files, it is possible to use the FACPR and the dividend amount to compute the approximate price of the underlying security on the ex-distribution date. Also for the monthly file, distributions with FACPR's are in effect reinvested on the ex-distribution date and not at the end of the month.

Other cases where factor to adjust price may not be equal to factor to adjust shares are issuances and offers. For issuances, RDISTS(FACPR,I) is set to zero. For offers, RDISTS(FACPR,I) is set to the percent of shares accepted multiplied by negative one.

FACSHR **Factor to Adjust Shares Outstanding** **REAL**

RDISTS(FACSHR,I) is an adjustment for shares outstanding. See the definition of shares outstanding, for more information on the use of FACSHR to derive share numbers. RDISTS(FACSHR,I) equals RDISTS(FACPR,I) for most distribution events. There are three types of distributions where this is the case. See FACPR for these cases and how they are handled.

For spin-offs RDISTS(FACSHR,I) is set to zero. For rights issues, RDISTS(FACSHR,I) is computed as the reciprocal of the subscription ratio. For issuances and offers, if RDISTS(FACSHR,I) is nonzero, the factor to adjust shares is calculated the same as for stock splits.

DCLRDT **Declaration Date** **INTEGER**

DISTS(DCLRDT,I) is the date on which the board of directors declares a distribution. If a declaration date cannot be found, DISTS(DCLRDT,I) is set to zero. This date is coded as YYMMDD. For example, March 28, 1974 is represented as 740328.

EXDT **Ex-Distribution Date** **INTEGER**

DISTS(EXDT,I) is the ex-dividend or ex-distribution date, the date on which the security is first traded without the right to receive the distribution. This date is coded as YYMMDD.

For a merger, exchange or total liquidation where the company disappeared, DISTS(EXDT,I) is, by convention, set equal to the trading day immediately after the date of the last price (DELIST(DLSTDT,1)):

DISTS(EXDT,I)=CALDT(ENDPRC+1)

Ex-dates of liquidating payments after delistings are reported when available, and set to record or pay date of payments if unavailable.

RCRDDT **Record Date** **INTEGER**

RDISTS(RCRDDT,I) is the record date, on which the stockholder must be registered as holder of record on the stock transfer records of the company in order to receive a particular distribution directly from the company. This date is coded as YYMMDD.

For a merger, exchange or total liquidation where the company disappeared, RDISTS(RCRDDT,I) is, by convention, set equal to the date of the last price DISTS(DLSTDT,I)

RCRDDT, ≡ PAYDT
= DLSTDT
= CALDT(ENDPRC) if a daily file

Record dates of liquidating payments after delisting are reported when available and set to 0 when unavailable.

PAYDT **Payment Date** **INTEGER**

DISTS(PAYDT,I) is the date upon which dividend checks are mailed or other distributions are made. For a merger, exchange or total liquidation where the company disappeared PAYDT is, by convention, set equal to the date of the last price DISTS(DLSTDT,I)

PAYDT ≡ RCRDDT
= DLSTDT
= CALDT(ENDPRC) if a daily file

For rights offerings the PAYDT is set equal to the record date, RCRDDT, found in "Moody's Dividend Record" by convention.

Payment dates of liquidating payments after delisting are reported when available and set to 0 when unavailable.

ACPERM **Acquiring PERMNO** **INTEGER**

DISTS(ACPERM,I) is the PERMNO of another security linked to the distribution. DISTS(ACPERM,I) is the PERMNO of stock received in a spinoff, exchange, merger, or other distribution event. It can also link to a security that was acquired in a merger causing a shares increase.

DISTS(ACPERM,I) is zero if not applicable or unknown. If multiple distributions exist with the code on the same EXDT, ACPERM is numbered from 0 to distinguish them. No ACPERMs are available before 1985.

ACCOMP **Acquiring PERMCO** **INTEGER**

DISTS(ACCOMP,I) is the PERMCO of another company linked to the distribution. If ACPERM is nonzero, ACCOMP is the PERMCO of that security. If ACPERM is zero, ACCOMP can still be set. In this case, ACCOMP represents cash received by shareholders of this security paid by the company indicated by ACCOMP.

DISTS(ACCOMP,I) is zero if not applicable or unknown. No ACCOMPs are available before 1985.

SHARES(,) **INTEGER**

The Shares structures array consists of the number of shares outstanding, a statement date and a share flag identifier. There are two types of shares structures. Primary shares structures contain a shares outstanding value from an annual or quarterly report. These are supplemented by shares structures containing values derived from distributions affecting shares outstanding using FACSHR. A new entry does not imply that there was a change in the number of shares outstanding. In general, every company has at least one shares structure per year.

SHROUT **Number of Shares Outstanding** **INTEGER**

SHARES(SHROUT,I) is the number of publicly held shares, recorded in thousands.

SHRSDT **Date of Shares Structure** **INTEGER**

SHARES(SHRSDT,I) is a specific date corresponding to a shares outstanding value. The shares date is either the statement date from a firm's annual or quarterly report, or the ex-distribution date (EXDT) of the distribution affecting the shares.

SHRFLG **Share Flag** **INTEGER**

SHARES(SHRFLG,I) is an integer value indicating the sources of the I^{th} shares structure. A value of zero indicates a share structure extracted from our data sources. The value of one will correspond to a share structure derived from a distribution.

DELIST(,) **INTEGER**
RDELIS(,) **REAL**

Delisting structures contain a delisting code, delisting date, and information about the next observation after delisting, if available. In this version of the file, the delisting array only contains the most recent delisting structure for each company.

Integer values should be accessed using `DELIST(*,*)`, while real values should be accessed using `RDELIS(*,*)`.

CRSP has developed guidelines for closing an issue to further delisting research. These guidelines apply to all types of delistings. An issue is considered to be closed to further research if any of the following conditions apply:

- Research has verified that a final distribution has been paid to stockholders.
- Research has verified that no distributions are ever to be paid to stockholders.
- Some distributions have been paid to stockholders, but no final distribution information can be found and 10 years have passed since the date of the most recent delisting information.
- No information concerning the delisting can be found and 10 years have passed since the delist date.

If none of these conditions apply to a delisted issue, the issue is considered to be pending, which means that further research is required until one of the above conditions has been met.

DLSTDT **Delisting Date** **INTEGER**

`DELIST(DLSTDT,1)` contains the date (in YYMMDD format) of a security's last price on the current exchange. `CALDT(ENDPRC)` equals `DELIST(DLSTDT,1)` for all securities, even if the security is still active.

DLSTCD **Delisting Code** **INTEGER**

`DELIST(DLSTCD,1)` is a three-digit code. It either `(1)` indicates that a security is still trading or `(2)` provides a specific reason for delisting. All coded delistings belong by the first digit of the delisting code.

Delisting Codes

Code	Category
100	still trading or halted but not yet delisted
200	merger
300	exchange
400	liquidation
500	delisted by NYSE/AMEX or Nasdaq
700	delisted by the Securities and Exchange Commission
800	trading simultaneously on more than one exchange

The second and third digits of the delisting codes provide further details of delisting events. Additional delisting codes, specific to various delisting categories, have been created to indicate if an issue is closed to further research, or if the issue is pending further research. (See Appendix A.3 for a complete list of delisting codes.)

NWPERM **New CRSP Permanent Number** **INTEGER**

`DELIST(NWPERM,1)` is assigned when an issue ceases trading as a result of a merger or exchange where shareholders receive stock in the acquiring company. The `NWPERM` is

the permno of the acquiring company's stock. It acts as a forward pointer, allowing the user to trace the ongoing history of surviving companies. DELIST(NWPERM,1) may identify an issue that exists on a different CRSP stock file.

NWCOMP **New CRSP Permanent Company Number** **INTEGER**

DELIST(NWCOMP,1) is assigned when an issue ceases trading as a result of a merger or exchange when shareholders receive some payment from the acquiring company. If DELIST(NWPERM,1) is nonzero, DELIST(NWCOMP,1) is the PERMCO of that security. If DELIST(NWPERM,1) is zero, DELIST(NWCOMP,1) can still be nonzero if payment from the acquiring company not including stock is valued at more than fifty percent of the delisting security's value.

NEXTDT **Date of Next Available Information** **INTEGER**

DELIST(NEXTDT,1) is the date (in YYMMDD format) of a security's delisting price, RDELIS(DLPRC,1). It is set to zero if the security is still active, if CRSP determines the final value of the security by a distribution (e.g. merger terms), or if the value of the security is unknown after suspension or delisting. If a liquidation was announced in advance and trading continued on the exchange after the announcement of the liquidation, DELIST(NEXTDT,1) is set to DELIST(DLSTDT,1). Otherwise it is set to the YYMMDD date of a price found off the exchange.

If our research determines that a security became worthless at any time after delisting and there is no evidence of any trading of the security in the interim, then DELIST(NEXTDT,1) is set to one trading day after the delist date and the delisting price is set to zero.

DLAMT **Amount After Delisting** **REAL**

RDELIS(DLAMT,1) is the value of the issue used to calculate the ending value in the delisting return. It is either the price on another exchange after delisting or the sum of merger or exchange payments. On a monthly file if no value after delisting exists and daily prices exist after the last month-end trading date, RDELIS(DLAMT,1) is set to the last daily price.

DLRETX **Delisting Return Without Dividends** **REAL**

RDELIS(DLRETX,1) is the return of the security after it is delisted. Ordinary dividends that exist after the last trading date and the date of delist payment are excluded. See DLRET for calculation and missing values.

DLPRC **Delisting Price** **REAL**

RDELIS(DLPRC,1) is the price on the date specified in DELIST(NEXTDT,1). If DELIST(NEXTDT,1) is zero, RDELIS(DLPRC,1) is set to zero, it usually refers to a price quote on a new exchange after an exchange change. If RDELIS(DLPRC,1) is positive, then it is a trade price. If RDELIS(DLPRC,1) is negative, then it represents the average of bid and ask quotations.

DLPDT **Date of Delisting Payment** **INTEGER**

DELIST(DLPDT,1) is the YYMMDD effective date of any amount after delisting used in delisting return payments. It is set to 0 if no payments exist.

DLRET **Delisting Return** **REAL**

RDELIS(DLRET,1) is the return of the security after it is delisted. It is calculated by comparing a value after delisting against the price on the security's last trading date. The value after delisting can include a delisting price or the amount from a final distribution. If DELIST(NEXTDT,1) is nonzero, it can be used as the effective date of RDELIS(DLRET,1). Otherwise, by convention, CALDT(ENDDAT+1) is the effective date of the delisting return.

RDELIS(DLRET,1) is only set to a non-missing value, if the information for a value after delisting, used in calculating the return pertains to the period for which the return is being calculated. There are four possible missing return values for this field.

Missing Return Codes

DLRET	Reason For Missing Return
-55.0	CRSP has no sources to establish a value after delisting or is unable to assign a value to one or more known distributions after delisting.
-66.0	more than 10 trading periods between a security's last price and its first available price on a new exchange.
-88.0	security is still active.
-99.0	security trades on a new exchange after delisting, but CRSP currently has no sources to gather price information.

For any issue that has been closed to further research, a delisting return is calculated as follows:

1. If a final distribution has been verified, the delisting return is calculated using all known distribution information.

2. If distributions are available, but no final distribution is known, the delisting return is calculated using all known distribution information.

3. If there is evidence that no distributions will ever be paid to stockholders, the stock is considered worthless and the delisting return is set to -1 (100% loss).

For any issue that is pending further research, delisting codes 470 and 480 only, the missing return code is set to -55.

NASDIN(,) **INTEGER**

The Nasdaq information structure array describes an issue's trading status. This information is supplied by the National Association of Securities Dealers, Inc. (NASD). Nasdaq information structures are available for securities trading on Nasdaq beginning on April 1, 1982 for traits date and The Nasdaq National Market indicator. All fields are available beginning November 1, 1982[6]. If the traits code, The Nasdaq National Market indicator, the NASD index, or the number of market makers changes, then a new structure is added, and the date of the change is recorded in TRTSDT.

The Nasdaq National Market[SM] was initiated in April 1982 for larger and generally more actively traded Nasdaq securities. The Nasdaq National Market Securities must meet higher financial and non-financial criteria than other Nasdaq stocks, and are subject to last-sale reporting. In June of 1992 the regular Nasdaq segment of The Nasdaq Stock Market[SM] was renamed The Nasdaq Small Cap Market and for the first time these became subject to real-time price and volume reporting.

TRTSDT **Traits Date** **INTEGER**

NASDIN(TRTSDT,I) is the effective begin date for the traits structure. It is stored as YYMMDD.

TRTSCD **Traits Code** **INTEGER**

NASDIN(TRTSCD,I) is a one-digit code describing the trading status of the issue.

Nasdaq Trading Status Code

TRTSCD	Description
0	unknown
1	active
2	trading with only one market maker
3	suspended
4	inactive
5	delisted

NMSIND **Nasdaq National Market Indicator** **INTEGER**

NASDIN(NMSIND,I) is a one-digit code indicating whether or not the issue is a member of The Nasdaq National Market (formerly NMS). Prior to June 15, 1992, transaction data was not available for The Nasdaq Small Cap Market Securities. As of June 15, 1992 transaction data became available and bid and ask data are found on the Nasdaq File (SN).

Nasdaq National Market Indicators

NMSIND	Description
0	unknown
1	Nasdaq Small Cap before June 15, 1992
2	Nasdaq National Market
3	Nasdaq Small Cap after June 15, 1992

6 Nasdaq information data are missing in December, 1982 for all issues with NASD company numbers less than 1025 (approximately 20 percent of the securities active at that time), and are missing in February, 1986 for all issues. TRTSDT, TRTSCD, and NMSIND are complete. All other fields are missing.

MMCNT **Market Maker Count** **INTEGER**

NASDIN(MMCNT,I) is the number of registered market makers for the issue. This will return a 0 if there are no registered market markers for that time, and if the date falls in December of 1982 for a NASD Company Number less than 1025 or in February of 1986, the value is missing in the file.

NSDINX **NASD Index** **INTEGER**

NASDIN(NSDINX,I) is a one-digit code indicating the issue's classification within NASD's internal business description categories.

NASD Internal Business Description Categories

NSDINX	Description
0	unknown
1	no index
2	industrial company
3	bank
4	other financial institution
5	insurance company
6	transportation company
7	utility company

/DDATA/ — Price, Volume, and Return Arrays

The core of the CRSP stock file is the periodic data arrays. Three price, one volume and one return array are provided for each issue for each day or month it traded on the exchange.

If the security of a company is included in the Composite Pricing Network, the closing price on a NYSE/AMEX CRSP stock file for CALDT(I) is the last trading price for that day on the exchange that the security traded last. Similarly, highs, lows, and volumes include trades on all exchanges on which that security traded. For example, if a stock trades on both the NYSE and the PACX (Pacific Stock Exchange), and the last trade occurs on the PACX, the closing price on day I will represent the closing price on the PACX, not the NYSE. Price data for Nasdaq securities comes directly from the NASD with the close of the day being at 4:00 Eastern Time.

BIDLO() **Bid or Low** **REAL**

For the monthly files, BIDLO(I) is the lowest closing price or bid/ask average during month I. For the daily files, if PRC(I) is positive then BIDLO(I) is the lowest sale on day I. If PRC(I) is negative, then BIDLO(I) is the closing bid on day I. BIDLO(I) is set to zero if no bid or low price is available on day I.

Daily closing bid prices for The Nasdaq National Market securities can be found in the Nasdaq File starting December 1, 1982. Daily closing bid prices for The Nasdaq Small Cap Market can be found in the Nasdaq File starting June 15, 1992.

ASKHI() **Ask or High** **REAL**

For the monthly files, ASKHI(I) is the highest closing price or bid/ask average during month I. For the daily files, if PRC(I) is positive then ASKHI(I) is the highest sale on day I. If PRC(I) is negative, then ASKHI(I) is the closing ask on day I. ASKHI is set to zero if no ask or high price is available on day I.

Daily closing ask prices for The Nasdaq National Market securities can be found in the Nasdaq File starting December 1, 1982. Daily closing ask prices for The Nasdaq Small Cap Market can be found in the Nasdaq File starting June 15, 1992.

PRC() **Closing Price or Bid/Ask Average** **REAL**

PRC(I) is the closing price or the negative bid/ask average on the date CALDT(I). If the closing price is not available on any given trading day, the number in the price field has a negative sign to indicate that it is a bid/ask average and not an actual closing price on trading date CALDT(I). Please note that in this field the negative sign is a symbol and that the value of the bid/ask average is not negative. If neither closing price nor bid/ask average is available on day I, PRC(I) is set to zero.

VOL() **Share Volume** **INTEGER**

In monthly files, VOL() is the sum of the trading volumes during that month. In daily files, VOL() is the total number of shares of a stock sold on day I. Volume is expressed in units of one share, although our data source for NYSE/AMEX reports the number rounded to the nearest hundred. For example, 12,345 shares traded will be reported on the Nasdaq file as 12,345 and on the NYSE/AMEX file as 12,300. Volume is set to -99 if the value is missing. A volume of zero usually indicates that there were no trades on that day.

NYSE/AMEX volumes are the sum of volumes on all exchanges where that security traded that day.

RET( ) **Holding Period Return** **REAL**

A return is the change in the total value of an investment in a common stock over some period of time per dollar of initial investment. `RET(I)` is the return for a sale on day `I`. It is based on a purchase on the most recent time previous to `I` when the security had a valid price. Usually, this time is `I - 1`.

Returns are calculated as follows:

For time *t* (a holding period), let:
t' = time of last available price < *t*
r(t) = return on purchase at *t'*, sale at *t*
p(t) = last sale price or closing bid/ask average at time *t*
d(t) = cash adjustment for *t*
f(t) = price adjustment factor for *t*
p(t') = last sale price or closing bid/ask average at time of last available price < *t*.

then

$$r(t) = \frac{p(t)\, f(t) + d(t)}{p(t')} - 1$$

t' is usually one period before *t*, but *t'* can be up to ten periods before *t* if there are no valid prices[2] in the interval.

A series of special return codes specify the reason a return is missing.

Missing Return Codes

RET(*t*)	Reason For Missing Return
-66.0	more than 10 periods between time *t* and the time of the preceding price *t'*
-77.0	not trading on the current exchange at time *t*
-88.0	no return, array index *t* not within range of `BEGRET` and `ENDRET`
-99.0	missing return due to missing price at time *t*

/ADATA/ —Returns without Dividends in Portfolio Data, Secondary Monthly Arrays

/ADATA/ contains daily returns without dividends, or monthly secondary prices, and yearly portfolio data. The availability of these data vary from product to product; check the variable descriptions and the data availability chart for more information.

Excess Returns

There are two types of excess returns: beta excess returns and standard deviation excess returns. They are based on beta and standard deviation, respectively.

For each stock in the CRSP NYSE/AMEX Daily File, beta and standard deviation are computed for each year that the stock satisfies the relevant selection criteria. A beta is computed for a given year only if the stock traded at least half of the trading days in that year. A standard deviation is computed for a given year only if there are returns for at least 80% of the trading days in that year.

To calculate the beta excess return, the market is divided into ten risk classes, or portfolios, based on beta. Returns for each of the ten portfolios are calculated; returns for each of the individual securities are calculated as well. The beta excess return for a given security is the difference between its daily return and the daily return of the portfolio to which it has been assigned.

The calculation of standard deviation excess returns is identical to the calculation of beta excess returns, except the preliminary division into portfolios is based on standard deviation rather than beta.

The three steps in the calculation of excess returns are described in greater detail below. In the following discussion, "statistic" always refers to beta or to standard deviation, depending on the history.

1. Betas are calculated using methods developed by Scholes and Williams (Myron Sholes and Joseph Williams, "Estimating Betas from Nonsychronous Data", Journal of Financial Economics, vol. 5, 1977, 309-327). The returns input to the beta calculation are based on trade data only. Trade data includes only valid closing prices, not bid/ask averages.

   The Nasdaq File does not use the standard Scholes-Williams trade data only restriction, since most Nasdaq securities were not required to report transactions. Removing bid/ask averages would restrict Nasdaq data to only Nasdaq National market securities after 1982 and Nasdaq Small-Cap Market securities after June 15, 1992.

   Beta is calculated each year as follows:

   $reti_{,t}$=log of (1 + return for security *I* on day *t*)
   $mret_t$=log of (1 + value-weighted market return on day *t*)
   $mret_i = mret_{t-1} + mret_t + mret_{t+1}$ (a 3 day moving average market window)
   n = number of observations for the year

$$b_1 = \frac{\sum_t (ret_{i,t} mret3_t) - \left(\frac{1}{n}\right)\left(\sum_t ret_{i,t}\right)\left(\sum_t mret3_t\right)}{\sum_t (mret_t mret3_t) - \left(\frac{1}{n}\right)\left(\sum_t mret_t\right)\left(\sum_t mret3_t\right)}$$

   where summations over *t* are over all days on which security `I` traded, beginning with the second trading day of the year and ending with the second to last trading day of the year.

The regular returns history is used for the standard deviation calculation. For the days the stock did not trade, the returns are computed using prices set to (bid + ask)/2. The formula used in calculating the standard deviation is:

$ret_{i,t}$ =daily raw return (trade or average of bid and ask) of security i on day t.
n =number of observations for the year (of $ret_{i,t}$)
s ≡yearly standard deviation for the i^{th} company

$$s_i = \frac{\sqrt{\sum_t (ret_{i,t})^2 - \frac{1}{n}(\sum_t ret_{i,t})^2}}{n-1}$$

where summation over t is over all returns for the i^{th} company in the given calendar year.

In both series, multiple day returns are excluded from the calculation. Inclusion in the file depends on the number of days traded out of the number of trading days in a year. No adjustment is made for the first year a stock appears on the CRSP Daily NYSE/AMEX File. thus, if a stock begins on the file after mid-year, both excess returns series for the first year will be missing. The excess returns histories can be missing for any year even if the stock was trading for some or all of that year.

2. The next step in calculating the excess returns is the actual ranking of the stocks and the forma tion of the ten aggregate portfolios. The same logic is used for both series on the file.

   If a statistic is available for the preceding year, this value is used. If not, the current year's statistic is used, if available. If neither the preceding year's statistic nor the current year's statistic is available, the stock will not be assigned to a portfolio (i.e. portfolio number =0) for the given year.

   For each year, each stock with a relevant statistic (i.e. not in portfolio number 0) is assigned to one of the ten aggregate portfolios. Portfolio number 1 contains the stocks with the highest statistic values in that year.

   In forming the ten aggregate portfolios, only single day returns are used. For the NYSE/AMEX beta series, prices set as bid/ask averages are treated as missing. The aggregate portfolio returns are equally weighted.

   The formula used in calculating the portfolio return on day t *(prett)* is:

   m = number of securities in the portfolio where $ret_{i,t}$ is not missing.

$$pret_t = \frac{1}{m}\sum_{i=1}^{m} ret_{i,t}$$

3. Finally, the excess returns are computed. The regular CRSP daily returns are used as input, except for the NYSE/AMEX beta excess returns where trade-only returns are used. If a stock is in one of the ten portfolios for a given year, the excess returns for the stock for that year are computed by subtracting the appropriate portfolio return from the stock's return. Multiple day excess returns are computed where necessary by subtracting a compounding of the portfolio return form the multiple day stock return. If the excess returns cannot be computed for a given year (i.e. the stock appears in portfolio zero for that year), a large negative number is used as a missing value, as with raw returns.

Monthly Secondary Prices and Returns

Monthly secondary prices contain bid/ask spreads. Returns without dividends are calculated in a similar manner to regular returns (RET), except ordinary dividends are not included.

Yearly Data

Two yearly data arrays are included in /ADATA/, YRVAL and PRTNUM. YRVAL contains yearly values of capitalizations, betas, or standard deviations for the securities. PRTNUM contains decile portfolio rankings for capitalization based on a NYSE/AMEX/Nasdaq universe, a NYSE/AMEX universe or a Nasdaq universe, or beta or standard deviations.

BXRET() **Beta Excess Return** **REAL**

BXRET() denotes the excess return of a specific issue less the average return of all issues in its beta portfolio on day I. It is important to note that the BXRET() is computed by using trade-only returns data (i.e. excluding returns on bid-ask averages). A missing beta excess return due to a portfolio assignment of zero is set to –44.0. This variable is only available to CRSPAccess97 Stock File users who also have the CRSPAcces97 Indices File / Portfolio Assignments Product and cannot be accessed simultaneously with PRC2.

SXRET() **Standard Deviation Excess Return** **REAL**

SXRET(I) denotes the excess return of a specific issue less the average return of all issues in its standard deviation portfolio on day I. SXRET() is computed by using the returns on bid-ask averages as well as closing prices. A missing standard deviation excess return due to a portfolio assignment of zero is set to –44.0. This variable is available only available to CRSPAccess97 Stock File users who also have the CRSPAcces97 Indices File / Portfolio Assignments Product and cannot be accessed simultaneously with RETX.

RETX() **Return Without Dividends** **REAL**

RETX() contains returns without dividends. Ordinary dividends and certain other regularly taxable dividends are excluded from the returns calculation. The formula is the same as for RET except *d(t)* is usually 0.

PRC2() **Secondary Price** **REAL**

PRC2() is the secondary price information. It concerns primarily the spread of the stock. If PRC(I) is positive, PRC2(I) is always zero. If PRC(I) is negative and PRC2(I) is nonzero, PRC2 represents the bid/ask spread. If PRC(I) is zero and PRC2(I) is negative, PRC2(I) represents a bid price. If PRC(I) is zero and PRC2(I) is positive, PRC2(I) represents an ask price. Secondary price is only available for the monthly files.

PRTNUM(,) **INTEGER**

PRTNUM contains portfolio assignment data for three types of portfolios. The default available portfolios are dependent on the products installed and can be based on capitalizations, betas or standard deviations. Only one capitalization portfolio is available in a stock database unless it is supplemented with the CRSPAccess97 Indices File/Portfolio Assignments product. Default portfolios are described in the following table.

Product	CAP	CAPN or SDEV*	CAPQ or BETA*
DA	NYSE/AMEX/Nasdaq Capitalization Deciles		
DA + IX	NYSE/AMEX/Nasdaq Capitalization Deciles	NYSE/AMEX Capitalization Deciles	Nasdaq Capitalization Deciles
DX	NYSE/AMEX Capitalization Deciles		
DX + IX	NYSE/AMEX Capitalization Deciles	NYSE/AMEX Standard Deviation Deciles	NYSE/AMEX Beta Deciles
MA	NYSE/AMEX/Nasdaq Capitalization Deciles		
MA + IX	NYSE/AMEX/Nasdaq Capitalization Deciles	NYSE/AMEX Capitalization Deciles	Nasdaq Capitalization Deciles
MZ	NYSE/AMEX Capitalization Deciles		
MZ + IX	NYSE/AMEX Capitalization Deciles		

* See the CRSPAccess97 Indices File Guide for the CAPN or SDEV and CAPQ or BETA Indices.

There are three annual PRTNUM variables: CAP, CAPN/SDEV, and CAPQ/BETA. The FORTRAN parameters CAPN and SDEV are both set to 2 and CAPQ and BETA are set to 3. An issue's portfolio ranking is assigned using the previous year's capitalization, beta, or standard deviation value. If the previous year's statistic is missing, the current year's statistic is substituted to assign the portfolio ranking. If both statistics are missing, a portfolio ranking of zero is assigned. Portfolios are assigned for one year in the future for active stocks, although these assignments are subject to change in the next update depending on additions and deletions.

There are three FORTRAN variables available to choose portfolios to be loaded into the PRTNUM and YRVAL arrays. PORTTYPE1 specified the portfolio types loaded into the CAP field. PORTTYPE2 specifies the portfolio types loaded in to the CAPN/SDEV fields. PORTTYPE3 specifies the portfolio types loaded into the CAPQ/BETA fields. See the **CRSPAccess97 Indices File Guide** for available portfolio types in the CRSPAccess97 Indices File / Portfolio Assignments product. See the **CRSPAccess97 Programmers File Guide** for options on loading the different portfolios or the excess returns.

The PRTNUM array contains annual data and does not use the regular CRSP daily calendar. The second dimension is instead based on the year (YY) and is set to that year minus an offset of 24. Securities will have valid PRTNUM data only between the range of header variables BEGYR and ENDYR.

CAP **Year End Capitalization Portfolio Assignment** **INTEGER**

PRTNUM(CAP,I) is the decile portfolio ranking for this issue for year I+24, ranked according to its market capitalization for the previous year-end. The securities with the smallest capitalizations are placed in portfolio 1. The partitions on a CRSP file include all securities, excluding ADRs, that were active on the New York, American and Nasdaq exchanges for that year. If a capitalization was not available for the previous year, the capitalization on the date with the earliest available price in the current year was used for ranking. This ranking is available for all CRSP stock master files.

CAPN / SDEV **NYSE/AMEX Capitalization or Standard Deviation Assignment** **INTEGER**

PRTNUM(CAPN,I) is the decile portfolio ranking for year I+24. CAPN is similar to CAP except assignments are based on breakpoints of the NYSE/AMEX universe. In NYSE/AMEX SDEV portfolios, the issues with the highest standard deviation are in portfolio 1.

CAPQ / BETA **Nasdaq Capitalization or NYSE/AMEX Beta Assignment** **INTEGER**

PRTNUM(CAPQ,I) is the portfolio ranking for year I+24. CAPQ is similar to CAP except assignments are based on breakpoints of the Nasdaq universe. In NYSE/AMEX BETA portfolios, the issues with the highest beta are in portfolio 1.

YRVAL(,) **REAL**

YRVAL contains the individual year-end capitalization, standard deviation, and beta values for specific issues. Only one capitalization statistic is available in a stock database unless it is supplemented with the CRSP Access97 Indices File / Portfolio Assignments Product. Current portfolios are described in the table under the PRTNUM variable. Calculations of beta and standard deviations available in the CRSP Access97 Indices File / Portfolio Assignments Product are described in the **CRSP Access97 Indices File Guide**.

The YRVAL array contains annual data and does not use the regular CRSP daily calendar. The second dimension is instead based on the year (YY) and is set to that year minus an offset of 24. Securities will have valid YRVAL data only between the range of header variables BEGYR and ENDYR.

CAP **Year-End Capitalization** **REAL**

YRVAL(CAP,I) is the capitalization for the issue for year I+24. The capitalization is assigned by using the year-end share figure multiplied by the year-end price. If either no shares outstanding values exist or the year-end price is unavailable, no capitalization is calculated, and the value is set to zero.

CAPN / SDEV **NYSE/AMEX Year-End Capitalization or Annual Standard Deviation** **REAL**

YRVAL(CAPN,I) is the capitalization on NYSE or AMEX exchanges for the security for year I+24.

CAPQ / BETA **Year-End Capitalization on The Nasdaq Stock Market℠ or Annual Beta** **REAL**

YRVAL(CAPQ,I) is the capitalization on Nasdaq for the security for year I+24.

/NMSHDR/ — Identification and Data Range Variables

The Nasdaq header structure contains fields that specify the range of valid supplemental Nasdaq data for an issue. These remaining fields in /NMSHDR/ describe the individual security's range for arrays in /NMSDAT/.

BEGNMS **Beginning of Nasdaq Data** **INTEGER**

BEGNMS is the lowest subscript of a security's daily Nasdaq arrays. CALDT(BEGNMS) is the date that daily Nasdaq data begins for the security. BEGNMS is zero if a security has no supplemental data.

ENDNMS **End of Nasdaq Data** **INTEGER**

ENDNMS is the highest subscript of a security's daily Nasdaq data arrays. CALDT(ENDNMS) is the date that Nasdaq data ends for the security. ENDNMS is zero if a security has no supplemental data.

/NMSDAT/ — Bid, Ask, and Number of Trades Arrays

Closing bid, closing ask and the number of trades are provided for each issue for each day it traded on The Nasdaq National Market. Due to data source limitations data is missing for 15 Nasdaq National Market securities in December, 1982, and for all the Nasdaq National market securities in February, 1986.

Beginning June 15, 1992, all Nasdaq securities including The Nasdaq National Market and The Nasdaq Small Cap Market may have supplemental data including closing bid prices, closing ask prices and number of trades for each issue for each day the security is traded.

NMSBID () **Closing Bid** **REAL**

NMSBID (I) is the closing bid on day I. The close of the day is 4:00 PM Eastern Standard Time. Since July, 1980, Nasdaq has used the inside quotation as the closing bid and ask. The inside quotation is the highest bid and lowest ask.

NMSASK() **Closing Ask** **REAL**

NMSASK (I) is the closing ask on day I.

NMSTRD() **Number of Trades** **REAL**

NMSTRD (I) is the number of trades on the date CALDT (I). A large negative number, -99 is used to indicate that the data are unavailable.

2.3 C Stock Structure and Data Items

This section describes the CRSPAccess97 Stock Structure and gives detailed descriptions of the Stock File C variables. All data is organized by security and organized by a permanent number assigned by CRSP called PERMNO. Secondary identifiers are available in the header and name history that can be used to cross-reference securities to PERMNOs.

A CRSPAccess97 database is called a CRSPDB. A CRSPDB is a group of common files supporting predefined sets of data. There are two variations of stock databases available, both with daily and monthly time series. Both share common data items, although interpretation may be different for time series. A monthly or daily stock database is identified by a database name and a set identifier, or SETID.

SETID	Database Name	Description
10	CRSP_DSTK	Daily CRSPDB
20	CRSP_MSTK	Monthly CRSPDB

All data items in a CRSPDB are organized into data objects for headers, event arrays, time series, and calendars. The following subsections describe the objects and specific data items used to store CRSP Stock and Indices data.

- CRSP Data Objects
- CRSP Stock Data Objects
- CRSP Stock Data Items
- CRSP Index Structure and Data Objects
- CRSP Index Data Items

CRSPAccess97 Data Objects

There are four basic data objects used in CRSPAccess97 databases; `CRSP_ROW`, `CRSP_ARRAY`, `CRSP_TIMESERIES` and `CRSP_CAL`. The objects and their common fields are described below. The components of indices objects are described in the CRSP Index Data Items section. See the CRSPAccess97 Stock Programmers Guide for examples of usage of objects and components, and codes for predefined CRSP types of arrays or sub categories.

`CRSP_ROW` contains a single data structure. The components of the structure are applic able to the key in this file at all times. The fields in the data structure are dependent on the specific data item.

`objtype` **Object Type Code** **`int`**

`objtype` is a code defining the type of object. `objtype` is always 5 for a `CRSP_ROW`.

`arrtype` **Array Type Code** **`int`**

`arrtype` is a code defining the type of data in the object. `arrtype` can define a basic data type or a CRSP-defined structure.

`subtype` **Data Subcategory Type Code** **`int`**

`subtype` is a code further defining categories of data that otherwise have the same structure, such as the difference between a return and index level data item.

`size_of_array_width` **Size of Associated Structure** **`int`**

`size_of_array_width` is the number of characters used in each structure element for this array type.

CRSP_ARRAY contains a list of events, observations, or status changes. There is a count of the number of events. The fields in each event are dependent on the specific data item. The fields usually contain one or more date fields indicating the effective time of the event, plus descriptive fields describing the event.

objtype **Object Type Code** **int**

objtype is a code defining the type of object. objtype is always 3 for a CRSP_ARRAY.

arrtype **Array Type Code** **int**

arrtype is a code defining the type of data in the object. arrtype can define a basic data type or a CRSP-defined structure.

subtype **Data Subcategory Type Code** **int**

subtype is a code further defining categories of data that otherwise have the same structure, such as the difference between a return and index level data item.

size_of_array_width **Size of Associated Structure** **int**

size_of_array_width is the number of characters used in each structure element for this array type.

maxarr **Maximum Number of Events** **int**

maxarr is the maximum number of event structures available in the CRSP_ARRAY.

num **Actual Number of Events** **int**

num is the count of actual event structures available in the CRSP_ARRAY for the current entity.

CRSP_TIMESERIES contains a list of observations synchronized with a specific calendar of time periods. There is a beginning and ending index and a link to a calendar defining the time periods. The observation can contain a simple or multiple-component data structure, dependent on the specific data item.

objtype **Object Type Code** **int**

objtype is a code defining the type of object. objtype is always 2 for a CRSP_TIMESERIES.

arrtype **Array Type Code** **int**

arrtype is a code defining the type of data in the object. arrtype can define a basic data type or a CRSP-defined structure.

subtype **Data Subcategory Type Code** **int**

subtype is a code further defining categories of data that otherwise have the same structure, such as the difference between a return and index level data item.

size_of_array_width **Size of Associated Structure** **int**

size_of_array_width is the number of characters used in each structure element for this array type.

maxarr **Maximum Number of Time Periods** **int**

maxarr is the maximum number of time periods available in the CRSP_TIMESERIES, counting the 0^{th} period.

beg **First Time Period with Valid Data** **int**

beg is the first element in the data array with valid observations. If beg is 0 there is no data for the timeseries.

end **Last Time Period with Valid Data** **int**

end is the last element in the data array with valid observations. If end is 0 there is no data for this timeseries.

cal **Associated Calendar** **CRSP_CAL ***

cal is a pointer to the associated CRSP calendar structure. CRSP_CAL is described below.

CRSP_CAL contains a set of calendar time periods and a count of the number of periods. The calendars have an identifier and can be used by multiple time series. The calendar time periods are marked with an ending date and are synchronized with a time series such that the n th time series observation is associated with the nth calendar period.

objtype **Object Type Code** **int**

objtype is a code defining the type of object. objtype is always 1 for a CRSP_CAL.

calid **Unique Calendar Identifier** **int**

calid is a code assigned by CRSP to trading calendars. The following calendars are available in CRSPAccess97 databases.

calid	Description	Beginning Date
100	Daily Trading Calendar	19620702
101	Month-end Trading Calendar	19251231
300	Annual Trading Calendar	19251231
310	Quarterly Trading Calendar	19251231

ndays **Index of Last Period in Calendar** **int**

ndays is the number of periods in the calendar. Calendar and time series data can be available up to this index in the calendar array.

calname **Calendar Name** **char[80]**

calname is a text description of the calendar.

caldt **Calendar Date** **int***

caldt contains the list of trading dates. Each date represents the last date in a calendar period, in YYYYMMDD (year, month, date) format. The caldt dates begin in element 1 of the array and continue to element ndays. The value in the nth position of a CRSP_TIMESERIES array is effective on the nth caldt of its associated calendar.

CRSP Stock Data Objects

The following table summarizes the primary data items available in a CRSP stock record. See the CRSP Data Objects section for definitions of the object types and the CRSP Stock Data Items section for the elements in the stock structures.

The object name variable includes header object data plus the item array. Actual data is stored in the item array. Sets of objects represent an array of the given object type. The number of sets variable is the number of objects in the array. Each object in a set has its own set of object header variables, for example each portfolio type can have rebalancing based on a different calendar frequency.

Item	Object Type	Object Name	Item Name	Number of Sets
Stock header	CRSP_ROW	header_row	header	
Names array	CRSP_ARRAY	events.names_arr	events.names	
Distribution array	CRSP_ARRAY	events.dists_arr	events.dists	
Shares Observation array	CRSP_ARRAY	events.shares_arr	events.shares	
Delist Array	CRSP_ARRAY	events.delist_arr	events.delist	
Nasdaq Information Array	CRSP_ARRAY	events.nasdin_arr	events.nasdin	
Bidlo time series	CRSP_TIMESERIES	bidlo_ts	bidlo	
Askhi time series	CRSP_TIMESERIES	askhi_ts	askhi	
Price time series	CRSP_TIMESERIES	prc_ts	prc	
Returns time series	CRSP_TIMESERIES	ret_ts	ret	
Volume time series	CRSP_TIMESERIES	vol_ts	vol	
Bid time series	CRSP_TIMESERIES	bid_ts	bid	
Ask time series	CRSP_TIMESERIES	ask_ts	ask	
Returns without dividends time series	CRSP_TIMESERIES	retx_ts	retx	
Spreads time series (monthly only)	CRSP_TIMESERIES	spread_ts	spread	
Number of trades time series (daily) or Alternate price date time series (monthly)	CRSP_TIMESERIES	numtrd_ts	numtrd	
Alternate price time series (monthly only)	CRSP_TIMESERIES	altprc_ts	altprc	
Set of portfolio time series (portfolio types differ daily and monthly)	CRSP_TIMESERIES	port_ts[ ]	port[ ]	porttypes

CRSP Stock Data Items

header — Identification And Summary Data CRSP_STK_HEADER

The header structure contains fields that identify an issue and summarize a classification of the issue. There is no time component to the header data. The identifying fields in this structure are the unique issue and company identifiers assigned by CRSP, CUSIP, or an exchange. Summary fields include the last classification or status of the issue.

hcusip **Header CUSIP Identifier** **char[16]**

hcusip is the latest eight-character CUSIP identifier for the security through the end of the file. CUSIP identifiers are supplied to CRSP by the CUSIP Service Bureau, Standard & Poor's, a division of McGraw-Hill, Inc., American Bankers Associate database, Copyright 1997. See the CUSIP Copyright Information in Appendix A.6.

CUSIP identifiers were first assigned in 1968 as integers and expanded in 1984 to include alphabetic characters. The first six characters (including leading zeroes) identify the issuer, while the last two characters identify the issue. CUSIP issuer identifiers are assigned to maintain an approximately alphabetical sequence. The CUSIP identifier may change for a security if its name or capital structure changes.

No header or historical CUSIPs are reused on our files. For securities no longer in existence or that were never assigned an official cusip identifier, CRSP has assigned a dummy CUSIP identifier for use in this field in accordance with the rules published in the CUSIP Directory. There are two potential dummy hcusips, which are assigned by CRSP. One, ***99*9*, (containing a 9 in the 4th, 5th and 7th character positions) represents a CRSP assigned cusip with a dummy issuer number (the first 6 character positions) and a dummy issue number (the last 2 character positions). The other, ******9*, (containing a 9 in the 7 th character position) represents a CRSP-assigned cusip with a real issuer number but a dummy issue number. For example:

An hcusip such as 12399099 or 12345699 is assigned by CRSP, and an identifier such as 12345610 is assigned by the CUSIP Agency.

Securities actively traded on an international basis, domiciled outside the United States and Canada, will be identified by a cins (CUSIP International Numbering System) number. CINS numbers employ the same Issuer (6 characters)/Issue (2 characters) 8-character identifier system set by the CUSIP Numbering System. It is important to note that the first portion of a CINS code is always represented by an alpha character, signifying the issuer's country code (domicile) or geographic region. See the current CUSIP Directory for more information. See Appendix A.1 for a list of CINS country codes.

permno **CRSP Permanent Number** **int**

permno is a unique five-digit permanent identifier assigned by CRSP to each security in the file. Unlike cusip, ticker, and comnam, the permno neither changes during an issue's trading history, nor is reassigned after an issue ceases trading. The user may track a security through its entire trading history in CRSP's files with one permno, regardless of name changes or capital structure changes. A security that is included on both CRSP's Nasdaq and NYSE/AMEX files will have the same CRSP permanent number in both files. The data is sorted by this field.

permco **CRSP Permanent Company Number** **int**

permco is a unique permanent identifier assigned by CRSP to all companies with issues on a CRSP file. This number is permanent for all securities issued by this company regardless of name changes.

compno **Nasdaq Company Number** **int**

compno is a unique integer assigned by the National Association of Securities Dealers (NASD) to each company with a listed security on The Nasdaq Stock Market[SM]. If the company never traded an issue on The Nasdaq Stock Market[SM], or is unknown, compno is set to 0. The compno may change if Nasdaq assigns a new number to an issue that CRSP considers the same company as a company that previously traded on the Nasdaq Stock Market[SM].

issuno **Nasdaq Issue Number** **int**

issuno is a unique integer assigned by the National Association of Securities Dealers (NASD) to each listed security on The Nasdaq Stock Market[SM]. It is this issue-specific identifier which differentiates securities issued by the same company. If the issue number is unknown, ISSUNO is set to zero. If an NYSE/AMEX security was ever traded on Nasdaq, this number is set to the latest issue number assigned when it was trading on Nasdaq. The issuno in the CRSP file may change if Nasdaq assigns a new number to an issue CRSP considers to be a continuation of an existing issue.

hexcd **Header Exchange Code** **int**

hexcd displays the latest exchange code listed for a specific security. Valid hexcd's are 1, 2, or 3, which correspond to the NYSE, AMEX, and Nasdaq respectively. Other exchange codes are not included in hexcd.

hshrcd **Header Share Code** **int**

hshrcd is the last CRSP share code in a specific security's name history. See the names element shrcd for the description of the CRSP share type coding scheme.

hsiccd **Header Standard Industrial Classification (SIC) Code** **int**

hsiccd is the last non-zero Standard Industrial Classification (SIC) Code found in a specific security's name structure. hsiccd will be zero for companies where CRSP's data source did not provide SIC Codes.

begdt **Beginning of Data** **int**

begdt is the date that data begins for the security, in YYYYMMDD format. It is the date of the first period in the time series arrays. Begdt is always greater than zero.

enddt **End of Data** **int**

enddt is the date that data ends for the security, in YYYYMMDD format. It is the date of the last period in the time series arrays. Enddt is always greater than zero.

dlstcd **Delisting Code** **int**

dlstcd is the issue's delisting status at the end of the file. See the delist codes described in the delist array for possible codes.

htick **Header Ticker Code** **char[16]**

htick is the ticker on the current exchange. It is only set if the company is active at the end of the file and is blank otherwise. If the security trades on NYSE or AMEX and is distinguished with a non-blank share class, htick includes the primary ticker followed by a period and the share class. Otherwise it is set to the primary ticker. Tickers and share classes are derived from the most recent information in the name history.

Header tickers can be from one to five characters long.

hcomnam **Header Company Name** **char[64]**

hcomnam is the last company name found in the specific security's name history. Company names can be up to 32 characters long.

event — Name, Distribution, Share, Delisting, and Nasdaq Information Arrays

CRSP_STKEVENTS_STRUCT

The event structure is a group of five arrays containing status, event, and observation data for securities. Events include name histories, distribution event histories, share observation histories, delisting information, and the Nasdaq trait histories. The event arrays contain the complete history of all the observations of the pertinent events (names, shares, etc.) for that security.

names **Name History** **CRSP_STK_NAMES ***

The name history includes sets of description fields effective at different times during the history of the security. Each set of information, or name structure, contains name and classification fields and the effective date ranges of those fields. Each security has at least one name structure. If the CUSIP, name, exchange, ticker symbol, or share class changes during the security's trading history, a new name structure is added, with the date of the change (namedt).

Name histories may include periods, possibly outside the data range, when the security is trading on a different exchange or is not trading at all. The exchange code (exchcd) contains more detailed information on trading status and location for a given date range.

namedt **Name Date** **int**

namedt is the starting date of security in the CRSP file, stored in YYYYMMDD format. If the CUSIP, name, exchange or share class changes, CRSP adds a new name structure that records the change and the date the change became effective. namedt is the date associated with a specific name structure.

nameendt **Last Date of Name** **int**

nameenddt is the last effective date of the name structure. It is set to the date preceding the namedt of the next name structure, or the delist date if the last name structure.

The name information on any given date can be found by finding the name structure where the target date is between namedt and nameenddt. There will be exactly one such name structure for any date between the first namedt and the delisting date.

ncusip **Name CUSIP** **char[16]**

ncusip refers to the CUSIP identifier valid during the date range of the name structure. All nonblank ncusips are 8 characters long.

The CUSIP Agency will often change an issue's CUSIP identifier to reflect a change of name or capital structure. CRSP has preserved all CUSIPs that have been assigned to a given issue. ncusip might be a blank if the name structure predates the CUSIP Bureau.

For more details of the CUSIP identifier see the header CUSIP (hcusip) variable description.

ticker **Exchange Ticker Symbol** **char[16]**

ticker is an alphabetic symbol assigned to each security by an exchange. Tickers can be reused over time. The combination of ticker, share class, exchange, and date uniquely identifies a security. A ticker may be one to three characters for NYSE and AMEX securities or four and five characters for Nasdaq securities.

Nasdaq trading tickers have four base characters and a fifth character suffix that provides information about an issue's type or temporary information about an issue's status. CRSP only includes the suffix when it provides permanent descriptive information. This table describes the suffixes appearing on the CRSP file.

Nasdaq 5th Character Suffixes

Suffix	Definition
A	Class A
B	Class B
F	Companies incorporated outside the US
S	Shares of Beneficial Interest
U	Unit
V	When-issued
Y	ADR
Z	Miscellaneous common issues

Occasionally Nasdaq will add two additional suffixes to the base ticker to identify certain issues. However, because the Nasdaq ticker field only allows for five characters, one letter of the base ticker will be dropped. For example:

If a foreign company with a class A stock has a base ticker symbol ABCD, Nasdaq adds two additional characters, A and F. Due to Nasdaq's limited fields, they will delete a letter from the base ticker, so ABCDAF would be truncated to ABCAF.

There is no guarantee that the ticker suffix matches a share type. The shrcd variable should be used to determine the security's share type.

Nasdaq tickers in an issue's name history are presumed to represent legitimate trading symbols for that issue at some point in time, although these symbols may be listed out of proper chronological sequence. In addition, the Nasdaq file ticker symbols provided do not necessarily comprise a definitive list of all symbols used throughout an issue's trading history. Due to source limitations, the ticker field may be blank in name histories of Nasdaq securities that stopped trading from the early 1970's through the early 1980's. NYSE tickers prior to July 1962 are blank.

comnam **Company Name** **char[64]**

comnam is the name of the company as of the date specified in namedt. CRSP allocates a 32 character name description field for all securities. Preference is given to the spellings and abbreviations provided in Standard & Poor's CUSIP Directory. In cases where all name sources provide descriptions in excess of 32 characters, CRSP furnishes its own abbreviations.

shrcls **Share Class** **char[4]**

shrcls describes the class of share and is generally blank. Any letter that identifies the class of stock (e.g., "A" for class A common) is contained in the first position of this field.

shrcd **Share Code** **int**

shrcd is a two-digit code describing the type of shares traded. The first digit describes the type of security traded.

First Digit - Share Code - Security Type

Code	Definition
1	Ordinary Common Shares
2	Certificates
3	ADRs (American Depository Receipts)
4	SBIs (Shares of Beneficial Interest)
7	Units (Depository Units, Units of Beneficial Interest, Units of Limited Partnership Interest, Depository Receipts, etc.)

Note: "Units" (code 7) does not represent combinations of common stock and anything else, such as warrants.

The second digit gives more detailed information about the type of security traded.

Second Digit - Share Code - Security Type

Code	Definition
0	Securities which have not been further defined.
1	Securities which need not be further defined.
2	Companies incorporated outside the US
3	Americus Trust Components (Primes and Scores).
4	Closed-end funds.
5	Closed-end fund companies incorporated outside the US
8	REIT's (Real Estate Investment Trusts).

For example: A shrcd of 14 would represent ordinary common shares of a closed-end fund.

exchcd **Exchange Code** **int**

exchcd is a code indicating the exchange on which a security is listed. Normal exchange codes are respectively 1,2, and 3 for NYSE, AMEX and The Nasdaq Stock Market[SM]. An exchange code of zero indicates that a security is either trading on an unknown exchange, or is temporarily not trading at all. Adding 30 to the normal exchange codes (31, 32 & 33) delineates when-issued trading, such as during a reorganization. See Appendix A.1 for additional Exchange Code information.

Exchange Codes

Code	Definition
-2	suspended by the NYSE or AMEX
-1	halted by the NYSE or AMEX
0	not listed on exchange of current file
1	New York Stock Exchange
2	American Stock Exchange
3	The Nasdaq Stock Market[SM]
31	When-issued trading on the New York Stock Exchange
32	When-issued trading on the American Stock Exchange
33	When-issued trading on The Nasdaq Stock Market[SM]

`siccd` **Standard Industrial Classification (SIC) Code** `int`

`siccd` is the Standard Industrial Classification code. SIC Code is used to group companies with similar products or services.

The Standard Industrial Classification Manual contains descriptions of categories recognized by the US Government. SIC Code is an integer between 100 and 9999. The first two digits refer to a major group. The first three digits refer to an industry group. All four digits indicate an industry. Missing SIC Codes are set to 0. Most Nasdaq SIC codes have only the first three digits; CRSP has added a fourth digit of zero.

dists **Distribution Events Array** **CRSP_STK_DIST ***

The distribution structure array has entries describing cash dividends, capital adjustments, and other distributions to shareholders. Each distribution structure includes:

- The type of distribution
- The amount of distribution
- A factor to adjust price
- A factor to adjust shares outstanding, and
- The dates associated with the distribution (i.e., declare, ex-dividend, record, and payment).
- Links to acquiring securities or companies

If a distribution event has more than one component, CRSP codes each component of the event separately. All components of a distribution event share the same `exdt`. Distributions are sorted by `exdt`, `distcd`, and `acperm` and are unique on these three fields.

distcd **Distribution Code** **int**

CRSP describes distributions with a four-digit code. The first digit describes the type of distribution. The second digit describes the payment method. The third digit augments the type denoted by the first digit. The fourth digit provides information regarding the tax status of the distribution. The coding scheme is described in detail in Appendix A.2. In addition, Appendix A.2 contains a brief definition of some of the `distcd`s found in the CRSP stock files.

CRSP has not verified the tax status of ordinary cash dividends since 1987. Instead, CRSP assigned the most common tax code, taxable as dividend, to ordinary dividends. CRSP does verify the tax status of stock distributions and distributions associated with rights offerings, spin-offs, liquidations, mergers, reorganizations, and exchanges.

divamt **Dividend Cash Amount** **float**

`divamt` is the US dollar value per share of distributions resulting from cash dividends, spin-offs, mergers, exchanges, reorganizations, liquidations, and rights issues.

If the Distribution code is 6225, a nonzero amount represents an offer price given to a certain amount of shares. For these cases, the dollar value per share is actually `divamt` multiplied by the percent of shares accepted by the offer, where the percent of shares accepted can be derived by multiplying `facpr` by negative one. See `facpr`.

facpr **Factor to Adjust Price** **float**

facpr is used to adjust stock prices after a distribution so that a comparison can be made on an equivalent basis between prices before and after the distribution.

facpr equals facshr for most distribution events. There are three types of distributions where this is the case:

1. Cash dividends, facpr is set to zero;
2. Cases of mergers, total liquidations, or exchanges where the security disappeared, facpr is set to negative one by convention;
3. For stock dividends and splits, facpr is the number of additional shares per old share issued:

$$facpr = \frac{s(t) - s(t')}{s(t')} = \frac{s(t)}{s(t')} - 1$$

where *s(t)* is the number of shares outstanding, *t* is a date after or on the exdt for the split, and *t'* is a date before the split.
In other less common distribution events, spin-offs and rights, facpr is not equal to facshr. facpr is defined as the divamt divided by the stock price on the exdt, *(P(t))*.

$$facpr = \frac{DIVAMT}{P(t)}$$

If there is no available price on the exdt, and there is a price within ten days after *(P(t))*, CRSP substitutes that price for *(P(t))*.

Since only month-end prices are available on the monthly stock files, it is possible to use the facpr and the dividend amount to compute the approximate price of the underlying security on the ex-distribution date. Also for the monthly file, distributions with facpr's are in effect reinvested on the ex-distribution date and not at the end of the month.

Other cases where factor to adjust price may not be equal to factor to adjust shares are issuances and offers. For issuances, facpr is set to zero. For offers, facpr is set to the percent of shares accepted multiplied by negative one.

facshr **Factor to Adjust Shares Outstanding** **float**

facshr is an adjustment for shares outstanding. See the definition of shares outstanding, for more information on the use of facshr to derive share numbers. facshr equals facpr for most distribution events. There are three types of distributions where this is the case. See facpr for these cases and how they are handled.

For spin-offs facshr is set to zero. For rights issues, facshr is computed as the reciprocal of the subscription ratio. For issuances and offers, if facshr is nonzero, the factor to adjust shares is calculated the same as for stock splits.

dclrdt **Declaration Date** **int**

dclrdt is the date on which the board of directors declares a distribution. If a declaration date cannot be found, dclrdt is set to zero. This date is coded as YYYYMMDD. For example, March 28, 1974 is represented as 740328.

exdt **Ex-Distribution Date** **int**

exdt is the ex-dividend or ex-distribution date, the date on which the security is first traded without the right to receive the distribution. This date is coded as YYYYMMDD and is always an available daily trading date.

For a merger, exchange or total liquidation where the company disappeared, dists(exdt,i) is, by convention, set equal to the trading day immediately after the date of the last price (delist(dlstdt,1)):

```
dists(exdt,i)=caldt(endprc+1)
```

Ex-dates of liquidating payments after delistings are reported when available, and set to record or pay date of payments if unavailable.

rcrddt **Record Date** **int**

rcrddt is the record date, on which the stockholder must be registered as holder of record on the stock transfer records of the company in order to receive a particular distribution directly from the company. This date is coded as YYYYMMDD and set to 0 if unavailable.

For a merger, exchange or total liquidation where the company disappeared, rcrddt is, by convention, set equal to the date of the last price (dlstdt):

```
rcrddt, ≡ paydt
= dlstdt
= caldt(endprc)   if a daily file
```

Record dates of liquidating payments after delisting are reported when available and set to 0 when unavailable.

paydt **Payment Date** **int**

paydt is the date upon which dividend checks are mailed or other distributions are made. This date is coded as YYYYMMDD and set to 0 if unavailable. For a merger, exchange or total liquidation where the company disappeared paydt is, by convention, set equal to the date of the last price dlstdt:

```
paydt ≡ rcrddt
= dlstdt
= caldt(endprc)  if a daily file
```

For rights offerings the paydt is set equal to the record date, rcrddt, found in "Moody's Dividend Record" by convention.

Payment dates of liquidating payments after delisting are reported when available and set to 0 when unavailable.

acperm **Acquiring permno** **int**

acperm is the permno of another security linked to the distribution. Acperm is the permno of stock received in a spinoff, exchange, merger, or other distribution event. It can also link to a security that was acquired in a merger causing a shares increase.

acperm is zero if not applicable or unknown. If multiple distributions exist with the code on the same exdt, acperms are numbered from 0 to distinguish them. No acperms are available before 1985.

accomp **Acquiring permco** **int**

accomp is the permco of another company linked to the distribution. If acperm is nonzero, accomp is the permco of that security. If acperm is zero, accomp can still be set. In this case, accomp represents cash received by shareholders of this security paid by the company indicated by accomp.

accomp is zero if not applicable or unknown. No accomps are available before 1985.

shares **Shares Observations Array** **CRSP_STK_SHARE ***

The shares observations array consists of observations on security shares outstanding, including the number of shares outstanding, an observation date, and a share flag identifier. There are two types of shares structures. Primary shares structures contain a shares outstanding value derived from an annual or quarterly report. These are supplemented by shares structures containing values derived from distributions affecting shares outstanding using `facshr`. A new entry does not imply that there was a change in the number of shares outstanding. In general, every company has at least one shares structure per year.

Exactly one shares structure is effective each date in the security's history. A shares observation is effective until the next observation or the delisting date. The first shares observation is effective backwards to the beginning of data.

The shares array cannot be used to directly find the shares outstanding each calendar period. Utility functions or utility programs are available to map observations to time series.

shrout **Number of Shares Outstanding** **int**

`shrout` is the number of publicly held shares, recorded in thousands.

shrsdt **Date of Shares Structure** **int**

`shrsdt` is a specific date corresponding to a shares outstanding value. The shares date is either the statement date from a firm's annual or quarterly report, the ex-distribution date (`exdt`) of the distribution affecting the shares, or the date of a shares observation taken from another source.

shrsenddt **Last Date of Shares Observation** **int**

`shrsenddt` is the last effective date of the shares observation. It is set to the latest date prior to the `shrsdt` of the next observation. The `shrsenddt` of the last observation is set to the delisting date.

shrflg **Share Flag** **int**

`shrflg` is an integer value indicating the sources of the shares structure. A value of zero indicates a share structure extracted from CRSP data sources. A value of one corresponds to a shares structure imputed from a split or other distribution.

delist **Delisting Array** **CRSP_STK_DELIST ***

The delisting array contains information of the status of a security after it is no longer listed on an exchange supported by CRSP. Delisting structures contain a delisting code, delisting date, and information about the next observation after delisting, if available. In this version of the file, the delisting array only contains the most recent delisting structure for each company.

CRSP has developed guidelines for closing an issue to further delisting research. These guidelines apply to all types of delistings. An issue is considered to be closed to further research if any of the following conditions apply:

- Research has verified that a final distribution has been paid to stockholders.
- Research has verified that no distributions are ever to be paid to stockholders.
- Some distributions have been paid to stockholders, but no final distribution information can be found and 10 years have passed since the date of the most recent delisting information.
- No information concerning the delisting can be found and 10 years have passed since the delist date.

If none of these conditions apply to a delisted issue, the issue is considered to be pending, which means that further research is required until one of the above conditions has been met.

dlstdt **Delisting Date** **int**

`dlstdt` contains the date (in YYYYMMDD format) of a security's last price on the current exchange. `dlstdt` is the last daily date even if the price time series is monthly. `dlstdt` is set to the last date in the price time series calendar if the security is still active.

dlstcd **Delisting Code** **int**

`dlstcd` is a three-digit code. It either `(1)` indicates that a security is still trading or `(2)` provides a specific reason for delisting. All coded delistings belong by the first digit of the delisting code.

Delisting Codes

Code	Category
100	still trading or halted but not yet delisted
200	merger
300	exchange
400	liquidation
500	delisted by NYSE/AMEX or Nasdaq
700	delisted by the Securities and Exchange Commission
800	trading simultaneously on more than one exchange

The second and third digits of the delisting codes provide further details of delisting events. Additional delisting codes, specific to various delisting categories, have been created to indicate if an issue is closed to further research, or if the issue is pending further research. (See Appendix A.3 for a complete list of delisting codes.)

nwperm **New CRSP Permanent Number** **int**

nwperm is assigned when an issue ceases trading as a result of a merger or exchange where shareholders receive stock in the acquiring company. The nwperm is the permno of the acquiring company's stock. It acts as a forward pointer, allowing the user to trace the ongoing history of surviving companies. nwperm may identify an issue that exists on a different CRSP stock file.

nwcomp **New CRSP Permanent Company Number** **int**

nwcomp is assigned when an issue ceases trading as a result of a merger or exchange when shareholders receive some payment from the acquiring company. If nwperm is nonzero, nwcomp is the permco of that security. If nwperm is zero, nwcomp can still be nonzero if the shareholders receive a payment from the acquiring company valued at more than fifty percent of the delisting security's value that does include stock.

nextdt **Date of Next Available Information** **int**

nextdt is the date (in YYYYMMDD format) of a security's delisting price, dlprc. It is set to zero if the security is still active, if CRSP determines the final value of the security by a distribution (e.g. merger terms), or if the value of the security is unknown after suspension or delisting. If a liquidation was announced in advance and trading continued on the exchange after the announcement of the liquidation, nextdt is set to dlstdt. Otherwise it is set to the YYYYMMDD date of a price found off the exchange.

If our research determines that a security became worthless at any time after delisting and there is no evidence of any trading of the security in the interim, then nextdt is set to one trading day after the delist date and the delisting price is set to zero.

dlamt **Amount After Delisting** **float**

dlamt is the value of the issue used to calculate the ending value in the delisting return. It is either the price on another exchange after delisting or the sum of merger or exchange payments. In a monthly database if no value after delisting exists and daily prices exist after the last month-end trading date , dlamt is set the last daily price.

dlretx **Delisting Return Without Dividends** **float**

dlretx is the return of the security after it is delisted. Ordinary dividends that exist after the last trading date and up to the date of the delist payment are excluded. See dlret for calculation and missing values.

dlprc **Delisting Price** **float**

dlprc is the price on the date specified in nextdt. If nextdt is zero, there is no further trading of the security or quotes are unknown, and dlprc is set to zero. Otherwise dlprc refers to a trade price or quote on a new exchange.

If dlprc is positive, then it is a trade price. If dlprc is negative, then it represents the average of bid and ask quotations. If dlprc is zero and nextdt is after dlstdt, it indicates the stock became worthless.

dlpdt **Date of Delisting Payment** **int**

dlpdt is the effective date of any amount after delisting used in delisting return payments, in YYYYMMDD format. It is set to 0 if no payments are available. If there are multiple payments, dlpdt is set to the exdt of the last payment.

dlret **Delisting Return** **float**

dlret is the return of the security after it is delisted. It is calculated by comparing a value after delisting against the price on the security's last trading date. The value after delisting can include a delisting price or the amount from a final distribution. If nextdt is nonzero, it can be used as the effective date of dlret. Otherwise, by convention, enddt is the effective date of the delisting return.

dlret is only set to a non-missing value if the information for a value after delisting used in calculating the return pertains to the period for which the return is being calculated. There are four possible missing return values for this field.

Missing Return Codes

dlret	Reason For Missing Return
-55.0	CRSP has no sources to establish a value after delisting or is unable to assign a value to one or more known distributions after delisting.
-66.0	more than 10 trading periods between a security's last price and its first available price on a new exchange.
-88.0	security is still active.
-99.0	security trades on a new exchange after delisting, but CRSP currently has no sources to gather price information.

For any issue that has been closed to further research, a delisting return is calculated as follows:

1. If a final distribution has been verified, the delisting return is calculated using all known distribution information.

2. If distributions are available, but no final distribution is known, the delisting return is calculated using all known distribution information .

3. If there is evidence that no distributions will ever be paid to stockholders, the stock is considered worthless and the delisting return is set to -1 (100% loss).

For any issue that is pending further research, delisting codes 470 and 480 only, the missing return code is set to -55.

nasdin **Nasdaq Information History Array** **CRSP_STK_NASDIN ***

The Nasdaq information history array describes an issue's trading status on The Nasdaq Stock Market[SM]. This information is supplied by the National Association of Securities Dealers, Inc. (NASD). Nasdaq information structures are available for securities trading on Nasdaq beginning on April 1, 1982 for traits date and The Nasdaq National Market indicator. All fields are available beginning November 1, 1982[7]. If the traits code, The Nasdaq National Market indicator, the NASD index, or the number of market makers changes, then a new structure is added, and the date of the change is recorded in `trtsdt`.

The Nasdaq National Market was initiated in April 1982 for larger and generally more actively traded Nasdaq securities. The Nasdaq National Market Securities must meet higher financial and non-financial criteria than other Nasdaq stocks, and are subject to last-sale reporting. In June of 1992 the regular Nasdaq segment of The Nasdaq Stock Market[SM] was renamed The Nasdaq Small Cap Market and for the first time these became subject to real-time price and volume reporting.

trtsdt **Traits Date** **int**

`trtsdt` is the effective begin date for the traits structure. It is stored as YYYYMMDD.

trtsenddt **Last Effective Date for Traits** **int**

`trtsenddt` is the last date, in YYYYMMDD format, that information in the structure is valid for the security. `trtsenddt` is set to the last trading date before the next structure, or to 99999999 in the last structure.

trtscd **Traits Code** **int**

`trtscd` is a one-digit code describing the trading status of the issue.

Nasdaq Trading Status Code

TRTSCD	Description
0	unknown
1	active
2	trading with only one market maker
3	suspended
4	inactive
5	delisted

7 Nasdaq information data are missing in December, 1982 for all issues with NASD company numbers less than 1025 (approximately 20 percent of the securities active at that time), and are missing in February, 1986 for all issues. `TRTSDT`, `TRTSCD`, and `NMSIND` are complete. All other fields are missing.

nmsind **Nasdaq National Market Indicator** **int**

nmsind is a one-digit code indicating whether or not the issue is a member of The Nasdaq National Market (formerly NMS). Prior to June 15, 1992, transaction data was not available for Nasdaq Small Cap Securities. As of June 15, 1992 transaction data became available.

Nasdaq National Market Indicators

NMSIND	Description
0	unknown
1	The Nasdaq Small Cap Market before June 15, 1992
2	The Nasdaq National Market
3	The Nasdaq Small Cap Market after June 15, 1992

mmcnt **Market Maker Count** **int**

mmcnt is the number of registered market makers for the issue. This will contain a 0 if there are no registered market makers for that time, or if the date falls in December of 1982 for a NASD Company Number less than 1025 or in February of 1986, in which case the count is missing.

nsdinx **NASD Index** **int**

nsdinx is a one-digit code indicating the issue's classification within NASD's internal business description categories.

NASD Internal Business Description Categories

NSDINX	Description
0	unknown
1	no index
2	industrial company
3	bank
4	other financial institution
5	insurance company
6	transportation company
7	utility company

Time Series Data

bidlo **Bid or Low Time Series** **float ***

In a monthly database, bidlo is the lowest closing price or bid/ask average during each month. In a daily database, if prc is positive then bidlo is the lowest sale during a trading day. If prc is negative, then bidlo is the closing bid on that day. bidlo is set to zero if no bid or low price is available.

Daily trading prices for The Nasdaq National Market securities were first reported November 1, 1982. Daily trading prices for the The Nasdaq Small Cap Market were first reported June 15, 1992. bidlo for Nasdaq securities is always a bid before this time.

askhi **Ask or High Time Series** **float ***

In a monthly database, askhi is the highest closing price or bid/ask average during each month. In a daily database, if prc is positive then askhi is the highest sale during a trading day. If prc is negative, then askhi is the closing ask on that day. askhi is set to zero if no ask or high price is available.

Daily trading prices for The Nasdaq National Market securities were first reported November 1, 1982. Daily trading prices for The Nasdaq Small Cap Market were first reported June 15, 1992. askhi for Nasdaq securities is always an ask before this time.

prc **Closing Price or Bid/Ask Average Time Series** **float ***

prc is the closing price or the negative bid/ask average for a trading day . If the closing price is not available on any given trading day, the number in the price field has a negative sign to indicate that it is a bid/ask average and not an actual closing price. Please note that in this field the negative sign is a symbol and that the value of the bid/ask average is not negative. If neither closing price nor bid/ask average is available on a date, prc is set to zero.

In a monthly database, prc is the price on the last trading date of the month. The price series begins the first month-end after the security begins trading and ends the last complete month of trading.

If the security of a company is included in the Composite Pricing network, the closing price while listed on NYSE or AMEX on a trading date is the last trading price for that day on the exchange that the security last traded. Similarly, highs, lows, and volumes include trades on all exchanges on which tat security traded. For example, if a stock trades on both the NYSE and the PACX (Pacific Stock Exchange), and the last trade occurs on the PACX, the closing price on that day represents the closing price on the PACX, not the NYSE.

Price data for Nasdaq securities comes directly from the NASD with the close of the day at 4:00 p.m. Eastern Time. Automated trades after hours on Nasdaq are counted on the next trading date, although the volumes are applied to the current date.

Daily trading prices for The Nasdaq National Market securities were first reported November 1, 1982. Daily trading prices for The Nasdaq Small Cap Market were first reported June 15, 1992. prc for Nasdaq securities is always a negative bid/ask average before this time.

All prices are raw prices as they were reported at the time of trading.

ret **Holding Period Return Time Series** **float ***

A return is the change in the total value of an investment in the security over some period of time per dollar of initial investment. ret is the return for a sale on the given day. It is based on a purchase on the most recent time previous to this day when the security had a valid price. Usually, this time is the previous calendar period.

Returns are calculated as follows:

For time *t* (a holding period), let
t' = time of last available price < *t*
r(t) = return on purchase at *t'*, sale at *t*
p(t) = last sale price or closing bid/ask average at time *t*
d(t) = cash adjustment for *t*
f(t) = price adjustment factor for *t*
p(t') = last sale price or closing bid/ask average at time of last available price < *t*.

$$r(t) = \frac{p(t)\,f(t) + d(t)}{p(t')} - 1$$

t' is usually one period before *t*, but *t'* can be up to ten periods before *t* if there are no valid prices in the interval.

ret is a total return. In daily databases, dividends are reinvested in the security on the exdt. In monthly databases, the returns are holding period returns from month-end to month-end, not compounded daily returns, and dividends are reinvested in the security at month-end.

A series of special return codes specify the reason a return is missing.

Missing Return Codes

ret(*t*)	Reason For Missing Return
-66.0	valid current price but no valid previous price. Either first price, unknown exchange between current and previous price, or more than 10 periods between time *t* and the time of the preceding price *t'*
-77.0	not trading on the current exchange at time *t*
-88.0	outside the range of the security's price range
-99.0	missing return due to missing price at time *t*

vol **Share Volume Time Series** **int ***

In monthly databases, vol is the sum of the trading volumes during that month. In daily databases, vol is the total number of shares of a stock traded on that day. Volume is expressed in units of one share, although our data source for NYSE/AMEX reports the number rounded to the nearest hundred. For example, 12,345 shares traded will be reported on the Nasdaq Stock Exchange as 12,345 and on the NYSE or AMEX exchanges as 12,300.

Volume is set to –99 if the value is missing. A volume of zero usually indicates that there were no trades on that day and is usually paired with bid/ask quotes in price fields.

Volumes are raw values in terms of actual shares traded on the date. Monthly volumes are the sum of raw shares and are not adjusted for splits during the month.

On Nasdaq, volumes of after-hours trades are included in the current day, while the trades or quotes are included the next day. Therefore, it is possible to have bid/ask or missing price quotes paired with nonzero volumes.

NYSE/AMEX volumes are the sum of volumes on all exchanges where that security traded that day.

bid **Nasdaq Closing Bid Time Series** **float ***

bid is the closing bid on a trading date. It is only available for issues trading on The Nasdaq Stock Market[SM] during time periods when bidlo can contain the low price. bid is reported for all securities listed on The Nasdaq National Market since November 1, 1982, and all Nasdaq securities since June 15, 1992.

The close of the day is 4:00 PM Eastern time. Since July 1980, Nasdaq has used the inside quotation as the closing bid and ask. The inside quotation is the highest bid and lowest ask.

Due to source limitations, data is missing for 15 Nasdaq National Market securities in December, 1982, and all The Nasdaq National Market securities in February, 1986.

In monthly databases, bid is the closing bid on the last trading date of each month.

ask **Nasdaq Closing Ask Time Series** **float ***

ask is the closing ask on a trading date. It is only available for issues trading on The Nasdaq Stock Market[SM] during time periods when askhi can contain the low price. Ask is reported for all securities listed on The Nasdaq National Market since November 1, 1982, and all Nasdaq securities since June 15, 1992.

The close of the day is 4:00 PM Eastern time. Since July 1980, Nasdaq has used the inside quotation as the closing bid and ask. The inside quotation is the highest bid and lowest ask.

Due to source limitations, data is missing for 15 Nasdaq National Market securities in December, 1982, and all The Nasdaq National Market securities in February, 1986.

In monthly databases, ask is the closing ask on the last trading date of each month.

retx **Return Without Dividends Time Series** **float ***

retx contains returns without dividends, or capital appreciation. Ordinary dividends and certain other regularly taxable dividends are excluded from the returns calculation. See ret for missing values. The formula is the same as for ret except *d(t)* is usually 0.

spread **Month End Bid/Ask Spread Time Series** **float ***

spread is secondary price information for monthly databases. If prc is negative and spread is nonzero, spread is the difference between bid and ask on the date. If prc is zero and spread is negative, spread represents a bid price. If prc is zero and spread is positive, spread represents an ask price.

numtrd **Nasdaq Number of Trades or Alternate Price Date Time Series** **int ***

In daily databases, numtrd contains the number of trades each date. If the number of trades is unavailable numtrd is set to –99.

Number of trades is only available for issues trading on The Nasdaq Stock Market[SM]. It is reported for all securities listed on The Nasdaq National Market since November 1, 1982, and all Nasdaq securities since June 15, 1992. Data is missing for 15 Nasdaq National Market securities in December, 1982, and all The Nasdaq National Market securities in February, 1986.

In monthly databases numtrd contains the date of altprc, in YYYYMMDD format. If altprc contains a nonzero price, numtrd contains the date of that price. If there are no non-missing prices in the month, altprc and numtrd are set to zero. New issues that do not begin on the last trading date of a month have the first price and date of the first price at the beginning of the altprc and numtrd time series.

altprc **Alternate Price Time Series** **float ***

altprc is an alternate monthly price derived from daily prices. altprc contains the last non-missing price in the month. The date of this price is in numtrd. If there are no non-missing prices in the month, altprc and numtrd are set to zero. New issues that do not begin on the last trading date of a month have the first price and date of the first price at the beginning of the altprc and numtrd time series.

altprc is only available on monthly databases during time periods when daily data are available.

port **Portfolio Statistics and Assignment Time Series** **CRSP_STK_PORT ****

port is a set of portfolio time series. Each portfolio time series is based on a portfolio type and contains a history of statistics and portfolio assignments. Portfolio types are predefined groupings based on CRSP indices. The portfolio time series can be linked to CRSP index returns data to calculate excess returns of a security against its assigned index portfolio at any time during its history.

A capitalization portfolio group is provided with all CRSPAccess97 stock databases. Additional groups are available with the CRSPAccess97 Indices File / Portfolio Assignment companion product. The associated portfolio group indices are also available in the CRSPAccess97 Indices File / Portfolio Assignments Product.

Each portfolio type represents a predefined index group with its own methodology and rebalancing period. The portfolio time series can be linked to different calendars based on the rebalancing frequency of the index, and the timing and calculation of the statistic and assignment rules are also dependent on the index. Calendars used in portfolios are not the same calendars used for security price and returns data. Portfolio time series header information can differ for all portfolio types. The time series header element subtype contains the indno of an index that contains the performance results of the index series built using the assignments.

NYSE/AMEX or NYSE/AMEX/Nasdaq capitalization deciles are provided with the respective NYSE/AMEX and NYSE/AMEX/Nasdaq stock files. In these portfolios, all securities except ADRs are ranked by capitalization each year. The capitalization at the end of the previous year is used to rank the security. If a capitalization is not available at the end of the previous year, the capitalization on the date with the earliest available price in the current year is used for ranking. The securities on the file with available capitalization each year are divided into ten portfolios with equal counts. The largest securities are placed in portfolio 10 and the smallest in portfolio 1. Port is set to zero if the security is not assigned a portfolio in a year.

The capitalization portfolio statistic is the capitalization, in 1000's, of a non-ADR security trading on one of the valid exchanges. If the security is an ADR, has no shares, or did not trade at year-end, the statistic is set to 0.

The following table describes the possible portfolio types available. The available types differ in daily and monthly databases and by product. All portfolio types are available only in the CRSPAccess97 Indices File / Portfolio Assignment Product. See the **CRSPAccess97 Indices File Guide** for details of the different portfolio methodologies.

Portfolio Type Name	Statistic	Rebalancing Calendar	Portfolio with highest statistics	Associated indno	Daily Portfolio Type	Monthly Portfolio Type
NYSE/AMEX/Nasdaq Capitalization Deciles	Year-end Market capitalization	Annual	10	1000092	1	1
NYSE/AMEX Capitalization Deciles	Year-end Market capitalization	Annual	10	1000052	2	2

The following portfolio types are available with the CRSPAccess97 Indices File / Portfolio Assignment Product. See the Indices Guide for additional information.

Portfolio Type Name	Statistic	Rebalancing Calendar	Portfolio with highest statistics	Associated indno	Daily Portfolio Type	Monthly Portfolio Type
Nasdaq Capitalization Deciles	Year-end Market capitalization	Annual	10	1000072	3	3
NYSE Capitalization Deciles	Year-end Market capitalization	Annual	10	1000012	4	4
AMEX Capitalization Deciles	Year-end Market capitalization	Annual	10	1000032	5	5
NYSE/AMEX Beta Deciles	Preceding Annual Beta	Annual	1	1000112	6	
NYSE/AMEX Standard Deviation Deciles	Preceding Annual Standard Deviation	Annual	1	1000132	7	
Nasdaq Beta Deciles	Preceding Annual Beta	Annual	1	1000152	8	
Nasdaq Standard Deviation Deciles	Preceding Annual Standard Deviation	Annual	1	1000172	9	
Cap-Based NYSE/AMEX/Nasdaq National Market Portfolios	Quarter-end Market Capitalization	Quarterly	1	1000357		6
Cap-Based NYSE Market Portfolios	Quarter-end Market Capitalization	Quarterly	1	1000317		7
Cap-Based NYSE/AMEX Market Portfolios	Quarter-end Market Capitalization	Quarterly	1	1000337		8

port **Portfolio Assignment** **int**

port is the assignment of the security for the portfolio type for the time period. If no assignment is made for the security during the period, port is set to zero.

Portfolio assignment rules are based on the methodology in a portfolio type. The time period of port is the time the security is held in the portfolio, but is usually based on the statistic in a previous period.

stat **Portfolio Statistic** **double**

stat is the statistic calculated for the security for the time period. If no statistic is calculated, a missing value dependent on the portfolio statistic is set. Missing market capitalizations are set to zero, and missing beta or standard deviations are set to –99.0.

Statistic calculations are based on the methodology of the portfolio type. The statistic is for the current period, and usually determines the portfolio assignment of the next period.

2.4 C Index Structure and Data Items

In CRSPAccess97 databases, all indexes are assigned a permanent number called `indno` and have common data items. The data items includes identifying and descriptive information, index composition information, characteristic counts and weights, and result returns and index levels based on total returns, capital appreciation, and dividend yields. Not all items are available for each index.

There are two types of index sets supported:

1. **`series`** – all `indno`s represent a single index, such as the CRSP NYSE Value-Weighted Market Index , the CRSP NYSE Equal-Weighted Market Index, or the CRSP 9-10 Cap-Based index for the NYSE/AMEX/Nasdaq National Market. Each index is an independent portfolio.

2. **`group`** – all `indno`s represent a group of related series based on the same rules, such as the portfolios in the CRSP Cap-Based index for the NYSE/AMEX/Nasdaq National Market. All group `indnos` are tied to CRSP Stock portfolio types described in the port data item and are only available in the CRSPAccess97 Indices File / Portfolio Assignments Product. These types can be used to determine which series in the group is applicable to each security at any given point in time. Each series within the group is also available in the `series` set under its own `indno`.

An index must be referenced by a set identification number `setid` and an `indno`. The `setid` is based on the frequency of the time series and whether the `indno` refers to a `series` or `group`. An `indno` will be present in either a `series` or a `group` `setid` but not both. However, the same `indno` can appear in daily and monthly sets. The following set identifiers are available:

SETID	**Description**
400	Monthly frequency `groups`
420	Monthly frequency `series`
440	Daily frequency `groups`
460	Daily frequency `series`

Each CRSPAccess97 database directory contains a header file named `headind.dat` with a list of index `setid`s and `indno`s available in that index. Utility programs *dindsearch* and *mindsearch* can be used to search the list for `indno`s and `setid`s. The list of `indno`s available in the Stock Files are in the following table. The availability of these depends upon the universes of your product subscription. See `/CAL/` descriptions in Section 2.2 for the methodology used for each index. See the **CRSPAccess97 Indices File Guide** for a complete list of all Index Series and Portfolio Types available in the CRSPAccess97 Indices File / Portfolio Assignments Companion product.

CRSP Indices

Index	**indno**	**Daily `setid`**	**Monthly `setid`**
CRSP NYSE/AMEX Value-Weighted Market Index	1000040	460	420
CRSP NYSE/AMEX Equal-Weighted Market Index	1000041	460	420
CRSP NYSE/AMEX/Nasdaq Value-Weighted Market Index	1000080	460	420
CRSP NYSE/AMEX/Nasdaq Equal-Weighted Market Index	1000081	460	420
S&P 500 Composite	1000502	460	420
Nasdaq Composite	1000503	460	420

Each index in monthly set 400 or daily set 440 is associated with a portfolio type available in stock security data. The `port` time series set in CRSPAccess97 stock databases has `subtype` set to the `indno` of the index group. Index groups are only available in the CRSPAccess97 Indices File / Portfolio Assignments companion product.

CRSP Indices Data Objects

The following table summarizes the primary data items available in a CRSP index record. See the CRSP Data Objects section for definitions of the object types and the CRSP Index Data Items section for the elements in the index structures.

The object name variable includes header object data plus the item array. Actual data is stored in the item array. Sets of objects represent an array of the given object type. The number of sets variable is the number of objects in this array. Each object in a set has its own set of object header variables. Sets of objects are used to represent groups of related objects, such as the different decile portfolios in a group index.

Item	Object Type	Object Name	Item Name	Number of Sets
Index Header	CRSP_ROW	indhdr_row	indhdr	
Set of Rebalancing Arrays	CRSP_ARRAY	rebal_arr	rebal	rebaltypes
Set of List Arrays	CRSP_ARRAY	list_arr	list	listtypes
Set of Used Count Time Series	CRSP_TIMESERIES	usdcnt_ts	usdcnt	indtypes
Set of Total Count Time Series	CRSP_TIMESERIES	totcnt_ts	totcnt	indtypes
Set of Used Value Time Series	CRSP_TIMESERIES	usdval_ts	usdval	indtypes
Set of Total Value Time Series	CRSP_TIMESERIES	totval_ts	totval	indtypes
Set of Total Return Time Series	CRSP_TIMESERIES	tret_ts	tret	indtypes
Set of Capital Appreciation Time Series	CRSP_TIMESERIES	aret_ts	aret	indtypes
Set of Income Return Time Series	CRSP_TIMESERIES	iret_ts	iret	indtypes
Set of Total Return Index Levels	CRSP_TIMESERIES	tind_ts	tind	indtypes
Set of Capital Appreciation Index Levels	CRSP_TIMESERIES	aind_ts	aind	indtypes
Set of Income Return Index Levels	CRSP_TIMESERIES	iind_ts	iind	indtypes

CRSP Index Data Item Descriptions

indhdr **Index header** **CRSP_IND_HEADER**

indhdr is a structure containing index header information. This includes identification and description fields describing the composition and methodology of the index.

indno **CRSP Permanent Index Number** **int**

indno is the unique permanent identifier assigned by CRSP to every supported index. All indnos are seven digit numbers. There is no inherent meaning in the numbers. The indices sets in a database are sorted by this field.

indco **CRSP Permanent Index Group Number** **int**

indco is the permanent number assigned by CRSP to all groups of indices. All indices based on the same market and statistical grouping are assigned the same indco.

primflag **Link to Primary Index** **int**

primflag is either the indno of a group index or a zero if there is no primary group index. A series index representing one portfolio of a group can use primflag to refer back to the primary index. The primary index will contain rebalancing information and data for all portfolios in that group. Only series indices have nonzero indno.

portnum **Portfolio Number if Subset Series** **int**

portnum is the portfolio number of an index series that is also a member of an index group. Primflag is the indno of the index group. This index is the portnum'th series within the group index or zero if there is no primary group index.

indname **Index Name** **char[80]**

indname is the name of the portfolio index. The index names are listed in the CRSP indnos table.

typename **Index Group Name** **char[80]**

typename is the name of the index group to which the index belongs. All indices with the same indco will have the same typename.

method **Methodology Description Structure** **CRSP_IND_METHOD**

method contains fields describing the rules used to build the index. Fields contain information on primary and secondary methodologies and rules for weighting securities within the index.

methcode **Method Type Code** **int**

methcode is a code for the combination of primtype, subtype, wgttype, and wgtflag characteristics. Methodologies are described earlier in this section. Current codes are:

Code	Description
1	CRSP Cap-Based Portfolios
3	CRSP Risk-Based Indices
4	CRSP Value-Weighted Market Indices
5	CRSP Equal-Weighted Market Indices
6	CRSP Capitalization Decile Market Indices
7	S&P 500® Composite
8	CRSP Value-Weighted Index on the S&P 500® Universe
9	CRSP Equal-Weighted Index on the S&P 500® Universe
10	Nasdaq Composite
12	CRSP Fixed Term Bond Returns
13	CRSP Fixed Term Bill Returns
14	Provided by External Source

primtype **Primary Methodology Type** **int**

primtype is a code describing the type of index. The possible index types are

Code	Name	Description
0	Fractile Index	Breakpoints based on some rule and/or statistic are used to divide eligible issues into portfolios at different intervals. The breakpoint function is continuous so that all eligible issues are in exactly one portfolio during each period.
1	Selected Index	Universe is supplied from an outside source, with given issues or companies and the data ranges for each.
3	Market Index	Portfolio of all eligible issues is reevaluated each period based on constant universe restrictions
4	Other	Not applicable

subtype **Secondary Methodology Group** **int**

subtype is a code for further detail of the primary index methodology type. The following codes are used

Code	Description
0	No further description
10	Portfolios based on market capitalization
12	Portfolios based on result statistic: beta or standard deviation
13	Issues in S&P 500® Index
14	Issues in the Nasdaq Composite Index
15	Treasury issues of selected maturity ranges

wgttype **Reweighting Type Flag** **int**

wgttype is a code indicating the method of weighting the issues in the portfolio index. The following codes are used.

Code	Description
0	Not available
1	Value-weighted, weights not supplied by CRSP
2	Value-weighted
3	Equal-weighted

wgtflag **Reweighting Timing Flag** **int**

wgtflag is a code indicating how frequently the weights are applied to the existing portfolio. The following codes are used.

Code	Description
0	Not available
11	Weights are applied each time period

flags **Exception Handling Flags** **CRSP_IND_FLAGS**

flags is a group of fields describing how the index supports exceptions in the data, such as new and delisted issues and missing data. Flags contains fields flagcode, addflag, delflag, delretflag, and missflag.

flagcode **Code of Basic Exception Types** **int**

flagcode is a code of the basic exception types. The following codes are currently used.

Code	Description
0	Unknown or Not available
1	CRSP Market Index Flags
2	Cap-Based Index Flags
3	CRSP Market Index Trade-only Prices Flags

addflag **Handling of New Issues Flag** **int**

addflag is a code describing how new issues are used in an index. The following codes are used.

Code	Description
0	Unknown or Not available
1	Adding securities are included the first period they meet existing portfolio restrictions

delflag **Handling of Ineligible Issues Flag** **int**

delflag is a code describing how issues that become ineligible are handled. The following codes are used.

Code	Description
0	Unknown or Not available
1	Issues becoming ineligible are held until the next time period

delretflag **Return of Delisted Issues Flag** **int**

delretflag is a code describing whether delisting returns are applied to securities delisting from the exchange during a rebalancing period. The following codes are used.

Code	Description
0	Unknown or Not applicable
1	Delisting return is applied to issues that delist during the period
2	Only returns while trading on the target exchange are used

missflag **Flag for Handling Missing Data** **int**

missflag describes the possible actions taken for securities with missing data during the range in an index portfolio. The following codes are used.

Code	Description
0	Unknown or Not applicable
3	Issues without single period returns are excluded
5	Alternate prices are used if possible to generate single period returns
13	Quotes without trades are treated as missing prices

partuniv **Partition Subset Screening Structure** **CRSP_UNIV_PARAM**

induniv **Index Subset Screening Structure** **CRSP_UNIV_PARAM**

partuniv and induniv are structures of fields used to restrict a database using various screening variables. The screen fields are univcode, begdt, enddt, wantexch, wantnms, wantwi, wantinc, and shrcd. partuniv screens are used to restrict the securities used in defining partition breakpoints. induniv screens are used to restrict the securities used in the actual index.

univcode **Code of Universe Subset Types** **int**

univcode is a code defining a set of restrictions used in subset screening. The following codes are used.

Code	Description
0	Identifier restriction not applicable
10	NYSE common excluding foreign, ADR, REIT, Closed End Funds
11	NYSE/AMEX common excluding foreign, ADR, REIT, Closed End Funds
12	NYSE/AMEX/The Nasdaq National Market common excluding foreign, ADR, REIT, Closed End Funds
20	NYSE common excluding ADRs
21	AMEX common excluding ADRs
22	NYSE/AMEX common excluding ADRs
23	Nasdaq common excluding ADRs
24	NYSE/AMEX/Nasdaq common excluding ADRs
30	NYSE common
31	AMEX common
32	NYSE/AMEX common
33	Nasdaq common
34	NYSE/AMEX/Nasdaq common
35	NYSE common excluding ADRs and foreigns
36	AMEX common excluding ADRs and foreigns
37	NYSE/AMEX common excluding ADRs and foreigns
38	Nasdaq common excluding ADRs and foreigns
39	NYSE/AMEX/Nasdaq common excluding ADRs and foreigns

begdt **First Trading Date Allowed in Restriction** **int**

begdt is the first date, in YYYYMMDD format, of data included in the universe. begdt is set to 0 if there is no date restriction.

enddt **Last Trading Date Allowed in Restriction** **int**

enddt is the first date, in YYYYMMDD format, of data included in the universe. enddt is set to 0 if there is no date restriction.

wantexch **Valid Exchange Codes in Universe** **int**

wantexch is code indicating the base exchanges in the universe subset. The following codes are used. The sum of two or more codes indicates those selected exchanges are valid.

Code	Description
0	No exchange restriction
1	New York Stock Exchange
2	American Stock Exchange
4	Nasdaq Stock Market

wantnms **Valid Nasdaq Market Groups in Universe** **int**

wantnms is a code indicating valid Nasdaq markets in the universe subset. The Nasdaq markets are subsets of The Nasdaq Stock Market[SM]. The following codes are used.

Code	Description
0	No National Market restriction or not applicable
1	Only issues listed on The Nasdaq National Market are included

wantwi **Valid When-Issued Securities in Universe** **int**

wantwi describes the types of when-issued trading allowed in a subset. The following codes are used.

Code	Description
0	No When-Issued restrictions or Not applicable
10	Initial when-issued trading is included when available. Ex-Distributed Trading is excluded. When-issued trading during reorganizations is included
110	Initial when-issued trading is excluded until issue attains Regular-Way status. Ex-Distributed Trading is excluded. When-issued trading during reorganizations is included.

wantinc **Valid Incorporation of Securities in Universe** **int**

wantinc describes the incorporation of companies selected in a subset. The following codes are used.

Code	Description
0	Not applicable or no restriction by country of incorporation
1	Companies incorporated outside of the U.S. are excluded

shrcd **Share Code Screen Structure** **CRSP_UNIV_SHRCD**

shrcd contains fields defining groups of CRSP security share codes included in the subset. See Apepndix A for details of the two-digit share codes used by CRSP in the shrcd data item. The fields in the structure are sccode, fstdig, and secdig.

sccode **Share Code Groupings for Subsets** **int**

sccode is a code describing the generic share code groupings used in subsets. The following codes are used.

Code	Description
0	No Share Code Restriction or Not applicable
1	Common stocks excluding ADRs
2	Common stocks excluding ADRs and foreign incorporated companies
3	Common stocks excluding ADR's, foreign incorporated companies, REITS, and Closed End Funds
4	Common stocks

fstdig **Valid First Digit of Share Code** **int**

fstdig is a binary code describing the valid digits in the first digit of the share code in a subset universe. fstdig is the decimal representation of a 10-digit binary number. The **n**th digit of the binary number is 1 if an **n** in the first digit of the share code is valid in the subset, and a 0 otherwise.

secdig **Valid Second Digit of Share Code** **int**

secdig is a binary code describing the valid digits in the second digit of the share code in a subset universe. secdig is the decimal representation of a 10-digit binary number. The **n**th digit of the binary number is 1 if an **n** in the second digit of the share code is valid in the subset, and a 0 otherwise.

rules **Rules For Building Portfolios** **CRSP_IND_RULES**

rules is a group of fields describing any rules used to build portfolios. rules contains fields rulecode, buyfnct, sellfnct, statfnct, and groupflag.

rulecode **Code of Basic Rule Types** **int**

rulecode is a code of basic rule types. The following codes are currently used.

Code	Description
0	Unknown or Not applicable
1	Group by Previous Period End Issue Capitalization
2	Group by Previous Period End Company Capitalization
3	Group by Scholes-Williams Beta over previous year
4	Group by Standard Deviation over previous year

buyfnct **Function Code for Buy Rules** **int**

buyfnct is a code defining a function used to determine whether to buy new issues in a portfolio at a rebalancing period. It is always 0 in current indices.

sellfnct **Function Code for Sell Rules** **int**

sellfnct is a code defining a function used to determine whether to sell current issues in a portfolio at a rebalancing period. It is always 0 in current indices.

statfnct **Function Code for Generating Statistics** **int**

statfnct is a code defining a function used to generate a statistic to be used in determining inclusion in a portfolio when it is rebalanced. The following codes are used.

Code	Description
0	Unknown or Not applicable
1	Capitalization at end of Previous Period
2	Scholes-Williams Beta over previous year
3	Standard Deviation over previous year

groupflag **Statistic Grouping Code** **int**

groupflag is a code describing the type of grouping done on issues before any statistics are applied. The following codes are used.

Code	Description
0	Unknown or Not applicable
1	Each issues is grouped independently
2	Multiple issues of a company are combined

assign **Related Assignment Information** **CRSP_IND_ASSIGN**

assign is a group of fields defining the time periods and associated indexes used to form portfolios. It primarily defines the rebalancing periods when the portfolio is reformed based on new information. assign contains fields assigncode, asperm, asport, rebalcal, assigncal, and calccal.

assigncode **Code of Basic Assignment Types** **int**

assigncode is a code of basic assignment types. The following codes are currently used.

Code	Description
0	Unknown or Not applicable
1	Annual rebalancing
2	Quarterly rebalancing
3	Monthly rebalancing

asperm **Indno of Associated Index** **int**

asperm is an indno of an associated index used to supply rebalancing breakpoint information used for assignments or buy/sell rules to this index. It is always 0 in current indices.

asport **Portfolio Number in Associated Index** **int**

asport is the portfolio number in an associated index defined in asperm. The associated portfolio's breakpoint information for that portfolio is used for the current index. It is always 0 in current indices.

rebalcal **Calid of Rebalancing Calendar** **int**

rebalcal identifies a calendar that determines the dates when the portfolios in the index are held. The new portfolio universe is held from one period in the rebalancing calendar until the next. The following calendars are used.

Code	Description
0	Unknown or Not applicable
300	Annual Calendar
310	Quarterly Calendar

assigncal **Calid of Assignment Calendar** **int**

assigncal identifies a calendar that determines the dates when breakpoints and buy/sell rules are valid. The assignment calendar is used when using rebalancing information to assign issues to a portfolio. The period numbers of the calendars of the rebalcal, assigncal, and calccal are synchronized, although the actual date ranges for each period number may be different. assigncal uses the same calendars described in rebalcal.

calccal **Calid of Calculations Calendar** **int**

calccal identifies a calendar that determines the range of dates used to calculate statistics used to form portfolios. The period numbers of the calendars of the rebalcal, assigncal, and calccal are synchronized, although the actual date ranges for each period number may be different. calccal uses the same calendars described in rebalcal.

rebal **Rebalancing History** **CRSP_IND_REBAL**

Rebal is a set of event array structures containing historical rebalancing statistical information for rebalancing periods in an index. The variable rebaltypes contains the count of the rebalancing history arrays available for all indices in a set. There are 10 possible rebalancing arrays in group indices and 1 in series indices. Each array has its own count of periods, which is zero if not applicable to the particular index.

Each period contains the fields rbbegdt, rbenddt, usdcnt, maxcnt, totcnt, endcnt, minid, maxid, minstat, maxstat, medstat, and avgstat. Not all statistics are available for each index.

rbbegdt **Rebalancing Beginning Date** **int**

rbbegdt is the date, in YYYYMMDD format, of the first date in the rebalancing period.

rbenddt **Rebalancing Ending Date** **int**

rbenddt is the date, in YYYYMMDD format, of the last date in the rebalancing period.

usdcnt **Count Used as of Rebalancing** **int**

usdcnt is the count of entities in the portfolio as of the beginning of the rebalancing period.

maxcnt **Maximum Count During Period** **int**

maxcnt is the largest count of issues in the portfolio at any point within the rebalancing period.

totcnt **Available Count as of Rebalancing** **int**

totcnt is the count of entities available in the universe eligible for the portfolio at the beginning of the rebalancing period

endcnt **Count at End of Rebalancing Period** **int**

endcnt is the count of entities belonging to the portfolio at the end of the rebalancing period.

minid **Identifier of Entity with the Minimum Statistic** **int**

minid is the identifier of the entity in the portfolio with the minimum statistic at the beginning of the rebalancing period.

maxid **Identifier of Entity with the Maximum Statistic** **int**

maxid is the identifier of the entity in the portfolio with the maximum statistic at the beginning of the rebalancing period.

minstat **Minimum Statistic in Period** **double**

minstat is the minimum statistic value in the portfolio at the beginning of the rebalancing period.

maxstat **Maximum Statistic in Period** **double**

maxstat is the maximum statistic value in the portfolio at the beginning of the rebalancing period.

medstat **Median Statistic in Period** **double**

medstat is the median statistic value in the portfolio at the beginning of the rebalancing period.

avgstat **Average Statistic in Period** **double**

avgstat is the average statistic value in the portfolio at the beginning of the rebalancing period.

list **List History** **CRSP_IND_LIST**

list is a set of array structures containing lists of issues comprising an index. The variable listtypes contains the count of the issue lists available for all indices in a set. There is 1 possible list arrays in group and series indices. Each array has its own count, which is set to zero if not applicable to the particular index.

Each list structure contains the fields permno, begdt, enddt, subind, and weight.

No list histories are available in the current index database.

permno **CRSP Permanent Number** **int**

permno is the CRSP permanent number, identifying the security included in the list.

begdt **First Date Included** **int**

begdt is the date, in YYYYMMDD format, of the first date the issue is included in the portfolio.

enddt **Last Date Included** **int**

enddt is the date, in YYYYMMDD format, of the last date the issue is included in the portfolio.

subind **Subcategory Code** **int**

subind is a flag indicating a subcategory of the primary index to which the security belongs. It is set to zero if there no subcategory applicable.

weight **Weight of Issue** **double**

weight is the defined weight of the issue within the index during the range indicated. It is set to zero if weighting is defined based on data and not part of the list definition.

usdcnt **Portfolio Used Count Arrays** **int**

usdcnt is a set of time series containing counts of issues used to create the index each period. In a market index, a security must have valid prices the current period and previous period to be included in the count.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible usdcnt time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if usdcnt is not available for the particular index.

totcnt **Portfolio Eligible Count Arrays** **int**

totcnt is a set of time series containing counts of issues eligible for the index each period. In a market index, a security must have valid prices the current period to be included in the count.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible totcnt time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if totcnt is not available for the particular index.

usdval **Portfolio Used Weight Arrays** **double**

usdval is a set of time series containing weights of issues used to create a value-weighted index each period. In a capitalization market index, usdval is the total market value, in $1000's, of all securities that are used in the index for a given period. A security must have available shares outstanding and valid prices the current period and previous period to be included in the weight.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible usdval time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if usdval is not available for the particular index.

tret **Total Returns Arrays** **float**

tret is a set of time series containing the return including all distributions of the index or portfolio each period. See index methodology for the weighting and derivation of the index.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible tret time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if tret is not available for the particular index.

aret **Capital Appreciation Arrays** **float**

aret is a set of time series containing the return excluding dividends of the index or portfolio each period. See index methodology for the weighting and derivation of the index.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible aret time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if aret is not available for the particular index.

iret **Income Returns Arrays** **float**

iret is a set of time series containing the income return of the index or portfolio each period. Income return is defined as the difference between the total return and the capital appreciation. See index methodology for the weighting and derivation of the index.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible iret time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if iret is not available for the particular index.

tind **Total Return Index Levels Arrays** **float**

tind is a set of time series containing the index level based on returns including all distributions of the index or portfolio each period. See index methodology for the base date and value.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible tind time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if tind is not available for the particular index.

aind **Capital Appreciation Index Levels Arrays** **float**

aind is a set of time series containing the index level based on returns excluding dividends of the index or portfolio each period. See index methodology for the base date and value.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible aind time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if aind is not available for the particular index.

iind **Income Return Index Levels Arrays** **float**

iind is a set of time series containing the index level based on income returns of the index or portfolio each period. Income returns are defined as the difference between total return and capital appreciation. See index methodology for the base date and value.

The variable indtypes contains the count of the result time series available for all indices in a set. There are 17 possible iind time series in group indices and 1 in series indices. Each time series has its own beginning and ending range, which is set to zero if iind is not available for the particular index.

3. CRSP BROWSE AND REPORTING TOOLS

CRSPAccess97 contains several tools that allow users to extract data from CRSP Stock and Indices databases. These command-line utilities require no programming, although they do require an understanding of CRSP data and identifiers. All utilities are run on a terminal window or command prompt window and are operated by entering instructions at a keyboard. Users can prepare input and in some cases must provide files with security input and specifications.

The following data utilities are available:

- *ts_print* time series report writer
- *stk_print* stock database report writer
- *dsxport* and *msxport* portfolio programs
- *dstksearch* and *mstksearch* namelist search utilities
- *dindsearch* and *mindsearch* index header search utilities

Report writers require securities selected by supported identifiers. The CRSP `permno` is the primary security identifier and must be used in *ts_print*. Other utilities support secondary identifiers such as header CUSIP, historical CUSIP, header ticker, `permco`, or SIC code. The *stksearch* utilities can be used to find identifiers using company name or other text data.

See Section 2 for stock and index data item descriptions and codes. The **CRSPAccess97 Indices File Guide** contains complete descriptions of additional index series and portfolio types available for Indices subscribers.

Appendix C includes information on system installation and additional available tools that can be used to manipulate stock and index databases.

Follow the Technical Support link through www.crsp.com for software updates of the browse and reporting tools described in this section.

3.1 *ts_print* Time Series Report Writer

ts_print is a report-writer program for CRSP Access97 data that generates report-style columnar text report files with raw and derived CRSP stock and indices data. *ts_print* is a command line executable program and enables users to control all of the specifications of reports without requiring previous programming knowledge. To maximize the potential of *ts_print*, the user should have a general familiarity of the data and the structure of their product.

ts_print is designed to facilitate data use by using CRSP timeseries data for raw and derived data types together with selected calendars. The input file is a text file that contains the specifications for the output data file or report. The output data file is in a delimited tabular text format and can either be used as is, or imported into spreadsheet or database programs for further processing.

Time series data is three-dimensional. Each data point refers an issue/index (`ENTITY`), a data item value (`ITEM`), and a date (`DATE`). *ts_print* allows the user to define these three components and to control the appearance of the output (`OPTIONS`), the fourth component. An input specification file controlled by the user determines the input and the output content and format.

ts_print is particularly useful for producing customized data sets for portfolio analysis and event studies. Time series data can be converted to different calendar frequencies, and header and event items can be mapped to a value at each period in a time series.

This section contains the following information:

- How to run *ts_print*
- Sample *ts_print* input files and output data
- Detail on creating the input file
- Tables containing the supported daily and monthly data items

ts_print is not designed for use with financial data other than data available through CRSP.

Running *ts_print*

ts_print runs in your command line terminal window.

At the command line, type `ts_print` and your input file name. The sample below would run the input file, `sample.txt`, created by the user. The file name can include a full path if it is not in the current directory. Note that the filenames must all be in lower case when using case-sensitive systems such as UNIX.

i.e. >`ts_print sample.txt` (enter)

A file name can be entered as the third parameter on the command line. If only two parameters are given , *ts_print* looks in the `OPTIONS` component of the input file for a file name. When the prompt returns, *ts_print* has generated the specified output file. If it returns an error message, revise your input file accordingly. The content of the error message should indicate what component you will need to revise to run with *ts_print*. The files default to the current directory that you are working in. If you want them in a specific directory, you will need to either work from that directory, specify file names with full path information, or move the files when you are done. For further data manipulation, these files can be edited with a text editor, or imported into a statistical package, a spreadsheet, or a database environment.

The input file below, `sample.txt`, will retrieve price, volume, unadjusted return and shares for `permno` 12490 (IBM Corporation, Inc.), using the daily file, to be reported on a monthly basis between January, 1995 to January, 1996 in the output file, `finsamp.out`.

`sample.txt`

```
ENTITY
LIST|PERMNO 12490
END
ITEM
ITEMID prc
ITEMID vol
ITEMID ret
ITEMID shr
END
DATE
CALNAME monthly|RANGE 19950101-19960101|CALFORMAT 4
END
OPTIONS
X ITEM,YES|Y DATE,YES|Z ENTITY,YES,1
OUTNAME finsamp.out|REPNAME Sample One
END
```

The following report is `finsamp.out`, the output from the input file, `sample.txt`, above. Keywords `REPNAME` (report title), `ENTITY`, `DATE` and `ITEMS` were added manually, and the `ENTITY (permno 12490)` and the `ITEMS (PRC, VOL, RET & SHR)` were bolded.

Sample One—`REPNAME` (report title)

	DATE	ITEMS			
ENTITY	**12490**	**PRC**	**VOL**	**RET**	**SHR**
	01|31|1995	72.12500	52494701	-0.018707	587172
	02|28|1995	75.25000	35186800	0.046835	587172
	03|31|1995	82.12500	63946402	0.091362	587172
	04|28|1995	94.62500	55918303	0.152207	584225
	05|31|1995	93.00000	64987300	-0.014552	580669
	06|30|1995	96.00000	66933901	0.032258	580669
	07|31|1995	108.87500	75667903	0.134115	580664
	08|31|1995	103.37500	70106402	-0.048351	568723
	09|29|1995	94.50000	97291801	-0.085852	568723
	10|31|1995	97.25000	82777901	0.029101	568723
	11|30|1995	96.62500	55654600	-0.003886	568723
	12|29|1995	91.37500	59581801	-0.054334	558315

Following are additional sample *ts_print* input and output files.

Sample *ts_print* Input and Output Files

There are three examples of input files that are provided with the data. They are provided in the `CRSP_LIB` directory and can be run directly if the correct product is available. The examples can be copied and used as templates for other reports. The examples assume a daily NYSE/AMEX/Nasdaq database. To run the first example, `ts_samp1.txt`, use the following command, dependent on your system:

`ts_print %crsp_lib%ts_samp1.txt` (Windows NT)

`ts_print $CRSP_LIB/ts_samp1.txt` (Unix)

`ts_print CRSP_LIB:ts_samp1.txt` (OpenVMS)

This produces an output file named `ts_samp1.dat` in the default directory. It produces monthly event output for two issues: IBM and Digital Equipment, each based on a different event date. The output data items are the calendar date, `permno`, last price, compounded return, and total volume for each month 2 months before until 1 months after each issue's event date. The output has the issues across, the dates down, with a separate table for each data item.

Input file: `ts_samp1.txt`

```
ENTITY
LIST|PERMNO 12490|EVDATE 19700101
LIST|PERMNO 43916|EVDATE 19800101
END
ITEM
ITEMID caldt|SUBNO 0
ITEMID permno|SUBNO 0
ITEMID prc|SUBNO 0
ITEMID ret|SUBNO 0
ITEMID vol|SUBNO 0
END
DATE
CALNAME monthly|RELATIVE –2,1
END
OPTIONS
X ENTITY|Y DATE|Z ITEM,3|OUTNAME ts_samp1.dat|REPNAME Sample1|DEFAULT 1|
END
```

The Output file**, `ts_samp1.dat`,** contains

```
Sample1
              12490     43916
Caldt   -2     19691128   19791130
Caldt   -1     19691231   19791231
Caldt    0     19700130   19800131
Caldt    1     19700227   19800229

Permno  -2        12490      43916
Permno  -1        12490      43916
Permno   0        12490      43916
Permno   1        12490      43916

Prc     -2    357.00000   67.62500
Prc     -1    364.50000   68.87500
Prc      0    335.25000   72.87500
Prc      1    340.25000   75.37500

Ret     -2    -0.006962   0.095142
Ret     -1     0.021008   0.018484
Ret      0    -0.080247   0.058076
Ret      1     0.018510   0.034305

Vol     -2       417000    1614200
Vol     -1       578300    1292700
Vol      0       789600    3798600
Vol      1       615100    2202200
```

The second sample, `ts_samp2.txt`, uses a predefined input file of `permno`s and event dates named `ts_list.txt`. The input file must be copied to the default directory from `CRSP_LIB` before running. The sample produces an output file named `ts_samp2.dat` in the default directory. It produces daily event output for the issues in the input file, with items across, dates and `permno`s down. `ts_samp1.txt` and `ts_samp2.txt` are 2 ways of accessing the same data; one by listing individual `permnos` and event dates in the report file and the other by putting both in a secondary input file. The output files look slightly different, since the order the data is requested and the report layout vary.

The input file **`ts_samp2.txt`** contains

```
ENTITY
LIST|FILE ts_list.txt,F2DL;PED1
END
ITEM
ITEMID caldt|SUBNO 0
ITEMID permno|SUBNO 0
ITEMID ret|SUBNO 0
ITEMID prc|SUBNO 0
ITEMID vol|SUBNO 0
END
DATE
CALNAME daily|RELATIVE –2,1
END
OPTIONS
X ITEM|Y DATE|Z ENTITY,3|OUTNAME ts_samp2.dat|REPNAME Sample2|DEFAULT 1
END
```

The sub-input file **`ts_list.txt`** contains

```
12490;19700101
43916;19800101
```

The output file **`ts_samp2.dat`** contains

```
Sample2

           Caldt      Permno     Ret        Prc        Vol

12490 -2    19691230      12490   0.003489      359.5      19400
12490 -1    19691231      12490   0.013908      364.5      29200
12490  0    19700102      12490   0.000686     364.75      15800
12490  1    19700105      12490   0.009596     368.25      21200

43916 -2    19791228      43916   0.001848      67.75      32200
43916 -1    19791231      43916   0.016605     68.875      17200
43916  0    19800102      43916   -0.04719     65.625      45800
43916  1    19800103      43916   -0.00952         65     265800
```

The third sample, `ts_samp3.txt`, prints quarterly capital appreciation returns for a single issue plus returns on the Standard and Poors 500 Composite.

Input file **ts_samp3.txt** contains

```
ENTITY
LIST|PERMNO 43916
INDEX|INDNO 1000502
END
ITEM
ITEMID retx|SUBNO 0
END
DATE
CALNAME quarterly|RANGE 19930101-19951231
END
OPTIONS
X ENTITY|Y DATE|Z ITEM,3|OUTNAME ts_samp3.dat|REPNAME Sample3|DEFAULT 1|
END
```

Output file **`ts_samp3.dat`** contains

```
Sample3
                43916      1000502
Ret 19930331     0.285185   0.036630
Ret 19930630    -0.037464  -0.002524
Ret 19930930    -0.119760   0.018645
Ret 19931231    -0.068027   0.016386
Ret 19940331    -0.135036  -0.044335
Ret 19940630    -0.341772  -0.003365
Ret 19940930     0.358974   0.041506
Ret 19941230     0.254717  -0.007434
Ret 19950331     0.139098   0.090230
Ret 19950630     0.075908   0.087955
Ret 19950929     0.119632   0.072804
Ret 19951229     0.405479   0.053935
```

The four components required to be in the input file, `ENTITY`, `ITEM`, `DATE`, and `OPTIONS`, are each detailed in the following section.

Creating the Input File

It is necessary to create an input text file to run *ts_print*. The input file contains the data specifications and controls the appearance of the report or output data file. Each input file must contain each of the following four components: `ENTITY`, `ITEM`, `DATE` and `OPTIONS`.

1. **`ENTITY`** –One or more issues assigned a CRSP `permno`, or a precalculated CRSP supported index.
2. **`ITEM`** – One or more *ts_print* supported CRSP timeseries variables (data items). These can have possible translations from one periodicity (time period of aggregation) to another.
3. **`DATE`** - Dates can be a set of `YYYYMMDD` ranges or a set of relative days to be matched against event dates supplied with each entity.
4. **`OPTIONS`** – controls the appearance and name of the output file.

Input File Rules

- Names in uppercase `COURIER` are keywords and must be typed "as is"
- # represents a number to be supplied by the user
- # . . .represents a list of numbers, which can be a single number, a range of numbers separated by a dash, or any number of these two types separated by commas. For example, 1, 3-5, 7-9, 15 represents the numbers l ,3,4,5,7,8,9,15.
- Z represents an alphanumeric character to be supplied by the user
- names in lowercase `courier` are replaced by the user. For example, a filename is replaced by the name of a user's file.
- Anything in brackets is optional. If names in brackets are used, all punctuation in the brackets are required. The brackets do not appear in the input file.
- Two or more keywords on a line must be separated with the `|(pipe)` character. Information specifying a keyword must be on the same line as the keyword. Additional keywords can also be placed on multiple lines; in this case the first line does not end in a pipe character.

ts_print is case sensitive. The user must follow the notational conventions, provided in this section, when creating the input file. The input file should be a text file with a `*.txt` extension. We recommend that the user not work with the input file in both windows-based and command line text editors, as they may appear to have different layouts.

The following section details each of the four components, `ENTITY`, `ITEM`, `DATE`, and `OPTIONS` and the keywords available for each.

CRSP recommends creating and editing the specifications file on the same system you intend to run it in. PC text editors inssert carriage return characters at the end of lines which may not be readable on UNIX or OpenVMS systems. Users should use the ASCII format when using FTP to move files between systems.

Each component entry, numbered below, is comprised of three parts:

- A heading row which identifies the component,
- The center row(s) which detail(s) the desired function(s) of the component, and
- The END row, which closes the component input information. A basic example follows:

sample.txt

```
1.   ENTITY
     LIST|PERMNO 12490
     END
2.   ITEM
     ITEMID prc
     ITEMID vol
     ITEMID ret
     ITEMID shr
     END
3.   DATE
     CALNAME monthly|RANGE 19950101-19960101|CALFORMAT 4
     END
4.   OPTIONS
     X ITEM,YES|Y DATE,YES|Z ENTITY,YES,1
     OUTNAME finsamp.out|REPNAME Sample One
     END
```

In *ts_print*, ENTITY, ITEM and DATE identify what your report will contain, while OPTIONS determines how your report will appear.

Explanation of Input File: sample.txt

1. In the sample layout above the ENTITY is for one issue, permno 12490.

2. Under ITEM, the report will contain price, volume, return, and share information for the ENTITY, with permno 12490.

3. In this sample, DATE specifies that the report will contain one value each month for each ENTITY and ITEM (CALNAME). Note the source of the ITEMs selected above is the Daily Stock File. Thus, the monthly value for Daily ITEMs is a monthly summary of the selected daily data item. In this case, prc and shr are prices and shares at month-end, vol is the sum of volumes during the month, and ret is the compounded daily return during the month (dividends are reinvested on exdate), reported between January 1, 1995 and December 31, 1995. Each date will be in a MM|DD|YYYY calendar format (CALFORMAT 4).

4. The OPTIONS selected contain X, Y and Z axes. ITEM options will be displayed on the X-axis, the DATE options on the Y axis and the entities will append themselves to the date or Y axis. (This is indicated by the number 1 at the end of the Z options.) The YES in each of the axis groups indicates that the report will contain axes headers finsamp.out (OUTNAME)is the name of the output file and Sample One (REPNAME) is the report title in the output file.

ENTITY Specifications

There are two ways to describe entities:

1. LIST identifies one or more selected issues. These may be specified by individual permnos on one or more rows, by a subroutine or secondary input file, or by ALL, which runs for all permnos available in the CRSP database.

2. An INDEX describes a precalculated index series supported by CRSP, specified by INDNO.

Each component entry is comprised of three parts:

- A heading row which identifies the component,
- The center row(s) which detail(s) the desired function(s) of the component, and
- The END row, which closes the component input information. A basic example follows:

The ENTITY section of the input file must begin with a heading or label, ENTITY. The center row(s) must contain the LIST(s) or the INDEX that the report is being run for, and the last row of the ENTITY section must be the word END.

Heading Row
```
ENTITY
```

Center Row(s):
```
LIST|PERMNO # or LIST|FILE filename or LIST|ALL
```
Or
```
INDEX|INDNO #
```
(Additional options described below can be added to each line with the following syntax of entity options:)
```
|EVDATE #|USERHEAD text
```

End Row:
```
END
```
Following are portions of 2 examples provided earlier in this section, which demonstrate the 3 primary ways to set up the ENTITY component of your input file. The first pulls data for both a permno and an indno. The second pulls data using an input file of desired permnos and specific event dates.

```
e.g.  ENTITY
      LIST|PERMNO 43916
      INDEX|INDNO 1000080
      END
```

```
e.g.  ENTITY
      LIST|FILE ts_list.txt,F2DL;PED1
      END
```

sub-input file ts_list.txt contains

```
10107;19900101
12490;19700101
14593;19850101
43916;19800101
```

ENTITY Keywords and Usage

The capitalized words need to be used as is. The lowercase words and symbols are indicative of user-specified information.

LIST – Indicator that for each use, a single permno or file containing permnos will be used to identify an ENTITY.

PERMNO # - one CRSP permno, (permanent and unique CRSP numbers assigned to individual issues) of an issue. # is the actual permno. Search functions and programs in CRSPAccess97 can be used to identify permnos.

FILE filename – (input permno file) Indicator that a subroutine or secondary file containing permnos (required) and dates (optional) will be called. Filename, as shown in the above format, is replaced with the actual name of your input file (permno input file).

The character layout of the subroutine used must include the following 2-character code specifications in the center row following LIST|filename of your input file.

Fl - formatted with each code followed by (begpos,endpos); begpos is the first character position in the input permno file that contains the data for that specification, endpos the last. Therefore, if your input file contains permnos in the first 5 character spaces, followed by the beginning date (D1) starting in the 7th character position and end date (D2) starting in the 13th character position of data desired for each permno, your ENTITY entry would look like this:

```
e.g. ENTITY
     LIST|FILE permin.txt,F1PE(1,5)D1(7,12)D2(13,18)
     END
```

F2 - is delimited and requires the DL, or delimiting character indicator, between fields

```
e.g. ENTITY
     LIST|FILE permin.txt,F2DLSPE
     END
```

DL - delimiter character used with F2. *ts_print* supports delimiters: P for pipe, S for space (DLP, DLS), and any other character can be used by adding a character on after DL (DL, for comma delimited input).
PE - permno
D1 - beginning date or event date
D2 - ending date
SD - to supply text for the permno as a header

ALL - all permnos in relevant databases will be used. Relevant databases will be determined by the data items (daily or monthly) selected. When this option is used, issued with no data inside the selected date range are ignored.

INDEX - Indicator that the following ENTITY is one of CRSP's precalculated indices.

INDNO - (index number) permanent identifier of an index. There are several standard indices included with CRSPAccess97 Stock databases. Your product mix will determine which indices are available. Appendix D has a list of INDNOs available in each product. Additional indices, compatible with CRSPAccess97, are available in the CRSPAccess97 Indices and Excess Returns product and described in the CRSPAccess97 Indices File Guide. The index search programs, *dindsearch* and *mindsearch*, can also be used to find available daily or monthly indices and their INDNOs.

EVDATE (event date) is the event date for a permno. EVDATE must be in the form YYYYMMDD. EVDATE is required if the calendar type in the DATE component is RELATIVE and is ignored otherwise.

USERHEAD (user description for header) is used to generate output headers (short descriptions) for the ENTITY, in the output file. The default header, the ENTITY permno or indno, is used unless the user specifies a USERHEAD string (up to 20 characters including spaces). Short description (SD) from a permno input file overrides the USERHEAD for supplying headers on an entity basis.

Note: All entities will have result data for the calendar date range specified in the DATE component of the input file for the data. If dates are selected outside this range in the D1, D2 or EVDATE Fields, the output data will contain missing value codes rather than values for the ITEMS selected.

Data Item Specification

A complete list of currently supported data items are listed in the *ts_print* Data Items Tables at the end of the *ts_print* section. They are organized alphabetically by ITEMID, and contain the following information:

- group identifier (GROUPID),
- a further breakdown of available items into SUBNOs,
- the default header for each ITEM as it appears in the output file,
- the default format and data type assigned to each ITEM and
- the ENTITY types the data ITEMs are compatible with.

Each ITEMID selected will generate one output for each ENTITY on each DATE.

There are daily and monthly sets of items. Monthly item names are prefixed with an **m**. A monthly item requires an available CRSPAccess97 monthly database and a daily item requires an available CRSPAccess97 daily database. If both databases are available, both daily and monthly items can be included in the same report. A monthly or daily indices-only database can be used with *ts_print*, but only INDEX entities and index-compatible items can be used.

The ITEM specification is comprised of three parts

- A heading row which identifies the component
- The center row(s) which detail(s) the desired function(s) of the component, and
- The END row, which closes the component input information. A basic example follows:

Heading Row:

```
ITEM
```

Center Row:

```
ITEMID text  or  GROUPID text
```

One of ITEMID or GROUPID must be chosen. Additional options can be added to each line with the following syntax:

```
|SUBNO #|SDESC text |FORMAT text|DATALEN #
```

A line with END is used after the last item is specified.

End Row:

```
END
```

Following is an example of a sample ITEM section. If you look up each of the data items in the Daily Item Table at the end of this section, you will see that of these data items, only prc (price) has 2 SUBNO choices - 0 last price and 1 last non-missing price. The caldt (calendar date), permno, ret (returns) and vol (volumes) do not have any SUBNOs listed since there is only one available, SUBNO 0, the default, for each. The INDNOs and PORTTYPES included in the CRSPAccess97 Stock File products are listed in the Data Item Keywords and Usage section following this one. Your product mix determines which of these are available. Additional indices and porttypes are available when using the CRSPAccess97 Stock File Data in conjunction with the CRSPAccess97 Indices File Data. port requires a PORTTYPE SUBNO. In this case it is chosen to select the NYSE/AMEX/Nasdaq Capitalization Portfolio assignment.

```
e.g.  ITEM
      ITEMID caldt|SUBNO 0
      ITEMID permno|SUBNO 0
      ITEMID ret|SUBNO 0
      ITEMID prc|SUBNO 0
```

```
ITEMID vol|SUBNO 0
ITEMID port|SUBNO 1
END
```

Data Item Keywords and Usage

Additional details for each of the ITEMs, can be found in the Daily and Monthly File Item Tables at the end of the chapter. Please refer to this table when creating your input file.

ITEMID - (item or variables selected) Code for the specific data item, ITEMID, requested. The data ITEMS are variables available in the database. Please refer to the variable descriptions (data definitions) for a more extensive definition.

ITEMIDs contains secondary or sub-levels. Secondary code refines the data request for specific data ITEMS. There are three secondary types of information; SUBNO Flags to provide missing price options in the output file, INDNO to identify the index to use for the required data ITEM, and PORTTYPE to identify the portfolio to be associated with the data request. Each of these categories is identified in the SUBNO column of the Daily and Monthly File Item Tables at the end of this section as follows:

— **SUBNO** numbers (0, 1, 2) are used to select on eof multiple variations of a derived ITEMID. SUBNO 0 is the default for all ITEMIDs and does not have to be explicitly included in the input file. For use definition, see the SUBNO column of the Daily and Monthly File Item Tables at the end of this section.

The Daily and Monthly File ITEM Tables at the end of this section list possible SUBNOs for each ITEM. If the table indicates PORTTYPE or INDNO there is only one variation and SUBNO is used to pick a specific Portfolio type or index (see INDNO and PORTTYPE below).

— **INDNO** index series identifiers identify a single index series that will be used for comparison, such as for an excess return. These items have indno in the SUBNO column of the Daily and Monthly File Item Tables. An item indno requires qualification of the item with the SUBNO option followed by the INDNO number of an associated index series.

There are several standard indices included with CRSP Access97 Stock databases. Additional indices, compatible with CRSPAccess97 stock databases, are available in the CRSPAccess97 Indices product. A search with the *dindsearch* and *mindsearch* utilities can be used to find available daily or monthly indices and their INDNOs.

Below is a partial list of index series INDNOs. See Appendix D for a list of INDNOs available with each stock product, and see the CRSPAccess97 Indices File Guide for a complete list of available series.

Index No.	Universe Coverage
1000081	NYSE/AMEX/Nasdaq Equal-Weighted Market Index
1000080	NYSE/AMEX/Nasdaq Value-Weighted Market Index
1000040	NYSE/AMEX Value-Weighted Market Index
1000041	NYSE/AMEX Equal-Weighted Market Index
1000502	S&P 500 Composite Index
1000503	Nasdaq Composite Index

— **PORTTYPE** - index series items use portfolio type numbers to qualify the data item. The portfolio type identifies a specific portfolio group, where a portfolio assignment or statistic for an issue each time period is used independently or as a link to a set of portfolio indices based on the portfolio group. The items this applies to have porttype in the SUBNO column of the Daily and Monthly File Item Tables. Porttypes require a SUBNO # between 1-9 to run.

Below is a partial list of portfolio types available for daily and monthly databases. See Appendix D for a list of portfolio types available with each stock product, and see the CRSPAccess97 Indices User's Guide for a complete list of available types.

Portfolio Types

Port #	Description
1	NYSE/AMEX/Nasdaq Capitalization
2	NYSE/AMEX Capitalization

GROUPID - Code for a grouping of specific data items. If a group is selected all the items in the group are selected. At the end of the Data ITEMs section, there is a list, alphabetical by group, which provides the GROUPID with associated ITEMIDs and a brief description for each.

SDESC - (ITEM-header description) a short description for headers to override the default. The defaults are listed in the Daily and Monthly File Item Tables at the end of this section. If a group, this will be used for all items if specified. This is user-defined and not in the File Items Tables. This field may contain up to the maximum number character spaces allocated for in the format of each item (see the Daily & Monthly File ITEM Tables). This allows the user to define the names or headings for the items.

FORMAT – (report formatting) allows you to modify the formatting assigned to each of the data items. The user may not change the data type, but may change the length of spaces and justification (right or left). FORMAT must contain: a % sign followed by the field width then the data type. Data types include: d for integers, f for floating point numbers, s for character strings, lf for double precision numbers. Floats and doubles can have a decimal point followed by a second number to indicate values after a decimal point. Any field can have a dash after the % sign to left justify the output in the field. Examples are %11.6f to print the output in an 11 character field with an explicit decimal point in the 5th position, or %-5s to print a left justified character field with a width of five. The FORMAT field allows you to change the character length and justification, but not the data type.

DATALEN - (columnar data length) the number of characters needed to store the output data to override the default. This should be at least as large as any field width specified in the format. This field should be modified when you wish to assign the field a header, which does not fit within the default FORMAT for the ITEM.

The following 2 pages contain the available daily and monthly data items, and group options, organized by group. See the data item tables (end of section) for a complete description of the data items for the input file.

Note that when using a GROUPID, the SUBNOs/INDNOs/PORTTYPEs must be compatible. Use the Data ITEMs table to verify compatibility.

The data length has been set to produce an output file, which is easily readable. If you are importing the data into another program for additional data manipulation, you may need to change the DATALEN (data length) field. This is particularly true with the character fields. The non-character fields may add spaces to the total allocated. If this occurs, use the FORMAT field to correct the total spaces for importing.

Daily File Items, Arranged by Group

groupid	itemid	Item Description
adjdata	adjprc	adjusted closing price or negative bid/ask average
	adjaskhi	adjusted closing ask or high
	adjbidlo	adjusted closing bid or low
	adjshr	adjusted shares outstanding in 1000's
	adjvol	adjusted trading volume
	adjask	adjusted closing ask
	adjbid	adjusted closing bid
caldt	caldt	yyyymmdd trading date
cumindrets	cumindiret	cumulative index income returns
	cumindtret	cumulative index total returns
	cumindaret	cumulative index returns without dividends
cumprets	cumparet	cumulative capital appreciation from first day in range of assigned portfolio of this type
	cumpiret	cumulative income return from first day in range of assigned portfolio of this type
	cumptret	cumulative total return from first day in range of assigned portfolio of this type
cumrets	cumaret	cumulative capital appreciation from first day in range
	cumiret	cumulative income return from first day in range
	cumtret	cumulative total return from first day in range
cumxsprets	cumxsparet	cumulative excess capital appreciation from first day of range between issue and current portfolio of this type
	cumxspiret	cumulative excess income return from first day of range between issue and current portfolio of this type
	cumxsptret	cumulative excess total return from first day of range between issue and current portfolio of this type
cumxsrets	cumxsaret	cumulative capital appreciation excess return between issue and index
	cumxsiret	cumulative excess income return between issue and index
	cumxstret	cumulative excess total return from first day of range between issue and index
ddata	ret	total return compounded daily return of period
	askhi	highest trade or closing ask
	bidlo	lowest trade or closing bid
	prc	last closing price or negative bid/ask average
	vol	cumulative daily volume
dists	divamt	total dividend amount in period
	facpr	cumulative factor to adjust prices in period
	odivamt	total ordinary dividend amount in period
headid	compno	Nasdaq company number
	cusip	header CUSIP
	permco	CRSP permco or indco if index
	permno	CRSP permno or indno if index
inddata	cap	effective (end of previous period) capitalization of issue or index
indres	aind	capital appreciation index level (if an issue, based on 100.0 on first period of price range)
	iind	income return index level (if an issue, based on 100.0 on first period of price range)
	tind	total return index level (if i an issue, based on 100.0 on first period of price range)

groupid	itemid	Item Description
indrets	indaret	capital appreciation of associated selected index compounded over period
	indiret	income return of associated selected index compounded over period
	indtret	total return of associated selected index compounded over period
indstat	cnt	count in index valid current and previous day
names	comnam	company name at end of period
	exchcd	exchange code at end of period
	ncusip	CUSIP at end of period
	shrcd	share code at end of period
	shrcls	share class at end of period
	siccd	SIC code at end of period
	ticker	ticker at end of period
nasdin	mmcnt	Nasdaq status code at end of period
	nmsind	Nasdaq National Market indicator at end of period
	nsdinx	NASD index at end of period
	trtscd	Nasdaq status code at end of period
portrets	portaret	capital appreciation of assigned portfolio of selected type compounded over period
	portiret	income return of assigned portfolio of selected type compounded over period
	porttret	daily total return of assigned portfolio of selected type compounded over period
ports	port	portfolio assignment in selected portfolio type at end of period
	portstat	statistic at end of current period for portfolio of this type
portxsrets	portxsaret	excess capital appreciation returns over period vs. assigned portfolio of selected type
	portxsiret	excess income returns over period vs. assigned portfolio of selected type
	portxstret	excess total returns over period vs. assigned portfolio of selected type
qdata	ask	closing ask
	bid	closing bid
	numtrd	cumulative number of trades
rdata	reti	income return
	retx	capital appreciation return - compounded daily return of period
sdata	high	highest closing trade or bid/ask average in period
	low	lowest closing trade or bid/ask average in period
	volavg	average daily volume
	volmed	median daily volume
shares	shr	shares outstanding in 1000's at end of period
xsrets	xsaret	excess compounded index series capital appreciation index returns over period vs. selected index series
	xsiret	excess compounded income returns over period vs. selected index series
	xstret	excess compounded total returns over period vs. selected index series

Monthly File Items, Arranged by Group

groupid	itemid	Item Description
madjdata	madjprc	adjusted closing price or negative bid/ask average
	madjaskhi	adjusted closing ask or high
	madjbidlo	adjusted closing bid or low
	madjshr	adjusted shares outstanding in 1000's
	madjvol	adjusted trading volume
	madjask	adjusted closing ask
	madjbid	adjusted closing bid
mcaldt	mcaldt	yyyymmdd trading date
mcumindrets	mcumindiret	cumulative index income returns
	mcumindtret	cumulative index total return
	mcumindaret	cumulative index returns without dividends
mcumprets	mcumparet	cumulative capital appreciation from first day in range of assigned portfolio of this type
	mcumpiret	cumulative income return from first day in range of assigned portfolio of this type
	mcumptret	cumulative total return from first day in range of assigned portfolio of this type
mcumrets	mcumaret	cumulative capital appreciation from first day in range
	mcumiret	cumulative income return from first day in range
	mcumtret	cumulative total return from first day in range
mcumxsprets	mcumxsparet	cumulative excess capital appreciation from first day of range between issue and current portfolio of this type
	mcumxspiret	cumulative excess income return from first day of range between issue and current portfolio of this type
	mcumxsptret	cumulative excess total return from first day of range between issue and current portfolio of this type
mcumxsrets	mcumxsaret	cumulative excess capital appreciation between issue and index of portfolio assignment
	mcumxsiret	cumulative excess income return between issue and index of portfolio assignment
	mcumxstret	cumulative excess total return from first day of range between issue and index
mdata	mret	total return compounded monthly return of period
	maskhi	highest trade or closing ask
	mbidlo	lowest trade or closing bid
	mprc	last closing price or negative bid/ask average
	mvol	cumulative monthly volume
mdists	mdivamt	total dividend amount in period
	mfacpr	cumulative factor to adjust prices in period
	modivamt	total ordinary dividend amount in period
mheadid	mcompno	Nasdaq company number
	mcusip	header CUSIP
	mpermco	CRSP permco or indco if index
	mpermno	CRSP permno or indno if index
minddata	mcap	effective (end of previous period) capitalization of issue or index
mindres	maind	capital appreciation index level (if an issue, based on 100.0 on first period of price range)
	miind	income return index level (if an issue, based on 100.0 on first period of price range)
	mtind	total return index level (if an issue, based on 100.0 on first period of price range)

groupid	itemid	Item Description
mindrets	mindaret	capital appreciation of associated selected index compounded over period
	mindiret	income return of associated selected index compounded over period
	mindtret	total return of associated selected index compounded over period
mindstat	mcnt	count in index valid current and previous day
mnames	mcomnam	company name at end of period
	mexchcd	exchange code at end of period
	mncusip	cusip at end of period
	mshrcd	share code at end of period
	mshrcls	share class at end of period
	msiccd	sic code at end of period
	mticker	ticker at end of period
mnasdin	mmmcnt	Nasdaq status code at end of period
	mnmsind	Nasdaq National Market indicator at end of period
	mnsdinx	NASD index at end of period
	mtrtscd	Nasdaq status code at end of period
mportrets	mportaret	capital appreciation of assigned portfolio of selected type
	mportiret	income return of assigned portfolio of selected type
	mporttret	total return of assigned portfolio of selected type
mports	mport	portfolio assignment in selected portfolio type at end of period
	mportstat	statistic at end of current period for portfolio of this type
mportxsrets	mportxsaret	excess capital appreciation returns over period vs. assigned portfolio of selected type
	mportxsiret	excess income returns over period vs. assigned portfolio of selected
	mportxstret	excess total returns over period vs. assigned portfolio of selected type
mqdata	mask	closing ask
	mbid	closing bid
	mnmtrd	cumulative number of trades
mrets	mreti	income return
	mretx	capital appreciation return - compounded monthly return of period
msdata	mhigh	highest closing trade or bid/ask average in period
	mlow	lowest closing trade or bid/ask average in period
mrdata	mvolavg	average monthly volume
	mvolmed	median monthly volume
mshares	mshr	shares outstanding in 1000's at end of period
mxsrets	mxsaret	excess capital appreciation index returns over period vs. selected index series
	mxsiret	excess income returns over period vs. selected index series
	mxstret	excess total returns over period vs. selected index series

DATE Specification

The DATE input specifies the set of dates or ranges that will appear in the output file for the ITEMs selected. The calendar may be one of five calendars in the database: daily, weekly, monthly, quarterly, or annual. The ranges can be either the same for all input entities, or be based on an event date for each permno.

Each component is comprised of three parts

- A heading row which identifies the component
- The center row(s) which detail(s) the desired function(s) of the component, and
- The END row, which closes the component input information. A basic example follows:

Heading Row
```
DATE
```

Center Row
```
CALNAME text|RANGE or RELATIVE dates|CALFORMAT #
```

End Row
```
END
```

The calendar name and either an absolute or relative range must be chosen. The date specification keywords are described below.

ts_print contains a 'smart calendar', which allows the user to select the various calendar options with the various data items. For example a daily CALNAME with a monthly data file would return a daily range of data, using the month end value to fill in values for the days there is no additional data in the file. Likewise, a daily calendar with a monthly, quarterly or annual reporting frequency will utilize monthly, quarterly, and annual figures for the selected ITEMS.

Following are 2 examples. The first example will pull the daily or monthly items in the date range between January 1, 1990 and December 31, 1995. The calendar indicates the frequency of the data items selected for the report. The second example will report on a daily basis a total of 4 days, from 2 days before the event date, the event date (EVDATE), and 1 day after the event date. The event date is specified in the ENTITY specification section of your input file.

```
e.g.  DATE
      CALNAME quarterly|RANGE 19900101-19951231
      END

e.g.  DATE
      CALNAME daily|RELATIVE –2,1
      END
```

DATE Specification– Keywords

CALNAME - (calendar name) name of an existing calendar to report on. *ts_print* supports reporting for Daily, Weekly, Monthly, Quarterly and Annual Calendars. Either daily or monthly file data items can be used with any of the supported calendars. Input data frequency is determined by the data item specified in the ITEM section. The supported calendars must be chosen from the following table and set the frequency of reporting in the output file.

The following table contains a list of the calendars currently supported.

CALNAME	Calendar Description
daily	CRSP daily stock calendar
weekly	CRSP weekly stock calendar
monthly	CRSP monthly stock calendar
quarterly	CRSP quarterly stock calendar

annual	CRSP annual stock calendar

RANGE daterange - sets the fixed time period, with beginning and end dates, of the selected calendar that *ts_print* reports data. Ranges can be expressed as YYYY-YYYY, YYYYMM-YYYYMM, YYYYMMDD-YYYYMMDD, or YYMMDD-YYMMDD. If only a month or year is specified, all dates in the calendar belonging to that month or year will be included. If the chosen dates are not in the selected calendar, the beginning range uses the next following date in the calendar and the ending range uses the last previous date in the calendar. Output will be produced for all entities for all items for each period in the range. If the entity does not have data during the range or is restricted by date range in the ENTITY description section, missing values will be filled in.

RELATIVE daterange - sets the event time range of a report, and is used to select data for entities based on an entity-specific event date. Ranges are expressed as the first period relative to the event date followed by a comma and the last period relative to the event date. A range before the event date is indicated with a leading dash. A range on the event date is indicated as a 0.

The RELATIVE date is dependent on the EVDATE or the D1 setting in an input permno entity file. This option is typically used for event studies, when the data range sought for each permno is different. Using this option, RELATIVE –5,6, for example, would return results for the five reporting dates before the event date, the event date period, and the six reporting periods after the event date. This option does not work with a beginning and ending date. It is useful to include the ITEM caldt (mcaldt) in the output file when using relative dates.

CALFORMAT - (calendar format) a numeric code for the formatting of the dates in a range when date headers are chosen in the output options:

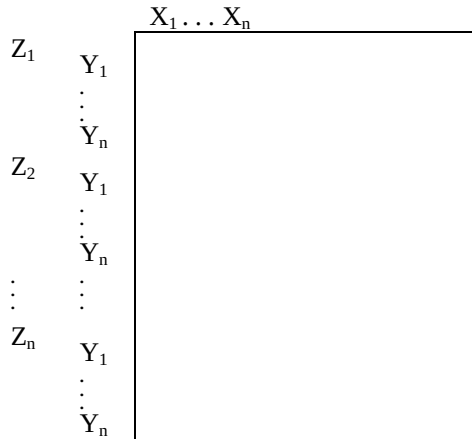
- 1 (default) =YYYYMMDD
- 2 =YYMMDD
- 3 =MM|DD|YY
- 4 =MM|DD|YYYY
- 5 =DD-MM-YYYY

OPTIONS and Output Specification

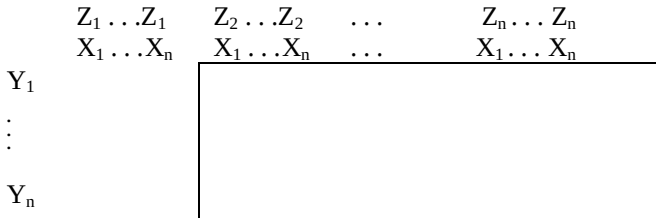
Each data point represents the data ITEM value for one ENTITY on a given DATE. These three points are plotted in a table to produce the output file. The OPTIONS component specifies the appearance of the output file.

Each of the three data dimensions, ITEM, ENTITY and DATE, are assigned by the user to the X, Y, or Z axis. The Z axis can be represented in two dimensions in one of the following three ways:

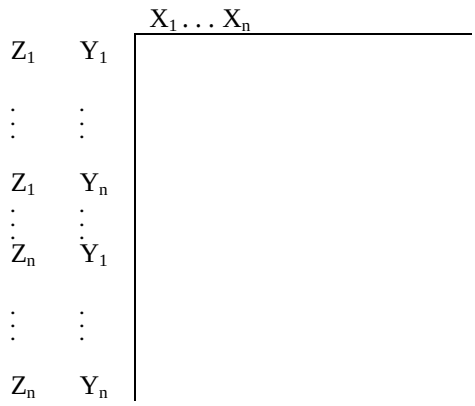
1. X and Y table are repeated for each Z.



2. On X axis, All X items for Z item 1, All X items for Z item 2 , ..., All X items for Z item N.



3. On Y axis, All Y items for Z item 1, All Y items for Z item 2 , ..., All Y items for Z item N. This is similar to the first option, but Z headers are available on each line.



These three dimension assignments are required in the OPTIONS component.

Each component is comprised of three parts

- A heading row which identifies the component
- The center row(s) which detail(s) the desired function(s) of the component, and
- The END row, which closes the component input information. A basic example follows:

Heading Row:

```
OPTIONS
```

Center Row:

```
X type[,headers]|Y type[,headers]|Z type[,headers], zflag#
OUTNAME filename|REPNAME text|FIELDDELIM text|BUFSIZE #|NOFILL
CHARDELIM text|ROWDELIM #,#|DEFAULT #
```

End Row:

```
END
```

The following example contains the required X, Y, and Z axes specifications. The ENTITY data will lie along the X axis, the DATE on the Y axis and Z will also lie along the y axis. The headers will all be displayed in the report, since they have been omitted in this example, but are the default for each axis specification. *ts_print* will generate an output file (OUTNAME) named ts_samp3.dat into the directory you are working in. The report will have a heading called Sample 6.

```
e.g. OPTIONS
     X ENTITY|Y DATE|Z ITEM,3|OUTNAME ts_samp3.dat|REPNAME Sample6
     END
```

Report OPTIONS Keywords and Usage

X axis, Y axis, and Z axis assignments are mandatory, and must allocate ENTITY, ITEM and DATE to the graphical axes.

— Header references must be included in the output file; YES, to show or NO, to hide. Header specification is included with each axis specification. The default is YES. The default header for an ENTITY is the permno for a security and indno for an index. The default header for a data ITEM is the item header in the Data Item Table at the end of the *ts_print* section. The default header for DATE is YYYYMMDD for calendar ranges and relative period numbers for relative dates.

— The Z-flag field is required. It is a number, 1, 2, or 3, as described in the diagrams on the previous page, which determines how the z axis data is printed in the two dimensional output.

— Performance for large datasets is greatly improved if ITEM is chosen for the X axis, DATE is chosen for the Y axis, ENTITY for the Z axis, and zflag# is set to 1 or 3.

OUTNAME - (output file name) is the name of the file where the output will be stored. Without this, the data will dump to the screen.

REPNAME - (report name) is a text description that will be placed at the top of the report (default is none).

NOFILL - rows outside an issue's date range or the user's date specification will not print to the output file. NOFILL is only applicable if ITEM is chosen for the X axis, DATE for the Y axis, and ENTITY for the Z axis, and zflag# is 1 or 3.

FIELDDELIM - (field delimiter-columnar) is a specified character string that will be placed as a delimiter between fields in output file rows (default is none).

BUFSIZE - (number of bytes) is the size of memory that will be allocated by the program. In a large study, the program will save intermediate data in a temporary file, and this can degrade performance severely. If memory is available on your system, you can use the BUFSIZE option to increase the size of the internal buffer. Do not use BUFSIZE for small reports. The program will report the necessary buffer size if a specific report requires more than the default limit. Switching axes can also be used to improve performance for large datasets.

CHARDELIM - (character delimiter-column) is a character string placed around all character string fields in output file rows (default is none). *ts_print* left justifies the character fields. This means fields that have a character format 's' or ITEMIDs – comnam/mcomnam, ticker/mticker, cusip/mcusip, or ncusip/mncusip, are left justified. If you expect to read the *ts_print* results as delimited data, it is recommended that you add this to your OPTIONS component when creating your input file.

ROWDELIM - (field delimiter-row) is the number of rows between output lines. The first is the number between tables when the Z dimension is a separate table (default is 1). The second is the number between data rows (default is 0). Use of this will look like this (#,#)

DEFAULT - (default settings) is a summary of output formats. It must be set to 1 if used. It can be used instead of the other output options. Each Z is a separate table. Field delimiter is a space. Within each table, rows are single-spaced. There are two rows between tables. All headers are printed. If default is chosen, the other output format options are ignored.

Note: The row detailing the functionality of a single option must wrap. Different keywords can be on separate lines, but the last keyword on a line cannot end with a pipe character, and the beginning of a line must be a keyword.

ts_print Daily File Items

itemid	groupid	subno	Item Header	Item Description	Format	Entity Types
adjask	adjqdata	0 - keep if last is missing 1 – use last non-missing if available	Adjask Adjaskprev	adjusted closing ask in period The last non-missing adjusted closing ask in the period	%11.5f	list
adjaskhi	adjddata	0 - keep if last is missing 1 – use last non-missing if available	Adjaskhi Adjaskhiprev	adjusted closing ask or high The last non-missing adjusted askhi in the period	%11.5f	list
adjbid	adjqdata	0 - keep if last is missing 1 – use last non-missing if available	Adjbid Adjbidprev	Adjusted closing bid The last non-missing adjusted bid in the period	%11.5f	list
adjbidlo	adjddata	0 - keep if last is missing 1 - use last non-missing if available	Adjbidlo Adjbidloprev	Adjusted closing bid or low The last non-missing adjusted bidlo in the period	%11.5f	list
adjprc	adjdata	0 - keep if last is missing 1 - use last non-missing if available	Adjprc Adjprcprev	Adjusted closing price or negative bid/ask average The last non-missing adjusted price in the period	%11.5f	list
adjshr	adjddata	0 - use all facshrs 1 - ignore facshr from rights	Adjshr Adjshrxr	Adjusted shares outstanding in 1000's Adjusted shares outstanding in 1000's, rights excluded	%9d	list
adjvol	adjddata		Adjvol Adjvolxr	Adjusted trading volume Adjusted trading volume, rights excluded	%9d	list
aind	indres		Aind	Capital appreciation index level (if an issue, based on 100.0 on first period of price range) without dividends	%11.5f	list index
ask	qdata	0 - keep if last is missing 1 - use last non-missing if available	Ask Askprev	Closing ask the last non-missing closing ask in the period	%11.5f	list
askhi	ddata	0 - keep if last is missing 1 - use last non-missing if available	Askhi Askhiprev	highest trade or closing askhi the last non-missing closing askhi in the period	%11 .5f	list
bid	qdata	0 – keep if last is missing 1 – use last non-missing if available	Bid Bidprev	closing bid the last non-missing closing bid in the period	%11.5f	list
bidlo	ddata	0 – keep if last is missing 1 – use last non-missing if available	Bidlo Bidloprev	lowest trade or closing bid the last non-missing closing bidlo in the period	%11.5f	list
caldt	caldt		Caldt	yyyymmdd trading date	%9d	list index
cap	inddata	0 – actual 1 – effective	Cap Cape	effective (end of previous period) capitalization of issue or index – for use with weighted values ending capitalization of issue or index	%1 5.5lf	list index
cnt	indstat	0 – actual 1 – effective	Cnt Cntprev	count in index valid current and previous day count in index valid current day	%6d	index
comnam	names	0 – actual 1 – effective 2 – last	Company Name Effective Name Last Compnay Name	Company name at end of period company name at beginning of period last company name	%-32.32s	list
compno	headid		Compno	Nasdaq company number	%8d	list
cumaret	cumrets		Cumaret	cumulative capital appreciation from first day in range without dividends	%11.6f	list index
cumindtret	cumindrets	indno – see Daily CRSP Index Series for indnos	Cumindtret	cumulative indices returns with dividends	%11.6f	list index
cumindaret	cumindrets	indno – see Daily CRSP Index Series for indnos	Cumindaret	cumulative indices returns without dividends	%11.6f	list index
cumindiret	cumindrets	indno – see Daily CRSP Index Series for indnos	Cumindiret	cumulative indices income returns	%11.6f	list index
cumiret	cumrets		Cumiret	cumulative income return from first day in range	%11.6f	list index
cumparet	cumprets	porttype - see Daily CRSP Portfolios for porttype numbers	Cumparet	cumulative capital appreciation from first day in range of assigned portfolio of this type without dividends	%11.6f	list
cumpiret	cumprets	porttype - see Daily CRSP Portfolios for porttype numbers	Cumpiret	cumulative income return from first day in range of assigned portfolio of this type	%11.6f	list
cumptret	cumprets	porttype - see Daily CRSP Portfolios for porttype numbers	Cumptret	cumulative total return from first day in range of assigned portfolio of this type with dividends	%11.6f	list
cumtret	cumrets		Cumtret	Cumulative active total return from 1st day in range	%11.6f	list index

ts_print Daily File Items *(Con't)*

itemid	groupid	subno	Item Header	Item Description	Format	Entity Types
cumxsaret	cumxsrets	indno of base index series	Cumxsaret	capital appreciation excess return between issue and index of portfolio assignment without dividends	%11.6f	list
cumxsiret	cumxsrets	indno of base index series	Cumxsiret	cumulative excess income returns	%11.6f	list
cumxsparet	cumxsprets	porttype - see Daily CRSP Portfolios for porttype numbers	Cumxsparet	cumulative excess capital appreciation from first day of range between issue and current portfolio of this type without dividends	%11.6f	list
cumxspiret	cumxsprets	porttype - see Daily CRSP Portfolios for porttype numbers	Cumxspiret	Cumulative excess income return from first day of range between issue and current portfolio of this type	%11.6f	list
cumxsptret	cumxsprets	porttype - see Daily CRSP Portfolios for porttype numbers 1 – effective 2 – last	Cumxsptret EXE EXL	Cumulative excess total return from first day of range between issue and current portfolio of this type with dividends exchange code at beginning of period last exchange code	%11.6f	list
divamt	dists		Divamt	total dividend amount in period	%11.6f	list
facpr	dists		Facpr	cumulative factor to adjust prices in period	%11.6f	list
high	sdata		High	highest closing trade or bid/ask average in period	%11.5f	list
iind	indres		Iind	income return index level (if an issue is based on 100.0 on first period of price range)	%11.6f	list index
indaret	indrets	indno of base index	Indaret	capital appreciation of associated selected index compounded over period without dividends	%11.6f	list index
indiret	indrets	indno of base index	Indiret	income return of associated selected index compounded over period	%11.6f	list index
indtret	indrets	indno of base index	Indtret	total return of associated selected index compounded over period with dividends	%11.6f	list index
low	sdata		Low	lowest closing trade or bid/ask average in period	%11.5f	list
mmcnt	nasdin	0 – actual 1 – effective 2 – last	Mmcnt Mmcnte Mmcntl	Nasdaq status code at end of period Nasdaq status code at beginning of period last Nasdaq status code	%4d	list
ncusip	names	0 – actual 1 – effective 2 – last	NCUSIP NCUSIPE NCUSIPL	CUSIP at end of period CUSIP at beginning of period last CUSIP	%-8.8s	list
nmsind	nasdin	0 – actual 1 – effective 2 – last	Nmsind Nmsinde Nmsindl	The Nasdaq National Market indicator at end of period The Nasdaq National Market indicator at beginning of period last Nasdaq National Market indicator	%2d	list
nsdinx	nasdin	0 - actual 1 – effective 2 - last	Nsdinx Nsdinxe Nsdinxl	nasd index at end of period nasd index at beginning of period last nasd index	%2d	list
numtrd	qdata		Numtrd	cumulative number of trades	%9d	list
odivamt	dists		Odivamt	total ordinary dividend amount in period	%11.6f	list
pemmco	headid		PERMCO	CRSP permco or indco if index	%8d	list index
permno	headid		PERMNO	CRSP permno or indno if index	%8d	list index
porttret	portrets	porttype – see Daily CRSP Portfolios for porttype numbers	Porttret	compounded daily total return of assigned portfolio of selected type with dividends	%11.6f	list
port	ports	porttype – see Daily CRSP Portfolios for porttype numbers	Port	portfolio assignment in selected portfolio type at end of period	%4d	list
portaret	portrets	porttype – see Daily CRSP Portfolios for porttype numbers	Portaret	Compounded capital appreciation of assigned portfolio of selected type	%11.6f	list
portiret	portrets	porttype – see Daily CRSP Portfolios for porttype numbers	Portiret	Compounded income return of assigned portfolio of selected type	%11.6f	list
portstat	ports	porttype – see Daily CRSP Portfolios for porttype numbers	Portstat	statistic at end of current period for portfolio of this type	%15.5lf	list
portxsaret	portxsrets	porttype - see Daily CRSP Portfolios for porttype numbers	Portxsaret	excess capital appreciation returns over period vs. assigned portfolio of selected type	%11.6f	list
portxstret	portxsrets	porttype - see Daily CRSP Portfolios for porttype numbers	Portxstret	excess total returns over period vs. assigned portfolio of selected type	%11.6f	list

ts_print Daily File Items *(Con't)*

itemid	groupid	Subno	Item Header	Item Description	Format	Entity Types
portxsiret	portxsrets	porttype – see Daily CRSP Portfolios for porttype numbers	Portxsiret	excess compounded income returns over period vs. assigned portfolio of selected type	%11.6f	list
prc	ddata	0 – last price	Prc	last closing price or negative bid/ask average in period	%11.5f	list
		1 – last non-missing price	Prcprev	last non-missing closing price or negative bid/ask average in period		
prcprevdt	prcprev		Prcprevdt	data of past non missing price in period	%8d	list
ret	ddata		Ret	total return compounded daily return of period, return and dividends	%11.6f	list index
reti	rdata		Reti	total income return -compounded daily, dividends only of the return period	%11.6f	list index
retx	rdata		Retx	capital appreciation return – compounded daily return without dividends of period	%11.6f	list index
shr	shares	0 – ignore facshr from rights	Shr	shares outstanding in 1000's at end of period	%9d	list
		1 – use all facshrs	Shrxr	shares outstanding in 1000's at end of period, rights excluded		
shrcd	names	0 – actual 1 – effective 2 – last	SC SCE SCL	share code at end of period share code at beginning of period last share code	%3d	list
shrcls	names	0 – actual 1 – effective 2 – last	CL CLE CLL	share class at end of period share class at beginning of period last share class	%-1.1s	list
siccd	names	0 – actual 1 – effective 2 – last	SIC SICE SICL	sic code at end of period sic code at beginning of period last sic code	%4d	list
ticker	names	0 – actual 1 – effective 2 – last	Ticker Tickere Tickerl	ticker at end of period ticker at beginning of period last ticker	%-5.5s	list
tind	indres		Tind	total return index level (if i an issue, based on 100.0 on first period of price range) with dividends	%11.5f	list index
trtscd	nasdin	0 – actual 1 – effective 2 - last	Trtscd Trtscde Trtscdl	Nasdaq status code at end of period Nasdaq status code at beginning of period last Nasdaq status code	%2d	list
vol	ddata		Vol	cumulative daily volume	%9d	list
volavg	sdata		Volavg	average daily volume	%9d	list
volmed	sdata		Volmed	median daily volume	%9d	list
xsaret	xsrets	indno of base index series	Xsaret	excess compounded index series capital appreciation index returns over period vs. selected index series without dividends	%11.6f	list
xsiret	xsrets	indno of base index series	Xsiret	excess compounded income returns over period vs. selected index series	%11.6f	list
xstret	xsrets	indno of base index series	Xstret	excess compounded total returns over period vs. selected index series with dividends	%11.6f	list

ts_print Monthly File Items0

Itemid	groupid	SUBNO	Item Header	Item Description	Format	Entity Types
madjask	madjqdata	0 - keep if last is missing 1 - use last non-missing if available	Adjask Adjaskprev	adjusted closing ask last non-missing adjusted closing ask in the period	%11.5f	list
madjaskhi	madjddata	0 - keep if last is missing 1 - use last non-missing if available	Adjaskhi Adjaskhiprev	adjusted closing ask or high last non-missing adjusted closing askhi in the period	%11.5f	list
madjbid	madjqdata	0 - keep if last is missing 1 - use last non-missing if available	Adjbid Adjbidprev	adjusted closing bid last non-missing adjusted closing bid in the period	%11.5f	list
madjbidlo	madjddata	0 - keep if last is missing 1 - use last non-missing if available	Adjbidlo Adjbidloprev	adjusted closing bid or low last non-missing adjusted closing bidlo in the period	%11.5f	list
madjprc	madjdata	0 - keep if last is missing 1 - use last non-missing if available	Adjprc Adjprcprev	adjusted closing price or negative bid/ask average last non-missing closing price in the period	%11.5f	list
madjshr	madjddata	0 - use all facshrs 1 - ignore facshr from rights	Adjshr Adjshrxr	adjusted shares outstanding in 1000's adjusted shares outstanding in 1000's excluding returns	%9d	list
madjvol	madjddata		Adjvol	adjusted trading volume	%9d	list
maind	mindres	indno of base index	Aind	capital appreciation index level (if an issue, based on 100.0 on first period of price range) without dividends	%11.5f	list index
mask	mqdata	0 - keep if last is missing 1 - use last non-missing if available	Ask Askprev	closing ask last non-missing closing ask in the period	%11.5f	list
maskhi	mdata	0 - keep if last is missing 1 - use last non-missing if available	Askhi Askhiprev	highest trade or closing askhi last non-missing closing askhi in the period	%11.5f	list
mbid	mqdata	0 - keep if last is missing 1 - use last non-missing if available	Bid Bidprev	closing bid last non-missing closing bid in the period	%11.5f	list
mbidlo	mdata	0 - keep if last is missing 1 - use last non-missing if available	Bidlo Bidloprev	lowest trade or closing bidlo last non-missing closing bidlo in the period	%11.5f	list
mcaldt	mcaldt		Caldt	yyyymmdd trading date at the end of the period	%9d	list index
mcap	minddata	0 - actual 1 – effective	Cap Cape	capitalization of issue or index (end of previous period) used with weighted values ending capitalization of issue or index	%15.51f	list index
mcnt	mindstat		Cnt	count in index valid current and previous day	%6d	index
mcomnam	mnames	0 – actual 1 – effective 2 – last	Company Name Effective Name Last Company Name	company name at end of period company name at beginning of period last company name	%-32.32s	list
mcompno	mheadid		COMPNO	Nasdaq company number	%8d	list
mcumaret	mcumrets		Cumaret	cumulative capital appreciation from first day in range	%11.6f	list Index
mcumindtret	mcumindrets	indno of base index series	Cumindtret	cumulative total returns of associated index from first period in range	%11.6f	list Index
mcumindaret	mcumindrets	indno of base index series	Cumindaret	cumulative returns without dividends of associated index from first period in range	%11.6f	list index
mcumindiret	mcumindrets	indno of base index series	Cumindiret	cumulative income returns of associated index from first day in range	%11.6f	list index
mcumiret	mcumrets		Cumiret	cumulative income return from first day in range	%11.6f	list index
mcumparet	mcumprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumparet	cumulative capital appreciation from first day in range of assigned portfolio of this type	%11.6f	list
mcumpiret	mcumprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumpiret	cumulative income return from first day in range of assigned portfolio of this type	%11.6f	list
mcumptret	mcumprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumptret	cumulative total return from first day in range of assigned portfolio of this type	%11.6f	list
mcumtret	mcumrets		Cumtret	cumulative total return from first day in range	%11.6f	list index

ts_print Monthly File Items *(Con't)*

Itemid	groupid	SUBNO	Item Header	Item Description	Format	Entity Types
mcumxsaret	mcumxsrets	indno of base index series	Cumxsaret	cumulative capital appreciation excess return between issue and index	%11.6f	list
mcumxsiret	mcumxsrets	indno of base index series	Cumxsiret	cumulative excess capital appreciation between issue and index	%11.6f	list
mcumxsparet	mcumxsprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumxsparet	cumulative excess capital appreciation from first day of range between issue and assigned portfolio of this type	%11.6f	list
mcumxspiret	mcumxsprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumxspiret	cumulative excess income return from first day of range between issue and current portfolio of this type	%11.6f	list
mcumxsptret	mcumxsprets	porttype - see Monthly CRSP Portfolios for porttype numbers	Cumxsptret	cumulative excess total return from first day of range between issue and current portfolio of this type with dividends	%11.6f	list
mcumxstret	mcumxsrets	indno of base index	Cumxstret	cumulative excess total return from first day of range between issue and index	%11.6f	list index
mcusip	mheadid		CUSIP	header cusip	%-8.8s	list
mdivamt	mdists		Divamt	total dividend amount in period	%11.5f	list
mexchcd	mnames	0 – actual 1 – effective 2 – last	EX EXE EXL	exchange code at end of period exchange code at beginning of period last exchange code	%2d	list
mfacpr	mdists		Facpr	cumulative factor to adjust prices in period	%11.6f	list
mhigh	msdata		High	highest closing trade or bid/ask average in period	%11.5f	list
miind	mindres		Iind	income return index level(if an issue, based on100.0 on first period of price range)	%11.6f	list index
mindaret	mindrets	indno of base index	Indaret	capital appreciation of associated selected index compounded over period	%11.6f	list
mindiret	mindrets	indno of base index	Indiret	income return of associated selected index compounded over period	%11.6f	list
mindtret	mindrets	indno of base index	Indtret	total return of associated selected index compounded over period	%11.6f	list
mlow	msdata		Low	lowest closing trade or bid/ask average in period	%11.5f	list
mmmcnt	mnasdin	0 – actual 1 – effective 2 – last	Mmcnt Mmcnte Mmcntl	Nasdaq status code at end of period Nasdaq status code at beginning of period last Nasdaq status code	%4d	list
mncusip	mnames	0 – actual 1 – effective 2 – last	NCUSIP NCUSIPE NCUSIPL	CUSIP at end of period CUSIP at beginning of period last CUSIP	%-8.8s	list
mnmsind	mnasdin	0 – actual 1 – effective 2 – last	Nmsind Nmsinde Nmsindl	The Nasdaq National Market indicator at end of period The Nasdaq National Market indicator at beginning of period last Nasdaq National Market indicator	%2d	list
mnsdinx	mnasdin	0 – actual 1 – effective 2 - last	Nsdinx Nsdinxe Nsdinxl	nasd index at end of period nasd index at beginning of period last nasd index	%2d	list
mnmtrd	mqdata		Numtrd	cumulative number of trades	%9d	list
modivamt	mdists		Odivamt	total ordinary dividend amount in period	%11.6f	list
mpemmco	mheadid		Permco	CRSP permco or indco if index	%8d	list index
mpermno	mheadid		PERMNO	CRSP permno or indno if index	%8d	list index
mporttret	mportrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Porttret	compounded monthly total return of assigned portfolio of selected type with dividends	%11.6f	list
mport	mports	porttype - see Monthly CRSP Portfolios for porttype numbers	Port	portfolio assignment in selected portfolio type at end of period	%4d	list
mportaret	mportrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Portaret	compounded capital appreciation of assigned portfolio of selected type without dividends	%11.6f	list
mportiret	mportrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Portiret	compounded income return of assigned portfolio of selected type	%11.6f	list

ts_print Monthly File Items *(Con't)*

Itemid	groupid	SUBNO	Item Header	Item Description	Format	Entity Types
mportstat	mports	porttype - see Monthly CRSP Portfolios for porttype numbers	Portstat	statistic at end of current period for portfolio of this type with dividends	%15.6lf	list
mportxsaret	mportxsrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Portxsaret	excess compounded capital appreciation returns over period vs. assigned portfolio of selected type without dividends	%11.6f	list
mportxstret	mportxsrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Portxstret	excess compounded total returns over period vs. assigned portfolio of selected type with dividends	%11.6f	list
mportxsiret	mportxsrets	porttype - see Monthly CRSP Portfolios for porttype numbers	Portxsiret	excess compounded income returns over period vs. assigned portfolio of selected	%11.6f	list
mprc	mdata	0 - last price 1 - last non-missing price	Prc Prcprev	last closing price or negative bid/ask average in the period last non-missing closing price in the period	%11.5f	list
mprcprevdt	mprcprev		Prcprevdt	data of past non missing price in period	%8d	list
mret	mdata		Ret	total return compounded daily return of period, return and dividends	%11.6f	list index
mreti	mrdata		Reti	total income return -compounded daily, dividends only of the return period	%11.6f	list index
mretx	mrdata		Retx	capital appreciation return – compounded daily return without dividends of period	%11.6f	list index
mshr	mshares	0 - ignore facshr from rights 1 - use all facshrs	Shr Shrxr	shares outstanding in 1000's at end of period shares outstanding in 1000's at end of period, rights excluded	%9d	list
mshrcd	mnames	0 - actual 1 – effective 2 – last	SC SCE SCL	share code at end of period share code at beginning of period last share code	%3d	list
mshrcls	mnames	0 – actual 1 – effective 2 – last	CL CLE CLL	share class at end of period share class at beginning of period last share class	%-1.1s	list
msiccd	mnames	0 actual 1 effective 2 last	SIC SICE SICL	sic code at end of period sic code at beginning of period last sic code	%4d	list
mticker	mnames	0 – actual 1 – effective 2 – last	Ticker Tickere Tickerl	ticker at end of period ticker at beginning of period last ticker	%-5.5s	list
mtind	mindres	indno of base index series	Tind	total return index level (if i an issue, based on 100.0 on first period of price range) with dividends	%11.5f	list index
mtrtscd	mnasdin	0 – actual 1 – effective 2 – last	Trtscd Trtscde Trtscdl	Nasdaq status code at end of period Nasdaq status code at beginning of period last Nasdaq status code	%2d	list
mvol	mdata		Vol	cumulative monthly volume	%9d	list
mvolavg	msdata		Volavg	average monthly volume	%9d	list
mvolmed	msdata		Volmed	median monthly volume	%9d	list
mxsaret	mxsrets	indno of base index series	Xsaret	excess compounded index series capital appreciation index returns over period vs. selected index series without dividends	%11.6f	list
mxsiret	mxsrets	indno of base index series	Xsiret	excess compounded income returns over period vs. selected index series	%11.6f	list
mxstret	mxsrets	indno of base index series	Xstret	excess compounded total returns over period vs. selected index series, with dividends	%11.6f	list

ts_print Formulas and Calculations

The following table contains formulas and calculations for derrived data in *ts_print* .

Primary Data Item type	Description	Formulas and Calculations	Related data items:

Cumulative Returns	The return reported for each period is the cumulative return from the beginning of the range to that period.	$r_c = (1+r_1)(1+r_2)¼(1+r_k)-1 \quad (1)$
		
		r_c: cumulative return at period k;
		r_i: return at period i;
		
		Any $r_i < -1$ is ignored. If all $r_I < -1$, result is -1.
		
		m: frequency of source data.
		n: frequency of target data.
		
		If $m=n$, for example, monthly source vs. monthly target, the source r_i is adopted directly as the single period return. Then formula *(1)* is used to calculate r_c.
		
		If m>n, for example, daily source vs. annually target, all the source returns corresponding to one target period will be cumulated using formula *(1)* to get each single period return. Then formula *(1)* is used again to calculate r_c.
		
		If m<n, for example, monthly source vs. weekly target, one source period corresponds to multiple target periods. In this case all the target returns related to the same period of a source return are set to the same value, the geometric average of the target return across all periods in the source. This value is calculated by solving the equation below.
		
		$(1+r)^n = 1+r_s$
		
		$r = (1+r_s)^{\frac{1}{n}} - 1$
		
		n: number of periods.
		r: single period target return.
		r_s: Source return.
		
		Then formula *(1)* is used to calculate r_c.
		
		Note: The source return can be a daily or monthly holding return in a CRSP stock or Indices file, or values derived from these returns. See stock item ret and `retx` and index items `tret`, `aret` and `itet` for details.
			`(m)cumaret`
			`(m)cumindtret`
			`(m)cumindaret`
			`(m)cumindiret`
			`(m)cumiret`
			`(m)cumparet`
			`(m)cumpiret`
			`(m)cumptret`
			`(m)cumtret`
			`(m)cumxsaret`
			`(m)cumxsparet`
			`(m)cumxspiret`
			`(m)cumxsptret`

Excess Returns	The return reported for each period is an excess return. It measures the extra return of a stock issue related to an index return.	r_s: The return for the specified stock issue. r_i: The related index return.
		r_e: The excess return.
		
		$r_e = r_s - r_i$
		
		Note: for returns with both excess and cumulative feature, the cumulative return is calculated first.
			`(m)cumxsaret`
			`(m)cumxiret`
			`(m)cumxparet`
			`(m)cumxpiret`
			`(m)cumxptret`
			`(m)portxsaret`
			`(m)portxstret`
			`(m)portxsiret`
			`(m)xsaret`
			`(m)xsiret`
			`(m)xstret`

Portfolio Returns	The portfolio return reported for each period is a return for a group of stock issues representing a statistically similar portfolio. For each time period, the portfolio is determined by assignment of the issue for the specified portfolio type. The portfolio return is the return of that portfolio over the time period. CRSP provides several market segment indices in the indices and portfolio file. The assignment and methodologies are described in the CRSPAccess 97 Indices File Guide. For example, if an issue is assigned to portfolio 3 in 1996 and portfolio 2 in 1997, the portfolio returns for that issue are those of portfolio 3 in 1996 and portfolio 2 in 1997 appended into one series.	$P_r(p,t)$ is the portfolio return for porttype p at time t. Then $P_r(p,t) = R(a_{p,}t)$ Where $R(a_p, t)$ is the return at time t of portfolio a_p. a_p is the assignment for porttype p effective at time t for that issue.	(m)cumparet (m)cumpiret (m)cumptret (m)cumxsparet (m)cumxspiret (m)cumxsptret (m)porttret (m)portaret (m)portiret (m)portstat (m)portxsaret (m)portxstret (m)portxsiret

3.2 *stk_print* Stock Database Report Writer

stk_print can be used to print CRSPAccess97 stock data. It is useful for browsing data formatted for a terminal or extracting data formatted for program input. It supports CRSP stock header, event, and time series data items and can support input typed at a terminal, securities in an input file, or all securities in the database. The user selects input and output options on the command line. If security identifiers are typed at the terminal, input or output options can be switched between each entry. Output can be printed to a terminal or saved in a file.

Use one of two commands to run *stk_print* :

`dstkprint` – to read the daily CRSP database

`mstkprint` – to read the monthly CRSP database

Normal usage is to type identifiers at the command line once the program is started. The enter key registers the input and triggers the program to report the desired data. Entering a blank line ends the program. Options can be added at the command line or after the program is started to select different identifiers, data, date ranges, or output options. A database can also be processed sequentially or from an input file with *stk_print*.

stk_print Options

Normally options are preceded with a forward slash. Multiple options can be placed on a single line. If an option does not require an additional description, another option can follow without a second slash. If the option does require additional information, there must be a space and another slash before another option is described. For example, `/hrndy 1 /fs` and `/hr /n /dy 1 /fs` are possible ways to select the `/hr, /n, /dy`, and `/fs` options. If there is any white space between the options, a front slash must be included. i.e. `/nif stkprt.txt /of names.out.` In this example, `/n` is requesting name information for each of the `permno's` in input file (`/if`) `stkprt.txt` to dump to an output file (`/of`) named `names.out`.

Following is a list of current *stk_print* options, grouped by option category, listing the options, the variables included in each option followed by an output sample for each option. The samples look at calendar dates 19961030-19961231. If the values are all 0, -88.0 or 99.0, there is no data in the file for the selected issue.

stk_print Data Items and Options

— Header Information

`/hh` identifiers

i.e.
```
/hh option: header information
Perm#  CUSIP#  Permco  Co. # Issue# EXCH SIC Name  Dist Share Delist Nasd
 12490 45920010  20990      0      0   1 3573    3   145  121      1     0

  BegDate/EndDate
19620702-19961231
```

/hr identifiers with ranges in terms of YYYYMMDD dates

i.e.
```
/hr option identifiers w/ calendar ranges
Perm#  CUSIP#  Permco  Co. # Issue# EXCH SIC Name  Dist Share Delist Nasd
 12490 45920010  20990      0      0   1 3573    3   145  121      1     0

  BegDat/EndDat           Bidlo               Prices            Spreads
19620702-19961231   19620702-19961231   19620702-19961231          0-       0

       Returns          Ret w/o Div           Trades            Askhi
19620702-19961231   19620702-19961231          0-       0   19620702-19961231

       Volume             Bids                Asks             Open
19620702-19961231          0-       0          0-       0

                    Portfolio types available

1 - nyse/amex/nasdaq cap assignments                                1962 - 1997
2 - nyse/amex cap assignment                                        1962 - 1997
4 - nyse cap assignment                                             1962 - 1997
6 - nyse/amex betas                                                 1962 - 1997
7 - nyse/amex standard deviations                                   1962 – 1997
```

/l, identifiers with ranges in terms of calendar day numbers (`/l` is used in conjunction with `/hr` in the following example) The `/l` option includes all of the options `/hr` does, and includes corresponding CRSP file calendar indexes instead of YYYYMMDD dates.

i.e.
```
/hrl option identifiers w/ calendar ranges & corresponding calendar day nos.
Perm#  CUSIP#  Permco  Co. # Issue# EXCH SIC Name  Dist Share Delist Nasd
 12490 45920010  20990      0      0   1 3573    3   145  121      1     0

  BegDat/EndDat           Bidlo               Prices            Spreads
19620702-19961231      1-    8686          1-    8686          0-       0

       Returns          Ret w/o Div           Trades            Askhi
       1-    8686        1-    8686         0-       0         1-    8686

       Volume             Bids                Asks             Open
       1-    8686          0-       0          0-       0

                    Portfolio types available

1 - nyse/amex/nasdaq cap assignments                                1962 - 1997
2 - nyse/amex cap assignment                                        1962 - 1997
4 - nyse cap assignment                                             1962 - 1997
6 - nyse/amex betas                                                 1962 - 1997
7 - nyse/amex standard deviations                                   1962 – 1997
```

— Event Information

/n name histories

i.e.
```
/n option (name histories)
Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
251231                      INTERNATIONAL BUSINESS MACHS COR       11   1  3570
620702               IBM    INTERNATIONAL BUSINESS MACHS COR       11   1  3573
680102   45920010   IBM    INTERNATIONAL BUSINESS MACHS COR       11   1  3573
```

/di distribution event histories

i.e.
```
/di option distribution event histories
Code     Amount Price fact Share fact Declare Ex    Record Paydt Acperm Accomp
1232    0.35000    0.00000    0.00000 960730 960807 960809 960910      0      0
1232    0.35000    0.00000    0.00000 961028 961106 961108 961210      0      0
```

/sh raw shares observation histories

i.e.
```
/sh option raw shares, event histories
      Shares       Date     Enddate          Flag
      527494    19960930    19961030          0
      527494    19961031    19961128          0
      527494    19961129    19961230          0
      517546    19961231    19961231          0
```

/sa shares histories adjusted for distributions

i.e.
```
/sa option, shares histories adjusted for distributions
      Shares       Date     Enddate          Flag
      597000    19960930    19961208          0
     1194000    19961209    19961230          1
     1198000    19961231    19961231          0
```

/de delisting histories

i.e.
```
Dlstdt Code Nwperm Nwcomp Nextdt Next Price  Dlpdt    Dlamt   Return     Returnx
961231  100      0      0      0     0.0000      0    0.000 -88.00000 -88.00000
```

/q Nasdaq information histories

i.e.
```
/q option delisting histories for an issue traded on The Nasdaq Stock Market℠
   Date       Last Date   Status    NMS ind    MM count   Index
   19960723     19961006      1         2         39         0
   19961007     19961008      1         2         40         0
   19961009     19961021      1         2         39         0
through 19961225
```

— Time Series Groups; only one of `dd, ds, dr, dx` can be used at the same time. These four cannot be used at the same time as one of the single time series.

/dd trading data including close, high/ask, low/bid, volume, and total return

i.e.
```
/dd option, trading data including close, high/ask, low/bid, volume and total return
  date        prc        high         low          vol        ret
19961001   123.75000   125.25000   123.12500     2446500    -0.006024
19961002   125.87500   125.87500   124.00000     2013200     0.017172
19961003   125.00000   126.25000   124.62500     1845100    -0.006951
```

/dr returns and returns without dividends

i.e.
```
/dr option, returns and returns without dividends
date        prc         ret         retx
19961001   123.75000    -0.006024    -0.006024
19961002   125.87500     0.017172     0.017172
19961003   125.00000    -0.006951    -0.006951
```

/dx price, shares, and returns. Shares outstanding are mapped to calendar of price and returns

i.e.
```
dx option, price, shares and returns.  Shares outstanding are mapped to calendar of
price and returns
date       price      shares     returns
19961001   123.75000    527494    -0.006024
19961002   125.87500    527494     0.017172
19961003   125.00000    527494    -0.006951
```

`/ds "YYYYMMDD|base.amt|"`

price, total returns index level, capital appreciation index level. Levels are set to `base.amt` on `YYYYMMDD`. The quotes are necessary on Unix or Windows NT systems.

i.e.
```
/ds 19961231|100.0| option
levels set to 100.000000 on 19961231

  date        prc      ind_tot     ind_apr
19961001   132.12500    79.95460    79.95460
19961002   134.75000    81.54310    81.54310
19961003   134.00000    81.08924    81.08924
```

— Portfolio Information for one or more Portfolio Types

/dy#-# portfolio assignments and statistics for portfolio type #. Porttype numbers are required. Porttype numbers can be a single number, a range separated by dashes, or a list separated by commas. For possible porttypes for this security, run the /hr option.

i.e.
```
/dy 1,3 option portfolio assignments and statistics for portfolio type 1,3.

Selected portfolios: 1,3
1 - nyse/amex/nasdaq cap assignments
3 – nasdaq cap assignment

  DATE            PORTFOLIO TYPE 1          PORTFOLIO TYPE 3
                  PORT#     STAT             PORT#     STAT
  1996            10    98984752.00000      10    98984752.00000
```

— Single Time Series; any of these active at the same time will be printed across

/pp prices

i.e.
```
/pp, option prices
Date       Prices
19961001     123.75000
19961002     125.87500
19961003     125.00000
```

/pr returns

i.e.
```
/pr option, returns
Date       Returns
19961001     -0.006024
19961002      0.017172
19961003     -0.006951
```

/px returns without dividends

i.e.
```
/px option, returns without dividends
Date       Ret w/o Div
19961001     -0.006024
19961002      0.017172
19961003     -0.006951
```

/pv volumes

i.e.
```
/pv option, volumes
Date       Volumes
19961001       2446500
19961002       2013200
19961003       1845100
```

/pl bidlo

i.e.
```
/pl option, bidlo
Date       Bidlow
19961001     123.12500
19961002     124.00000
19961003     124.62500
```

/ph askhi

i.e.
```
/ph option, askhi
Date       Askhigh
19961001     125.25000
19961002     125.87500
19961003     126.25000
```

/pb bid

i.e.
```
/pb option, bid
Date       Bids
19961001      0.000000
19961002      0.000000
19961003      0.000000
```

/pa ask

i.e.
```
/pa option, ask
Date       Asks
19961001      0.000000
19961002      0.000000
19961003      0.000000
```

/pn number of trades

i.e.
```
/pn option, number of trades
Date       Trades
19961001      4477
19961002      3971
19961003      3149
```

/ps shares. Shares outstanding are mapped to the calendar of prices

i.e.
```
/ps option, shares, shares outstanding are mapped to the calendar of prices
Date       Shares
19961001      527494
19961002      527494
19961003      527494
```

—use adjusted or raw values for price, volume or shares items. The default is unadjusted .

/djYYYYMMDDn

toggle to use adjusted values instead of raw values for any price, volume, or shares items above. N is the adjustment type code and YYYYMMDD is the date to use as the unadjusted base. n = 0 if only using stock splits and stock dividends to adjust, and n = 1 if all price factors are used to adjust. If YYYYMMDD precedes the range of the issue, the first day will be used for that issue. If YYYYMMDD follows the range of the issue, the last day will be used for that issue.

e.g. /dj option, toggle to adjusted values for the single timeseries options (/ dx, /dd, /ds, /ps, /pv and /pp). In the example, the security split December 12, 1996.

```
mstkprint /pp                                  /dj199612311
                                               (keeping the former options and selection)
10107
                                               Date        Prices
Date         Prices                            19961031     68.62500
19961031     137.25000                         19961129     78.43750
19961129     156.87500                         19961231     82.62500
19961231      82.62500
->
```

/du toggle back to unadjusted values for price, volume, and shares items.

e.g.

```
mstkprint /pp /dj 19961231

10107

Date        Prices
19961031      68.62500
19961129      78.43750
19961231      82.62500
->
```

```
/du
(keeping the former options and selection)

Date       Prices
19961031     137.25000
19961129     156.87500
19961231      82.62500
```

—set date ranges. The default is the last three months before the end of the calendar if date range is not set.

`/dt range1[-range2]`

ranges can be `YYYY`, `YY`, `YYYYMM`, `YYMM`, `YYYYMMDD`, or `YYMMDD`, in any combination. If only one range is given, and year only or month only is used, then the first period of the year or month is used for the beginning of the range and the last period of the year of month is used for the end of the range. Date ranges will be applied to all data selections except header, names, and delistings. If an issue does not trade the entire range, only the intersection of the issue range and the date range will be printed. Date `range1` must precede date `range2` if both are supplied. Date ranges relate to the event and time series data and do not alter the header information.

The output format options `/fr` and `/fs` alter the interpretation of date range. If the default `/fr` format option is used, names and delists are not restricted by date range, and the first shares observation or distribution event before and after the range, if any, are displayed. If the `/fs` format option is used, only names, shares, and distributions events in the range are displayed.

e.g.
```
/dt 199609-199612   all data from the beginning of September, 1996 to the end of December, 1996
/dt 1990            all data in the year 1990
/dt 1994-19940615   all data from the beginning of 1994 until June 15, 1994
/dt 19961231        data only on the date December 31, 1996
```

—set input method. The default is to allow the user to type in identifiers at the terminal. These options are only supported at the command line and only one can be used.

`/sq` reads all issues in database sequentially

i.e. For example, to display name history for all the issues in the monthly database,

```
mstkprint /n /sq
```

`/if filename.inp`

selects data for all identifiers in `filename.inp`. Any of the options may be selected to run with the input file. This input file should be a text file containing one column of identifiers, beginning in the first character space.

i.e. For example, to display name history for all `permno`s in an input file in the default directory named `perms.inp`,

```
mstkprint /n /if perms.inp
```

—set output method. The default is for output to be printed on the terminal. This option is only supported at the command line

`/of filename.out`

data is placed in `filename.out` instead of the terminal

i.e. For example, to save name history of selected securities to the file `filename.out` in the current directory,

```
dstkprint /n /of filename.out
```

—set output format. Default is for 80-character width output with headers

`/fr` toggle for 80-character formatted output with headers. This is the most readable when browsing data and supports multiple data items.

i.e.
```
/hh /fr
 Perm#  CUSIP#  Permco  Co. # Issue# EXCH SIC Name  Dist Share Delist Nasd
 12490 45920010  20990      0       0   1 3573    0     0    0      0     0

  BegDate/EndDate
19620702-19961231
```

/fs toggle for tabular delimited output, intended for input to another program. The permno is output on each line with this option. The /fs option is most useful when one data item (or multiple /p* data items) is used with sequential or file input, and file output.

i.e.
```
/fs (/fs /hh)
 12491|02407640|   196|   196|   228| 3|6711|  0|  0|  0|  0|  0|19780217|19961231
 12503|63934E10| 21265|     0|     0| 1|3523|  0|  0|  0|  0|  0|19620702|19961231
 12504|02432710|  5531|  5531|  6193| 2|6123|  0|  0|  0|  0|  0|19820525|19961231
 12511|56123210| 20992|     0|     0| 1|2819|  0|  0|  0|  0|  0|19620702|19961231
 ...
```

—set database. The default is the CRSP_DSTK database and daily data. These options are only supported on the command line at the initial program call, and cannot be switched. These commands can only be used with the stk_print command, since the database is automatically set with the dstkprint or mstkprint commands.

/d1 dbdirectory
to select an alternate database with a path of dbdirectory

i.e. stk_print /d1 mydirectory

/fm to indicate that the database is monthly

i.e. stk_print /fm /d1 mymonthdir

—set key. The default is permno. All input in the input file or at the terminal will be interpreted as this identifier. Sequential access will be in the order of this key. If a key is not unique such as permco, direct access will always find the first security with the identifier. Other securities can be found with next id (n) option.

/ky permno
to set input key to CRSP permno. This is the default if no /ky option is used.

i.e.
```
dstkprint /ky permno
10107

Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

/ky permco
to set input key to CRSP permco.

i.e.
```
dstkprint /ky permco
8048

Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

/ky cusip
to set input key to CRSP header CUSIP. Header CUSIPs are unique for each security

i.e.
```
dstkprint /ky cusip
59491810

Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

/ky hcusip

to set input key to CRSP historical CUSIP. Historical CUSIPs are the list of any CUSIPs in the name history plus the header CUSIP if no names exist in the name history. Each security will have one or more historical CUSIPs, and no historical CUSIP will appear in more than one security.

i.e.
```
dstkprint /ky hcusip
59491810

Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

/ky ticker

to set input key to header ticker. Header ticker is the latest ticker and is only set for securities active on the last date covered in the database. NYSE/AMEX securities with nonblank share class have a period and the share class appended to the ticker (`TICKER.A`). Header ticker is unique, but not all securities can be accessed by it.

i.e.
```
dstkprint /ky ticker
MSFT

Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

/ky siccd

to set input key to CRSP historical SIC code. A security can be accessed by any SIC classification in its history. More than one `siccd` can be used to access a security, and multiple securities can share the same `siccd`.

i.e.
```
dstkprint /ky siccd
7370
…
n     (step one at a time with the next option until finding the security of interest)
…
n
Date     CUSIP     Ticker         Company Name               Class Cd Exch  SIC
860313   59491810   MSFT    MICROSOFT CORP                        11   3  7370
```

The following codes can be used instead of a specified identifier at the command line or in an input file. These access securities by position relative to the current key set with the `/ky` option. These are input and not options and therefore do not require the forward slash line.

s same identifier
n next identifier
p previous identifier
f first identifier
l last identifier

stk_print Usage and Examples

stk_print is run at the command prompt with the form:

`dstkprint /options [/options] -` run off daily database `CRSP_DSTK`

`mstkprint /options [/options] -` use monthly database `CRSP_MSTK`

`stk_print /options [/options]`

stk_print defaults to the daily file. The `/d1` and `/fm` options can be used to select a user's daily or monthly database.

An on-screen help menu is available with *stk_print*. Within *stk_print*, type "?" which provides a listing and brief description of all of the `/options`.

Examples

```
dstkprint /hrndddt1995
```

- select header including ranges, names, all 1995 daily price, high, low, volume, and returns. User will type in `permnos` and output will be printed on the screen in a readable format.

```
mstkprint /di /fs /dt1994-1995 /if perms.inp /of dists.out
```

- use monthly database `CRSP_MSTK` and print delimited 1994 and 1995 distribution histories for all `permnos` in the file `perms.inp` (one `permno` per line) to the new file `dists.out`. `Perms.inp` must already exist in the default directory.

3.3 Portfolio Building Programs

There are two interactive programs available that can be used to easily calculate returns on stock portfolios. Portfolios can be created interactively, at the terminal, in a file, or randomly. Market indices can also be used as portfolios.

The user begins by setting the parameters that the program will use to create the portfolios:

- One of three identifiers is chosen to identify stocks: CRSP Permanent number, historical CUSIP, or header CUSIP.
- Beginning and ending dates of the portfolios are then entered. Dates can be entered in one of three formats, YYMMDD, YYMM, or YY. Beginning dates assume the first trading date of a month or year and ending dates assume the last trading date of the month or year.
- The returns series that will be used is chosen.
- The output filename is entered.
- These parameters are then set for all portfolios entered in this session. The program must be rerun if any of the parameters need to be changed.

Once the portfolio parameters have been entered, portfolios are entered one at a time. There are four methods for entering portfolios.

- Terminal input allows identifiers and weights of each stock in the portfolio to be entered interactively. Stock identifiers needed should be known before running a portfolio program. Unmatched identifiers generate a warning message and are ignored. Weights can be entered for each security, or the program can be directed to calculate weights one of three ways, equally-weighted by share, value-weighted, or equal-weighted by price.
- File input assumes that an input file already exists in your directory. This input file can be created with a text editor. Each line should have one security, with `CUSIPs` in columns 2-9 or perm numbers in columns 5-9, and weights in columns 11-20. Unmatched identifiers generate a warning message and are ignored by the portfolio. The program can override the weights provided and instead calculate weights equally by share or price, or by value.
- Random input causes the program to create a portfolio randomly. The number of securities and a random number seed are entered by a user. Random securities are chosen from the securities trading on exchanges. Only securities that traded the entire trading range chosen will be selected. Weighting can be done equally by share or price, or by value.
- Market indices are returns on portfolios of all or a significant part of the entire market that have already been calculated. These indices include value-weighted or equally-weighted returns with or without dividends on five different combinations of exchanges: NYSE/AMEX, Nasdaq, NYSE/AMEX/Nasdaq, NYSE only and AMEX only. Other indices such as the returns on certain decile portfolios or other types of performance data are also available. You must subscribe to the indices product to access indices other than the equal and value-weighted market indices for your stock product, or the S&P 500 Composite.

When the last portfolio has been entered, a results file and a log file are created in your current directory. The log file is an ASCII text file that can be typed to the terminal or printed. It describes the parameters used and lists the contents of each portfolio selected, including the securities used, their weights, a compounded portfolio return, and the corresponding column in the results file.

The results file is an ASCII text file. The first column is the date. Other columns are returns on portfolios. Each row contains the return of selected portfolios on one date.

dsxport *dsxport* is an interactive program that allows for the easy creation of daily portfolios. *dsxport* uses total returns to create its portfolios. The market indices available include equivalent value-weighted returns on entire individual and combinations of exchanges, returns or decile portfolios based on yearly betas, standard deviations, or year-end capitalizations, and returns on the S&P composite index with dividends. Returns on the S&P composite index without dividends are also available.

msxport *msxport* is an interactive program that allows for the easy creation of monthly portfolio returns. *msxport* can use monthly holding period with and without dividends to create its portfolios. The market indices available include equal and value weighted returns on entire individual or combinations of exchanges, returns on decile portfolios based on year-end capitalizations, and returns on the S&P composite index without dividends. In addition to the stock indices, returns on a long-term corporate bond portfolio, a long-term government bond portfolio, a U.S. Treasury bill portfolio, and the rate of change on the Consumer Price Index are also available.

3.4 Namelist Data and Search Utilities

CRSP provides header files for each CRSPAccess97 database. These name lists are useful for finding identifiers and name histories of securities when only partial information is known. The identifiers can then be used as input to other CRSP reporting utilities or programs. The files are fixed format text files and be accessed with system utilities or other tools. CRSP provides search utilities available to ease searches of the header files.

Every stock database contains four files:

1. `cheadfile.dat` - header list, one line per issue, sorted by `permno`, with the fields `permno`, `permco`, header CUSIP, Company Name, Ticker, CRSP Exchange Code, SIC code and price range.
2. `headfile.dat` - historical header list, one line per historical name, sorted by `permno` and effective name date, with the fields `permno`, `permco`, header CUSIP, Company Name, Ticker, CRSP Exchange Code, SIC code and effective range of name information.
3. `psortbyp.dat` - `permlist` of issues in the database; one `permno` per line sorted by `permno`.
4. `headind.dat` – index description, `setid`, and `indno` of all index series and groups in the database.

Namelist Search Utilities

There are two stock header search utilities,

1. *dstksearch* for historical header list in daily database
2. *mstksearch* for historical header list in monthly database

There are two index header search utilities,

1. *dindsearch* for index header list in daily database
2. *mindsearch* for index header list in monthly database

The utility uses a search string as input and finds program will ask for a search string. It will print the full list of header rows satisfying the simple search.

Operating System Specific Search Instructions

Windows NT

The command and the string, enclosed in double quotes, are entered at the command line at a command prompt window. For example,

```
> dstksearch "ibm"
>echo off
                        Daily Stock Headers
             Exchange Codes 1=NYSE, 2=AMEX, 3= Nasdaq
Perm# permco CUSIP             Company Name         Tick   EX SIC  date range

---------- N:\DATA\IEEELIT\DX9612\HEADFILE.DAT
 12490 20990          INTERNATIONAL BUSINESS MACHS C IBM    1 3573 620702-680101
 12490 20990 45920010 INTERNATIONAL BUSINESS MACHS C  IBM    1 3573 680102-961231
 75139 22064 03093810 AMERICUS TR FOR IBM SHS        BZP    2 6799 870720-920630
 75140 22064 03093820 AMERICUS TR FOR IBM SHS        BZS    2 6799 870720-920610
 75141 22064 03093830 AMERICUS TR FOR IBM SHS        BZU    2 6799 870720-920629
```

Unix or OpenVMS

Type the name of the search function. You will be prompted for the search string. No quotes are needed and case is ignored. For example,

```
$dstksearch

Enter search string: ibm

       Exchange Codes 1=NYSE, 2=AMEX, 3= Nasdaq

Perm# Permco CUSIP            Company Name           Tick  EX SIC date range

 12490 20990          INTERNATIONAL BUSINESS MACHS C IBM    1 3573 620702-680101
 12490 20990 45920010 INTERNATIONAL BUSINESS MACHS C  IBM    1 3573 680102-961231
 75139 22064 03093810 AMERICUS TR FOR IBM  SHS        BZP    2 6799 870720-920630
 75140 22064 03093820 AMERICUS TR FOR IBM  SHS        BZS    2 6799 870720-920610
 75141 22064 03093830 AMERICUS TR FOR IBM  SHS        BZU    2 6799 870720-920629

Try another string [y] ? n
```

APPENDICES

A. DATA CODE LISTING

A.1 Name Structure Codes

CINS Country Codes used in CUSIP

A = Austria	B = Belgium	C = Canada	D = Germany
E = Spain	F = France	G = United Kingdom	H = Switzerland
J = Japan	K = Denmark	L = Luxembourg	M = Mid-East
N = Netherlands	P = South America	Q = Australia	R = Norway
S = South Africa	T = Italy	U = United States	V = Africa - Other
W = Sweden	X = Europe Other	Y = Asia	

Share Type

This table lists the share type codes found in the CRSP stock files. The first digit describes the type of security traded.

Share Type - Digit #1 - Security Traded

Code	Definition
1	Ordinary Common Shares
2	Certificates
3	ADRs (American Depository Receipts)
4	SBIs (Shares Of Beneficial Interest)
7	Units (Depository Units, Units Of Beneficial Interest, Units Of Limited Partnership Interest, Depository Receipts, Etc.)

The second digit describes more detailed information about the type of security.

Share Type - Digit #2 - Type of Security

Code	Definition
0	Securities Which Have Not Been Further Defined.
1	Securities Which Need Not Be Further Defined.
2	Companies Incorporated Outside The U.S.
3	Americus Trust Components (Primes And Scores)
4	Closed-End Funds
5	Closed-End Fund Companies Incorporated Outside The U.S.
8	REIT's (Real Estate Investment Trusts)

Exchange

The following table is a list of CRSP codes for major North American security exchanges and indices.

North American Security Exchange & Indices Codes

Code	Exchange Name
-2	Halted By NYSE Or AMEX
-1	Suspended By NYSE Or AMEX
0	Not Trading On NYSE, AMEX, Or The NASDAQ Stock Market
1	New York Stock Exchange
2	American Stock Exchange
3	The NASDAQ National Market
5	Mutual Funds (As Quoted By NASDAQ)
10	Boston Stock Exchange
13	Chicago Stock Exchange
16	Pacific Stock Exchange
17	Philadelphia Stock Exchange
19	Toronto Stock Exchange
20	Over-The-Counter (Non-NASDAQ Dealer Quotations)
31	When-Issued Trading On The New York Stock Exchange
32	When-Issued Trading On The American Stock Exchange
33	When-Issued Trading On NASDAQ

A.2 Distribution Codes

A four-digit code describes distribution events. The first digit describes the distribution in general terms. The second digit describes the form or method of payment. The meaning of the third digit varies with the value of the first digit, and gives a more detailed description of the event. The fourth digit provides information about the tax status of the distribution.

The coding and meanings of the four digits are described below. For digits 2, 3, and 4, special conventions apply: a value of "0" implies that CRSP has not as yet discovered the descriptive information for the corresponding digit; a value of "1" implies sources have been checked and the status for the corresponding attribute is actually unspecified, not applicable, or not available for the distribution.

Distribution Codes

Digit	Code	Meaning
1 Event Type	1	ordinary dividend
	2	liquidating dividend
	3	exchanges and reorganizations
	4	subscription rights
	5	splits and stock dividends
	6	notation of issuance (change in shares outstanding)
2 Payment Method	0	unknown, not yet coded
	1	unspecified or not applicable
	2	cash, United States dollars
	3	cash, foreign currency converted to US dollars
	4	cash, Canadian dollars (now obsolete, converted to US dollars)
	5	same issue of common stock
	6	units including same issue of common stock
	7	an issue of a different common stock which is on the file
	8	other property
3 Dividend Frequency (for digit 1 equals “1” Only)	0	unknown, not yet coded
	1	unspecified or not applicable
	2	monthly
	3	quarterly
	4	semi-annual
	5	annual
	6	year-end or final
	7	extra or special
	8	interim
	9	non-recurring
3 Event Descriptor (for digit 1 equals “2” Only)	0	unknown, not yet coded
	1	unspecified or not applicable
	3	partial liquidation
	4	step in total liquidation
	5	final liquidation
	6	approval of liquidation
	7	sale of assets resulting in liquidation of company
	8	court proceedings determining status of company assets
3 Event Descriptor (for digit 1 = "2" only)	0	unknown, not yet coded
	1	unspecified or not applicable
	2	merger
	5	non-ordinary distribution in another stock
	6	reorganization
	7	option of stock
	8	exchange

Distribution Codes *(Con't)*

Digit	Code	Meaning
3 Rights Valuation Method (for digit 1 equals "4" Only)	0 1 2 3	unknown, not yet coded unspecified or not applicable market value indicated value
3 Split Type (for digit 1 equals "5" Only)	0 1 2 3 4 5 6 7	unknown, not yet coded unspecified or not applicable split stock dividend split & stock dividend option of cash distribution of different issue of common; same company initial distribution of other class of common; same company
3 Types of Offer or Reason for Issuance (for digit 1 equals "6" Only)	0 1 2 3 4 5 6 7 8	unknown, not yet coded unspecified or not applicable step in merger with company on file step in merger with company not on file stock conversion executive option exercise own tender offer: stock buy-back own exchange offer: recapitalization stock offering
4 Tax Status	0 1 2 3 4 5 6 7 8 9	unknown, not yet coded unspecified or not applicable normal taxable at same rate as dividends normal non-taxable return of capital (i.e., gain recognized, loss not) gain or loss realized compared with cost realized capital gain (Investment Companies) capital gains tax receipt fully taxable as ordinary income to individuals dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code

Coding convention note for distribution codes with the 4th digit (tax status) coded as 2 or 8: Until 1986, distribution codes 2 and 8 were used in conjunction with one another such that the 2 represented the part of the dividend qualifying for the dividend exclusion and the 8 representing the part that did not. Since the tax reform act of 1986, which eliminated the exclusion, these have been coded as 2's.

The following table describes some of the most commonly coded distribution events in the CRSP stock files. CRSP did not verify the tax status of ordinary dividends in the NYSE/AMEX file after April, 1987 or in the Supplemental NASDAQ file at any time. Instead, CRSP assigned ordinary dividends the default tax code (12*2); that is, US cash dividend, taxable in the normal way as a dividend. If a dividend received is in the form of a security which is traded on the CRSP Stock files, the dividend code will be in the form *7**.

The distribution codes 6***, excepting 6225, are informational. They indicate a significant change in the shares outstanding and the reason for the change. Code 6225 specifies a dividend amount. See the variable DIVAMT for additional information on the 6225 code.

Distribution Events Table

Category	Code	Description
DIVIDEND	1200	US cash dividend, tax status unknown.
	1202	US cash dividend, taxable in normal way.
	1212	US cash dividend, unspecified frequency, taxable same rate as dividends.
	1214	US cash dividend, tax status - return of capital, gain recognized, loss not.
	1218	US cash dividend, unspecified frequency, fully taxable as ordinary income to individuals.
	1222	US cash dividend, monthly, taxable same rate as dividends.
	1224	US cash dividend, monthly, tax status - return of capital, gain recognized, loss not.
	1228	US cash dividend, monthly, fully taxable as ordinary income to individuals.
	1232	US cash dividend, quarterly, taxable same rate as dividends.
	1234	US cash dividend, quarterly, tax status - return of capital, gain recognized, loss not.
	1238	US cash dividend, quarterly, fully taxable as ordinary income to individuals.
	1239	US cash dividend, quarterly, tax status - dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code.
	1242	US cash dividend, semi-annual, taxable same rate as dividends.
	1244	US cash dividend, semi-annual, tax status - return of capital, gain recognized, loss not.
	1248	US cash dividend, semi-annual, fully taxable as ordinary income to individuals.
	1252	US cash dividend, annual, taxable same rate as dividends.
	1254	US cash dividend, annual, tax status - return of capital, gain recognized, loss not.
	1258	US cash dividend, annual, fully taxable as ordinary income to individuals
	1262	US cash dividend, year-end or final, taxable same rate as dividends.
	1272	US cash dividend, extra or special, taxable same rate as dividends.
	1274	US cash dividend, extra or special, tax status - return of capital, gain recognized, loss not.
	1278	US cash dividend, extra or special, fully taxable as ordinary income to individuals.
	1282	US cash dividend, interim, taxable same rate as dividends.
	1292	US cash dividend, non-recurring, or proceeds from sale of rights, taxable same rate as dividends.
	1312	Cash dividend (foreign currency converted to US), unspecified frequency, tax status - unspecified or not applicable.
	1318	Cash dividend (foreign currency converted to US), unspecified frequency, fully taxable as ordinary income to individuals.
	1332	Cash dividend (foreign currency converted to US), quarterly, taxable same rate as dividends.
	1338	Cash dividend (foreign currency converted to US), quarterly. Fully taxable as ordinary income to individuals.
	1342	Cash dividend (foreign currency converted to US), semi-annual, taxable same rate dividends.
	1348	Cash dividend (foreign currency converted to US), semi-annual, fully taxable as ordinary income to individuals.
	1352	Cash dividend (foreign currency converted to US), annual, taxable same rate as dividends.
	1372	Cash dividend (foreign currency converted to US), extra or special, taxable same rate as dividends.
	1378	Cash dividend (foreign currency converted to US), extra or special, fully taxable as ordinary income to individuals.
	1712	Dividend in other issue on file, unspecified frequency, taxable same rate as dividends.
	1713	Dividend in other issue on file, non-taxable.
	1714	Dividend in other issue on file, taxable as return of capital.
	1718	Dividend in other issue on file, taxable as ordinary income to individuals.
	1772	Dividend in other issue on file with an extra or special frequency, taxable same rate as dividends.
	1812	Dividend in issue not on file, unspecified frequency, taxable as dividend.
	1813	Dividend in issue not on file, non-taxable.
	1814	Dividend in issue not on file, taxable as return of capital.
	1872	Special Dividend in issue not on file, taxable as normal dividend.
	1999*	Missing dividend terms, tax status - dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code.
LIQUIDATION	2161	Announcement of liquidation or liquidating plan, tax status unspecified.
	2171	Announcement of sale of assets, tax status unspecified.
	2181	Liquidation involved in court proceedings, tax status unspecified.
	2216	Cash paid in distribution, tax status - realized capital gains, (Investment Companies).
	2234	Cash paid in partial liquidation, tax status - return of capital, gain recognized, loss not.
	2235	Cash paid in partial liquidation, tax status - return of capital, gain, loss realized.
	2243	Cash paid as a step in liquidation, non-taxable.
	2244	Cash paid as a step in liquidation tax status - return of capital, gain recognized, loss not.
	2245	Cash paid as a step in liquidation, tax status - return of capital, gain or loss realized.
	2255	Cash paid as a final liquidating payment, tax status - return of capital, gain or loss realized.
	2744	Other issue on file distributed as a step in liquidation, tax status - return of capital, gain recognized, loss not.
	2817	Issue not on file distributed as a step in unspecified liquidation process, tax status - capital gains tax receipt.
	2844	Issue not on file distributed as a step in liquidation, tax status return of capital, gain recognized, loss not.
	2999*	Missing liquidation information, tax status - dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code.

* This code alerts the user to uncoded information and is inconsistent with the conventional distribution-coding scheme.

Distribution Events Table *(Con't)*

Category	Code	Description
ACQUISITION / REORGANIZATION	3215	Cash received, preferred redeemed, tax status -
	3222	Cash received in a merger, taxable same rate as dividends.
	3224	Cash received in a merger, tax status - return of capital gain, gain recognized, loss not.
	3225	Cash received in a merger, tax status - gain or loss realized compared with cost.
	3285	Cash received in an exchange of stock, tax status - gain or loss realized compared with cost.
	3723	Issue of file, received in a non-taxable merger.
	3724	Issue on file, received in a merger tax status - return of capital, gain recognized, loss not.
	3725	Issue on file, received in a merger, tax status - gain or loss realized compared with cost.
	3752	Issue on file, received as a non-ordinary stock distribution, taxable same rate as dividends.
	3753	Issue on file, received as a non-ordinary stock distribution, non-taxable.
	3762	Issue on file, received as a spin-off in reorganization, taxable same rate as dividends.
	3763	Issue on file, received as a spin-off in reorganization, non-taxable.
	3764	Issue on file, received as a spin-off in reorganization, tax status - return of capital, gain recognized, loss not.
	3783	Issue on file, received as an exchange, non-taxable.
	3784	Issue on file, received as an exchange, tax status - return of capital, gain recognized, loss not.
	3785	Issue on file, received as an exchange, tax status - gain or loss realized compared with cost.
	3823	Issue not on file, received in a merger, non-taxable.
	3824	Issue not on file, received in a merger, tax status - return of capital, gain recognized, loss not.
	3825	Issue not on file, received in a merger, tax status - gain or loss realized compared with cost.
	3852	Issue not on file, received as a non-ordinary distribution in another stock, taxable same rate as dividends.
	3853	Issue not on file, received as a non-ordinary distribution, non-taxable.
	3854	Issue not on file, received as a non-ordinary distribution, tax status - return of capital, gain recognized, loss not.
	3862	Issue not on file, received in a reorganization, taxable as dividend.
	3863	Issue not on file, received in a reorganization, non-taxable.
	3864	Issue not on file, received in a reorganization, tax status - return of capital, gain recognized, loss not.
	3883	Issue not on file, received in an exchange of stock, non-taxable.
	3885	Issue not on file, received in an exchange of stock, gain or loss realized compared with cost.
	3888*	Partially coded merger or exchange. Amount or some terms missing. Fully taxable as ordinary income to individuals
	3989*	Debenture without established market value, tax status - dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code.
RIGHTS	4523	Rights to buy more of this security, at market value, non-taxable.
	4533	Rights to buy more of this security at indicated value, non-taxable.
	4563	Rights to buy more of this security, non-transferable value at exdate, calculated (based on recdate if exdate unavailable), non-taxable.
	4623	Rights to buy 'units' that include this security, non-taxable.
	4722	Rights to buy another common issue on file, taxable same rate as dividends.
	4822	Rights to buy other securities at market value, taxable same rate as dividends.
	4823	Rights to buy other securities, nontaxable.
	4833	Rights to buy other securities at indicated value, non-taxable.
	4999*	Missing rights distribution, tax status - dividend reinvestment plan qualifies for the limited exclusion provided by Sec. 305(e) of the Internal Revenue Code.
STOCK	5523	Stock split, non-taxable.
	5533	Stock dividend, non-taxable.
	5763	Stock distribution in different issue of same company which trades on the file, non-taxable.
	5773	Initial stock distribution of other class of common, same company, which is on the file, non-taxable.
	5873	Initial stock distribution in different issue of common, same company, which does not trade on the file, non-taxable.
OFFER/ ISSUANCES	6511	Common shares increased/decreased for reasons not specified.
	6235	Common shares increased/decreased by merger with company not on file, tax status - gain or loss realized compared with cost.
	6261	Common shares increased/decreased through a stock buy-back, tax status - unknown.
	6521	Common shares increased/decreased by merger with company on file, tax status - unspecified or not applicable.
	6531	Common shares increased/decreased by merger with company not on file, tax status - unspecified or not applicable.
	6541	Common shares increased/decreased through stock conversion, tax status - unspecified or not applicable.
	6543	Common shares increased/decreased through stock conversion, non-taxable.
	6561	Common shares reduced through company's own tender offer or buy-back of shares, tax status - unspecified or not applicable.
	6571	Common shares increased/decreased through company's own exchange offer, tax status - unspecified or not applicable.
	6581	Common shares increased/decreased through sale of stock other than rights issue, tax status - unspecified or not applicable.

* This code alerts the user to uncoded information and is inconsistent with the conventional distribution-coding scheme.

A.3 Delisting Codes

Category	Code	Description
ACTIVE	100	Issue still trading NYSE/AMEX or NASDAQ
	150*	Issue still active, but no prices in this version of file
	160*	Issue stopped trading, but no prices in file after 840831
	170*	Issue stopped trading, but not delisted from current exchange (suspended or inactive)
MERGERS	200	Issue acquired in merger
	201	Merged into or in order to form an issue trading on NYSE
	202	Merged into or in order to form an issue trading on AMEX
	203	Merged into or in order to form an issue trading on NASDAQ
	205	When merged, shareholders receive shares of mutual funds
	240*	Flags merger with missing final distribution information
EXCHANGES	300	Issue acquired by exchange of stock
	301	Issue exchanged for issue trading on NYSE
	302	Issue exchanged for issue trading on AMEX
	303	Issue exchanged for issue trading on NASDAQ
	320	Issue exchanged for stock trading Over-the-Counter
	340*	Flags an exchange with missing final distribution information
	350*	Flags an exchange attempt that was not sufficient to "kill" issue
	390*	Flags an unsuccessful exchange attempt with missing distribution information
LIQUIDATIONS	400	Issue stopped trading as result of company liquidation
	450	Issue liquidated, final distribution verified, issue closed to further research.
	460	Issue liquidated, no final distribution is verified, issue closed to further research.
	470	Issue liquidated, no final distribution is verified, issue pending further research.
	480	Issue liquidated, no distribution information is available, issue is pending further research.
	490	Issue liquidated, no distributions are to be paid, issue closed to further research.
DROPPED	500	Issue stopped trading on exchange - reason unavailable
	501	Issue stopped trading current exchange - to NYSE
	502	Issue stopped trading current exchange - to AMEX
	503	Issue stopped trading current exchange - to NASDAQ
	505	Issue stopped trading current exchange - to Mutual Funds
	510	Issue stopped trading current exchange - to Boston Exchange
	513	Issue stopped trading current exchange - to Midwest Exchange
	514	Issue stopped trading current exchange - to Montreal Exchange
	516	Issue stopped trading current exchange - to Pacific Stock Exchange
	517	Issue stopped trading current exchange - to Philadelphia Stock Exchange
	519	Issue stopped trading current exchange - to Toronto Stock Exchange
	520	Issue stopped trading current exchange - trading Over-the-Counter
	550	Delisted by current exchange - insufficient number of market makers
	551	Delisted by current exchange - insufficient number of shareholders
	552	Delisted by current exchange - price fell below acceptable level
	560	Delisted by current exchange - insufficient capital, surplus, and/or equity
	561	Delisted by current exchange - insufficient (or non-compliance with rules of) float or assets
	570	Delisted by current exchange - company request (no reason given)
	572	Delisted by current exchange - company request, liquidation
	573	Delisted by current exchange - company request, deregistration (gone private)
	574	Delisted by current exchange - company request, bankruptcy, declared insolvent
	575	Delisted by current exchange - company request, offer rescinded, issue withdrawn by underwriter
	580	Delisted by current exchange - delinquent in filing, non-payment of fees
	581	Delisted by current exchange - failure to register under 12G of Securities Exchange Act
	582	Delisted by current exchange - failure to meet exception or equity requirements
	583	Delisted by current exchange - denied temporary exception requirement
	584	Delisted by current exchange - does not meet exchange's financial guidelines for continued listing.
	585	Delisted by current exchange - protection of investors and the public interest
	586	Delisted by current exchange - composition of unit is not acceptable
	587	Delisted by current exchange - corporate governance violation
	588	Conversion of a closed-end investment company to an open-end investment company
	600	Expired warrant or right
	601	Warrants, rights, or units called for redemption
	610	Unit split into its component parts
	700	Issue Delisted by Securities Exchange Commission
	801*	Issue Simultaneously listed on NASDAQ and NYSE
	802*	Issue Simultaneously listed on NASDAQ and AMEX

* These codes are intended to alert the user to delisting events undergoing further research. The individual digits in these codes do *not* necessarily conform to CRSP's standard delisting coding system.

A.4 NASDAQ Information Codes

Trading Status

TRTSCD	Description
0	unknown
1	active
2	only one market maker
3	suspended
4	inactive
5	delisted

NASDAQ National Market (NMS) Indicator

NMSIND	Description
0	unknown
1	NASDAQ Small-Cap before June 15, 1992
2	NASDAQ National Market
3	NASDAQ Small-Cap after June 15, 1992

Index

NSDINX	Description
0	unknown
1	no index
2	industrial
3	banks
4	other financial
5	insurance
6	transportation
7	utilities

A.5 Missing Return Codes

Missing Return Codes

Parameter	RET(I)	Reason For Missing Return
RMISSN	-44.0	missing excess return due to no portfolio assignment
RMISSD	-55.0	missing delisting return
RMISSG	-66.0	more than 10 trading days between this day and the day of latest preceding price
RMISSE	-77.0	not trading on an included exchange for this file
RMISSR	-88.0	no return, array index not within range of BEGRET and ENDRET
RMISSP	-99.0	missing return due to missing price

A.6 CUSIP Copyright Information

Subscriber agrees and acknowledges that the CUSIP Database is and shall remain valuable intellectual property owned by, or licensed to, CUSIP Service Bureau, Standard & Poor's ("CSB") and the American Bankers Association ("ABA"), and that no proprietary rights are being transferred to Subscriber in such materials or in any of the information contained therein. Subscriber agrees that misappropriation or misuse of such materials will cause serious damage to CSB and ABA and that in such event money damages may not constitute sufficient compensation to CSB and ABA; consequently, Subscriber agrees that in the even of any misappropriation or misuse, CSB and ABA shall have the right to obtain injunctive relief.

Subscriber agrees that Subscriber shall not publish or distribute in any medium the CUSIP standard numbers. Subscriber further agrees that the use of CUSIP numbers is not intended to create or maintain, and does not serve the purpose of the creation or maintenance of, a file of CUSIP numbers for any other third party recipient of such service and is not intended to create and does not serve in any way as a substitute of the CUSIP MASTER TAPE, PRINT, ELECTRONIC and/or CD-ROM Services.

NEITHER CSB, ABA NOR ANY OF THEIR AFFILIATES MAKE ANY WARRANTIES, EXPRESS OR IMPLIED, AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF ANY OF THE INFORMATION CONTAINED IN THE CUSIP DATABASE. ALL SUCH MATERIALS ARE PROVIDED TO SUBSCRIBER ON AN "AS IS" BASIS, WITHOUT ANY WARRANTIES AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE NOR WITH RESPECT TO THE RESULTS WHICH MAY BE OBTAINED FROM THE USE OF SUCH MATERIALS, NEITHER CSB, ABA NOR THEIR AFFILIATES SHALL HAVE ANY RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR OMISSIONS NOR SHALL THEY BE LIABLE FOR ANY DAMAGES, WHETHER DIRECT OR INDIRECT, SPECIAL OR CONSEQUENTIAL EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL THE LIABILITY OF CSB, ABA OR ANY OF THEIR AFFILIATES PURSUANT TO ANY CAUSE OF ACTION, WHETHER IN CONTRACT, TORT, OR OTHERWISE EXCEED THE FEE PAID BY SUBSCRIBER FOR ACCESS TO SUCH MATERIALS IN THE MONTH IN WHICH SUCH CAUSE OF ACTION IS ALLEGED TO HAVE ARISEN. FURTHERMORE, CSB AND ABA SHALL HAVE NO RESPONSIBILITY OR LIABILITY FOR DELAYS OR FAILURES DUE TO CIRCUMSTANCES BEYOND ITS CONTROL.

Subscriber agrees that the foregoing terms and conditions shall survive any termination of it right of access to the materials identified above.

B. CONTINUING RESEARCH

B.1 Missing or Extraneous Nasdaq Prices

The following issue histories in the Nasdaq file may contain non-Nasdaq prices or exclude Nasdaq prices.

Missing/Extraneous Nasdaq Prices

/HEADER/ CUSIP	PERMNO	Latest Historical Name
00157510	85033	A L C COMMUNICATIONS CORP
02680110	13179	AMERICAN INDS LTD
04708310	15500	ATHENS RED SVGS BK GA
09057810	18033	BIO TECHNOLOGY GEN CORP
11837410	63925	BUCKHORN INC DEL
12414510	20029	BUTTERFIELD EQUITIES CORP
15234210	22017	CENTOCOR INC
17725310	23852	CITIZENS TR CO PORTSMOUTH VA
22575510	27685	CRESCENT PETE CORP
26805710	31070	DYNAMICS RESEARCH
29443710	87653	EQUION CORP
30213210	59943	EXPERTELLIGENCE INC
37935510	65920	GLOBAL NAT RES INC
39841430	39758	GRAY GRAHILL EXPL CO
40490110	40687	HAILEY ENERGY CORP
45936210	46199	INTERNATIONAL CONTROLS CORP
69411210	60629	PACIFIC COAST SVGS & LN
72753710	63320	PLASMINE CORP
74905610	65163	QUIXOTE CORP
77570910	74477	ROSSINS ENVIRONMENTAL SVCS INC
86663510	63343	SUN BKS FLA INC
92763020	80881	VIPONT LABS CO
98836310	85009	YUBA GOLDFIELDS INC

B.2 Issues Not In The Nasdaq File

These issues are excluded from the Nasdaq file even though they may have traded on Nasdaq for some period.

Nasdaq Issues Not Included in the Nasdaq File

CUSIP	Company Name
00791910	ADVANCED ENERGY CONCEPTS INC
01074510	ALAMEDA BANCORPORATION
01170310	ALASKA GOLD CO
01976510	ALLNET COMMUNICATION SVCS INC
02476310	AMERICAN BUSINESS PRODS GA
02608910	AMERICAN FINL ENTPRISE
02714820	AMERICAN LEISURE CORP DEL
02742010	AMERICAN MED BLDGS INC
02990010	AMERICAN PUB ENERGY CO
03990310	AREA FINL CORP
04532110	ASPEN SYS CORP
04573510	ASSOCIATED MTG INVS
05591210	BRISA INTL S A
05591310	BPI RES LTD
06003010	BANGOR AMER INC
06737410	BARCLAY INDS INC
08953910	BIG SKY TRANSN CO
09062610	BIONEX CORP
12615810	C P PRODS CORP
13137930	CALTON INC NEW
13523110	CANADA SOUTHN PETE LTD
13739010	CALDEL OIL LTD
14068110	CAPTAIN INTL INDS LTD
17305910	CITINATIONAL DEV TR
19114410	COCA COLA BOTTLING CO L A
19114910	COCA COLA BOTTLING CO MIAMI
19115410	COCA COLA BOTTLING CO MID AMER
19118410	COCA COLA BOTTLING CO MIDWEST
20478120	COMPUCOMP CORP
20882L10	BRINCO LTD
20890310	CONSOLIDATED CAP INC OPP TR 1
20890410	CONSOLIDATED CAP INC OPP TR 2
25307520	DICKENSON MINES LTD
25307530	DICKENSON MINES LTD
25455210	DIPLOMAT ELECTRONICS CORP
26861310	ELXSI CORP
28138910	EDUCASTING SYS INC
28614510	ELECTROSOUND GRP INC
28614510	ELECTROSOUND GROUP INC
30241810	F D S HLDG CO
30241810	FDS HLDG CO
30938710	FARMERS BK ST DEL DOVER
31374720	FEDERAL REALTY INVT TR
39074820	GREAT LAKES REC CO
39126220	GREAT SOUTHWEST CORP
41711910	HARTMARX CORP
42220110	HEALTHCARE AFFILIATES INC

Nasdaq Issues Not Included in the Nasdaq File *(Con't)*

CUSIP	Company Name
44248710	HOVNANIAN ENTERPRISES INC.
44962210	IHOP CORP
44977410	INVG MORTGAGE SECS C
45168020	IDICO INC
45245130	IMASCO LTD CL A
45333710	INCOME OPPORTUNITY REALTY TR
45732630	INLAND CONTAINER CORP
45814110	INTELECOM CORP
46151110	INVS GNMA MTG BACKED SECS
47952810	JOHNSTOWN CONSOLIDATED RLTY TR
52770710	LEWIS BUSINESS PRODS INC
56613910	MARCADE GROUP INC
59514D10	MICROSIZE INC
60115810	MILLS MUSIC TR
60871010	MOLSON COMPANIES LTD
61578510	MOORE LTD
63645C10	NATIONAL INCOME REALTY TRUST SBI
66937610	NORWESCO INC
67061D10	NUTRI-FOODS INTL INC
69553310	PAGE PETE LTD
70154520	PARKWAY CO
70331710	PATINO N V
70978610	PEOPLES BAN CORP
72890020	PLAZA RLTY INVS
73282710	POPE & TALBOT INC
73923910	POWER CORP CDA LTD
74311410	PROFESSIONAL CARE INC
74835510	QUESTA OIL & GAS
75156810	RAMPAC
76289710	F D RICH HSG CORP
77580610	PRICE MEYERS CORP
78387810	SCA SVCS INC
79957010	SANANCO ENERGY CORP
81090010	SCOTTIE GOLD MINES LTD
81173710	SEAFIRST CORP
85852510	STELCO INC
88652010	TIERRA ENERGY CORP
89078410	TOPPS CHEWING GUM INC
89334960	TRANS WORLD AIRLS INC
89352610	TRANSCANADA PIPELINES LTD
89352810	TRANSCAPITAL FINANCIAL CORP
89584410	TRIANGLE BRICK CO
90465930	UNICORP AMERN CORP NEW
90659910	UNION FID CORP PA
91085810	UNITED MERCHANTS & MFRS INC
91259510	U S SHELTER CORP DEL
91674510	UPSON CO
92922210	VYQUEST INC
98180110	FIRST AR BANKSTOCK CORP

B.3 Issues Without NASD Data

These issues traded on Nasdaq after November 1, 1982, but CRSP does not have raw data from the NASD. There are no volumes or Nasdaq information data for these issues.

Issues without NASD Raw Data

/HEADER/ CUSIP	PERMNO	Latest Historical Name
02149210	12176	ALTIUS CORP
03987610	14876	AREA COMMUNICATIONS
05570010	16257	BSL TECHNOLOGY
12652010	64725	C3 INC
14975810	21822	CAYMAN IS REINS LTD
17725310	23852	CITIZENS TR CO PORTSMOUTH VA
21962010	27319	CORNWALL PETE & RES LTD
22575510	27685	CRESCENT PETE CORP
23808510	28792	DATAMETRICS CORP
37584710	84567	GILMAN SVCS INC
37935510	65920	GLOBAL NAT RES INC
40210010	40380	GULF AEROSPACE CORP
49116010	65314	KENTRON INTL INC
60359010	84727	MINI COMPUTER SYS INC
69411210	60629	PACIFIC COAST SVGS & LN ASSN
74670710	64793	PUTNAM DUOFUND INC
75885510	52548	REGENCY EQUITIES CORP
76027410	56848	REPUBLIC AIRLS INC
85764010	72742	STATESIDE ENERGY CORP
87241510	51035	TGC INC
89464410	56338	TRECO INC DEL
90490510	78183	UNION BANCORP DUBOIS PA INC
92022610	80120	VALLEYLAB INC
94733010	81788	WEB PRESS CORP

B.4 Nasdaq Issue Distinction

The following sets of Nasdaq issues represent either multiple price histories in the file that should be consolidated into one issue history, or multiple price histories that are presented as one price series in the file and should be divided.

Nasdaq Issue Distinction

CUSIP	PERMNO	NAME
66331110	58123	NORTH WEST TELECOMMUNICATIONS
66330810	58115	NORTH WEST TELECOMMUNICATIONS
68006110	59388	OLD NATL BK EVANSVILLE IN
68003310	59361	OLD NATL BANCORP IN
55340610	50535	M P S BANCORP
33741410		FIRST UNITED FINL SVCS INC
87185210	16265	B S L TECHNOLOGY
05570010	16257	B S L TECHNOLOGY
05554510		B M A CORP
12327710	19968	BUSINESS MENS ASSURN CO AMER
02889410	35431	AMERICAN PIONEER CORP
31991010	35431	FIRST FIDELITY SVGS & LN
58939910	52936	MERCURY GENERAL CORP (721214-790103)
		MERCURY GENERAL CORP (851120-851231)
91801510	79813	UTIL & INDS CORP DEL
		unknown
95258120	82086	WEST DRIEFONTEIN GOLD MNG LTD
27201420	30796	EAST DRIEFONTEIN GOLD MNG LTD
26202620		DRIEFONTEIN CONSOLIDATED
01976510	85033	ALLNET COMMUNICATION SVCS
00157510		ALC COMMUNICATIONS CORP
14975710	21814	CAYMAN CORP
14975910	21849	CAYMAN RES CORP

C. INSTALLATION AND DATABASE ADMINISTRATION

CRSPAccess97 databases are distributed on CD-ROM. The code and data for all supported platforms are distributed with each product set. Setup programs are provided on the first disk. See the machine-specific installation sections for details of installation on a particular system. Utility programs to convert CRSP databases to previous CRSP binary formats and create subset databases are described at the end of this section.

Installation consists of the following basic steps:

1. Copy program libraries and executables to a root directory on target machine
2. Copy data to a directory on the target machine
3. Set CRSP environment variables needed to use the CRSP programs

The installer must determine locations for program files, data, and log files before installation. Multiple database installations can exist if desired and disk space allows. Documentation is available in the CD `doc` directory but is not loaded by the setup programs. Program files are identical in each product and should only be copied once. Log directories are shared by all databases.

Use the CRSPAccess97 Products table in the CRSPAccess97 Release Notes supplement to determine the disk space needed for each CRSP product.

C.1 Using Multiple CRSP Products

CRSPAccess97 programs can support a daily and monthly database and indices. Each database includes security data with capitalization portfolio information plus a value-weighted, equal-weighted, and composite index. CRSP daily or monthly indices can be added to a corresponding stock database to add additional portfolio types and indices. Standalone fixed format ASCII Indices files are also included with the CRSPAccess97 Indices File / Portfolio Assignments Product. **Install the Indices last if both Stock and Indices are available.**

The programs supplied with each product are identical and should only be loaded once.

The log directory is shared by all databases and needs to be created only once. The log directory should be set in a scratch area where all intended users have write access.

CD-ROM Layout

- top level contains setup programs, CD descriptions, copyright information, and three folders.
- `root` – contains programs, libraries, header files, and sample programs in the following 6 directories: `bin`, `forsrc`, `include`, `lib`, `sample` and `src`. Root is only included on the first CD of a multiple CD product.
    - `bin` contains executable programs and scripts; there is a subdirectory for each supported operating system / platform.
        - `wnt40x86` – Windows NT 4.0 on Intel x86
        - `w9540x86` – Windows 95 4.0 on Intel x86
        - `vms65axp` – OpenVMS 6.2 on AXP
        - `sol25spa` – Solaris 2.5 on Sparc
        - `unx40axp` – Digital Unix 4.0 on AXP
        - `wnt40axp` – Windows NT 4.0 on AXP
        - `hpu10par` – HP/UX Unix 10.2 on PARISC
    - `lib` contains object libraries and initialization files; there is a subdirectory for each supported operation system / platform
        - `wnt40x86` – Windows NT 4.0 on Intel x86
        - `w9540x86` – Windows 95 4.0 on Intel x86
        - `vms65axp` – OpenVMS 6.2 on AXP
        - `sol25spa` – Solaris 2.5 on Sparc
        - `unx40axp` – Digital Unix 4.0 on AXP
        - `wnt40axp` – Windows NT 4.0 on AXP
        - `hpu10par` – HP/UX Unix 10.2 on PARISC
    - `include` – FORTRAN and C header files
    - `sample` – FORTRAN and C sample programs
    - `src` – C source code
    - `forsrc` – FORTRAN source code
- `data` – contains actual database files. Bottom level directories exist as needed based on the product and part of the set.
    - `ieeebig` – IEEE big-endian database files (Sun, HP). Subdirectories are named as the product code and the year and month of data (YYYYMM).
        - ppYYYYMM
    - `ieeelit` – IEEE little-endian database files (PC, Alpha)
        - ppYYYYMM
- `doc` – CRSP documentation in different formats
    - `bin` – Adobe Acrobat `pdf` versions of the Users Guide, Programmers Guide, and Release Notes.
    - `html` – html versions of the Users and Programmers File Guides and the Release Notes. (These files contain relative links beginning with the `xxxcovno.html` file, which contains the table of contents.)
    - `word` – Microsoft Word 97 versions of the Users Guide, Programmers Guide, and Release Notes.

C.2 Installation

Windows NT Installation on Intel x86 Computers

The setup program will copy program and data files to your machine and can also be used to set CRSP environment variables. To run the setup program:

1. login to an account with administrator permissions to set up CRSP for all users on the machine.
2. load the first disk of your product to your CD drive,
3. double click on the setup.exe application, and
4. follow the instructions in the setup menus.

When the setup program asks for a database folder, use the browse… option to choose a folder other than the default. A folder can be typed in or selected with the menus. If the folder does not exist the program will ask if you wish to create it. In these screens, the Back option is used to skip copying data and does not return to a previous screen. The setup program will check for available space before copying and will fail if the supplied directory does not have enough free space. If the program asks for a second CDROM, make sure the computer registers the new CDROM before continuing.

If the root has been loaded from one product, the section should be skipped in succeeding products.

An administrator can set permissions on directories as needed. Write access is not needed for any program or data directories. The log directory should have full write access.

Environment Variables

Environment variables must be set for the CRSP programs to work properly. Environment variables can be set at the system or user level. At the system level all accounts will have access to the CRSP data. A user with a local copy of the database can set the environment variables at the user level so other users are not affected.

The setup application can be used for changing environment variables in the initial setup or if files are moved. In the initial setup, environment variables can be set based on the destination directories used for program files or data. Run the application and skip the sections up to the environment setup to make changes.

The Environment tab of the Control Panel / System Window can be used to modify environment variables. The Path may need setting by hand. Use the delete option to remove CRSP environment variables that are no longer needed. The `set` command in a command prompt window can also be used to show the current assignments. To assure usage of new environment variables, hit okay in the Control Panel / System Window or re-login, and restart any programs.

The following list of environment variables are used in CRSPAccess97.

Environment variable	Usage
`CRSP_ROOT`	Top level program directory. Most other CRSP environment variables are set based on `CRSP_ROOT`
`CRSP_LOG`	Log directory used for user
`CRSP_MSTK`	CRSP Monthly Database directory (if available)
`CRSP_DSTK`	CRSP Daily Database directory (if available)
`CRSP_INCLUDE`	Programming header files; include subfolder of root
`CRSP_SAMPLE`	Sample programs; sample subfolder of root
`CRSP_LIB`	Object libraries; `acclib` subfolder of root
`CRSP_BIN`	Executables and scripts; `accbin` subfolder of root
`CRSP_ENV_ULOG`	Usage logs produced by users; =`CRSP_LOG`
`CRSP_ENV_ELOG`	Error logs produced by users; =`CRSP_LOG`
`CRSP_ENV_EMSG`	Location of error messages file; =`CRSP_LIB`
`CRSP_ENV_LMSG`	Location of log messages file; =`CRSP_LIB`
`CRSP_ENV_ROOT`	Location of base CRSPAccess97 database
`CRSP_ENV_PATH`	Location of default CRSPAccess97 database
`CRSP_ENV_INIT`	Location of initializations file; `CRSP_LIB`
`PATH`	The `CRSP_BIN` folder must be included in the `PATH` list of directories to run CRSP programs

Windows 95 Support

The setup application provided with Windows NT will work on a Windows 95 machine but cannot set environment variables. The environment variables must be included in the autoexec.bat file on the root hard drive. The `accbin` folder in the CRSP `root` folder contains a file called `crsp_symbols.bat`. This file must be edited to reflect the local CRSP file locations, and either called by the `autoexec.bat` file on the system disk or added directly to the `autoexec.bat` file. See the Windows NT section for Intel x86 descriptions of the environment variables.

Windows NT Support on Digital Alpha

The setup application is not supported on Alpha Windows NT. Files must be copied by hand with the NT Explorer and environment variables must be set directly in the Control Panel.

Copying Files

Files are copied with the NT Explorer Application:

1. create a new folder in the target location for CRSP root files (skip 1-5 if not the first CRSP product). The folder must be visible on the left portion of Explorer.
2. Double-click on the `root` folder of the CD. Drag the `include` and `sample` folders to the new root folder.
3. Double-click on the `lib` folder. Drag the `wnt40axp` folder to the new root folder. Highlight and select `rename` with the right mouse button. Type the folder name `acclib`.
4. Select the `bin` folder under `root` on the CD. Drag the wnt40axp folder to the new root folder. Highlight and select `rename` with the right mouse button. Type the folder name `accbin`.
5. Create a new folder in a scratch area for CRSP log files.
6. Create a new folder in the target location for the CRSP data files.
7. Navigate the CD down the `data` and `ieeelit` folders. Double click on the data directory.
8. Use `Select All` and `Copy` commands in the `Edit` menu.
9. Select the target folder and choose the `Paste` command from the `Edit` menu.

The CRSPAccess97 Indices File / Portfolio Assignments Product contains extra files for the daily or monthly stock databases. To overlay the additional files, select the data folder with the first two letters matching the existing stock database. Select and copy the files in the folder and paste into the existing database directory. Files will match existing files in this folder. Choose `Yes to All` to overwrite.

Environment Variables

Environment variables are entered by hand with the `Environment` tab of the `System` window of the `Control Panel`. You must be logged into an account with Administrator permissions to set system level variables. System level variables are accessible by all users on the computer. User level variables take precedence over system variables, but only for the user setting the variables.

Environment variable	Set To (end paths with \ unless otherwise noted)
CRSP_ROOT	Path of root folder created in step 1
CRSP_LOG	Path of log folder created in step 5 (no ending \ )
CRSP_MSTK	Path of CRSP Monthly Database directory (if available)
CRSP_DSTK	Path of CRSP Daily Database directory (if available)
CRSP_INCLUDE	Path of include subfolder of root
CRSP_SAMPLE	Path of sample subfolder of root (not ending \ )
CRSP_LIB	Path of acclib subfolder of root
CRSP_BIN	Path of accbin subfolder of root
CRSP_ENV_ULOG	Same as CRSP_LOG
CRSP_ENV_ELOG	Same as CRSP_LOG
CRSP_ENV_EMSG	Same as CRSP_LIB
CRSP_ENV_LMSG	Same as CRSP_LIB
CRSP_ENV_ROOT	Path of base CRSPAccess97 database
CRSP_ENV_PATH	Path of default CRSPAccess97 database
CRSP_ENV_INIT	Same as CRSP_LIB
PATH	Add a semicolon and the CRSP_BIN path to the end of the current PATH (no ending \ on CRSP_BIN path)

Solaris Installation on Sun Sparcstation

The setup shellscript will copy program and data files to your machine. There is a program file script included that can create a machine-specific `shellscript` that can be called by any user to set CRSP environment variables. To run the setup script:

1. load the first disk of your product to your CD drive. The system should automatically mount it
2. in a terminal window, `cd` to a local directory. Use the `df` command to find the path of the CDROM
3. copy the `setup.sh` file from the top level directory of the CDROM to your local directory, for example

```
cp /cdrom/max1-199612/setup.sh .
```

4. execute the script:

```
setup.sh
```

5. answer prompts as needed. Directories be specified without trailing slashes.
6. if the program prompts for another CD, eject the current CD and insert the requested CD from outside the terminal window running the program. The `eject cdrom` command at a second terminal window or the eject disk button in the file manager can be used to eject a CD.

If the root has been loaded from one product, the section should be skipped in succeeding products.

Digital Unix Installation on Digital Alpha

The setup shellscript will copy program and data files to your machine. There is a program file script included that can create a machine-specific shellscript that can be called by any user to set CRSP environment variables. To run the setup script:

1. load the first disk of your product to your CD drive. You must know the location on your system of the CDROM drive.

2. Mount the CD with the command (device `/dev/rz4c` is an example here and may be different on your machine) `mount –t cdfs –o noversion /dev/rz4c /cd`

3. in a terminal window, `cd` to a local directory. The path of the CDROM is `/cd`.

4. copy the setup.sh file from the top level directory of the CDROM to your local directory, for example

```
cp /cd/setup.sh
```

5. execute the script:

```
setup.sh
```

6. answer prompts as needed. Directories should not be specified with trailing slashes.

7. if the program prompts for another CD, eject the current CD and insert the requested CD from outside the terminal window running the program. The `umount /cd` command at a second terminal can be used to eject a CD. Load the new CD with the same command as in step 2.

8. Use `umount /cd` to dismount the CD

If the root has been loaded from one product, the section should be skipped in succeeding products.

HP/UX Installation

The setup shellscript will copy program and data files to your machine. There is a program file script included that can create a machine-specific shellscript that can be called by any user to set CRSP environment variables. To run the setup script:

1. Load the first disk of your product to your CD drive. You must know the location on your system of the CDROM drive.

2. Mount the CD with the command (device `/dev/dsk/c02d0` is an example here and may be different on your machine)

   ```
   mount/dev/dsk/c0t2d0 /cdrom
   ```

3. In a terminal window, `cd` to a local directory. The path of the CDROM is `/cdrom`

4. Copy the `setup.sh` file from the top level directory of the CDROM to your local directory, for example

   ```
   cp /cdrom/setup.sh
   ```

5. Execute the script:

   ```
   setup.sh
   ```

6. Answer prompts as needed. Directories should be specified **without** trailing slashes.

7. If the program prompts for another CD, eject the current CD and insert the requested CD from outside the terminal window running the program. The `umount /cdrom` command at a second terminal can be used to eject a CD. Load the new CD with the same command as in step 2.

8. Use `umount /cdrom` to dismount the CD

If the root has been loaded from one product, the section should be skipped in succeeding products.

Unix Environment Variables

Environment variables must be set for the CRSP programs to work properly. CRSP provides shellscripts that prompts for directories and creates either Unix `ksh` or `csh` scripts that can be used to set CRSP definitions.

1. `cd` to the root directory where program files have been loaded.

2. `cd accbin`

3. if you are running csh shell
   `source crsp_setup.csh`

   if you are running ksh shell,
   `crsp_setup.sh`

4. the script will prompt for data, root, and log directories. Follow the instructions on the prompts in terms of trailing slashes in directory names.

5. The script will create to new scripts, `crsp.cshrc` in `csh` or `crsp.kshrc` in `ksh`.

6. The new scripts can be incorporated into all accounts by the system administrator, or sourced into initialization files directly by individual users.

One method of incorporating CRSP initializations for `csh` users is:

Move the file `crsp.cshrc` to the `/local/bin` or other common directory

Add `source /local/bin/crsp.cshrc` to the `.cshrc` file of each account using CRSP

One method of incorporating CRSP initialization for `ksh` users is:

Move the file `crsp.kshrc` to the `/local/bin` or other common directory

Add `. /local/bin/crsp.kshrc` to the `.kshrc` file of each account

Add the line `ENV=$HOME/.kshrc` to the `/etc/.login` file

`env | grep CRSP` can be used to check the CRSP environment variables set

The script produced by the CRSP setup script also adds the `$CRSP_BIN` directory to the users' path.

The following list of environment variables are used in CRSPAccess97.

Environment variable	Usage
CRSP_ROOT	Top level program directory. Most other CRSP environment variables are set based on CRSP_ROOT
CRSP_LOG	Log directory used for user
CRSP_MSTK	CRSP Monthly Database directory (if available)
CRSP_DSTK	CRSP Daily Database directory (if available)
CRSP_INCLUDE	Programming header files
CRSP_SAMPLE	Sample programs
CRSP_LIB	Object libraries
CRSP_BIN	Executables and scripts
CRSP_ENV_ULOG	Usage logs produced by users
CRSP_ENV_ELOG	Error logs produced by users
CRSP_ENV_EMSG	Location of error messages file
CRSP_ENV_LMSG	Location of log messages file
CRSP_ENV_ROOT	Location of base CRSPAccess97 database
CRSP_ENV_PATH	Location of default CRSPAccess97 database
CRSP_ENV_INIT	Location of initializations file
PATH	The CRSP_BIN folder must be included in the PATH list of directories to run CRSP programs

OpenVMS Installation on Digital Alpha

The setup command file will copy program and data files to your machine. There are also command files included that can be called on a user, group, or system level to set CRSP logicals and symbols. To run the setup command file:

1. load the first disk of your product to your CD drive.

2. Mount the CD with the command (the LABEL is the first 11 characters of your external label, and LABEL#1 is the same label followed by the characters #1)

```
mount /media=cd cd LABEL /undefined_fat=(stream_lf:500) –
 /shared /bind=LABEL#1
```

3. in a different terminal, `set default` to a local directory. The path of the CDROM is `cd:`.

4. copy the `setup.com` file from the top level directory of the CDROM to your local directory, for example
```
copy cd:[000000]setup.com []
```

5. execute the command file:
```
@setup
```

6. answer prompts as needed. The CD location is CD`:[000000]`. The root directory must be specified with a full directory path, ending with a period following the last subdirectory, and no closing bracket. Data directories must have the closing bracket.

7. if the program prompts for another CD, return to the terminal that mounted the first CD and execute the command `dismount cd:` . Insert the requested CD and mount with the same command as step 2 but using the new label based on the first eleven characters of the new CD's external label.

8. Use `dismount cd:` when finished to dismount the CD

If the root has been loaded from one product, the section should be skipped in succeeding products.

Logicals and Symbols

Logicals and symbols must be set for the CRSP programs to work properly. CRSP provides command files that can be used at a system, group, or process level to set directories and program symbols. They are located in the `[.BIN]` subdirectory of the root directory where program files have been loaded.

`crsp_logicals_install.com` sets all directories of databases and programs. It is run with six arguments:

1. One of `P` or `G` or `S` – this determines whether logicals will be loaded to the process, group or system logical names table. The account must have `GRPNAM` permission to load to the group table, and `SYSNAM` permission to load to the system table

2. The root directory in the form `disk:[root.]`

3. A log directory in the form `disk:[log.]` . The directory and two subdirectories must exist. The names of the two subdirectories are `[.usage]` and `[.errs].` The log disk and directories should be in a scratch disk with write access to any CRSP user.

4. A CRSP Daily Database directory. A monthly database directory can be substituted if no daily database directory exists on the system.

5. A CRSP Monthly Database directory. A daily database directory can be substituted if no monthly database directory exists on the system.

6. A . (period)

`crsp_symbols_install.com` defines program names so they can be run with arguments. This command file is run with no arguments.

The two initialization files should be called by a system initialization or a user login.com so they are available to targeted users at login. A single user can run them in a local process to set a personal copy of a database. The command `show logicals crsp*` can be used to verify the logicals set.

The following list of logicals are used in CRSPAccess97.

Logical	Usage
CRSP_ROOT	Top level program directory. Most other CRSP environment variables are set based on CRSP_ROOT. CRSP_ROOT is set as a concealed logical.
CRSP_LOG	Log directory used for user files. CRSP_LOG is set as a concealed logical
CRSP_MSTK	CRSP Monthly Database directory (if available)
CRSP_DSTK	CRSP Daily Database directory (if available)
CRSP_INCLUDE	Programming header files
CRSP_SAMPLE	Sample programs
CRSP_LIB	Object libraries
CRSP_BIN	Executables and scripts
CRSPDB_ULOGDIR	Usage logs produced by users
CRSPDB_ELOGDIR	Error logs produced by users
CRSPDB_EMSGDIR	Location of error messages file
CRSPDB_LMSGDIR	Location of log messages file
CRSPDB_ROOTDIR	Location of base CRSPAccess97 database
CRSPDB_PATHDIR	Location of default CRSPAccess97 database
CRSPDB_INITDIR	Location of initializations file

C.3 CRSP Dump Utilities

There are two utility programs that can be used to create the CRSP Subscriber FORTRAN sequential binary files supported prior to 1997 from a CRSPAccess97 database. CRSP binary files are FORTRAN readable data and calendar/indices binary files backward compatible with existing CRSP subscriber sample binary programs and SAS PROC DATASOURCE.

The first program creates a stock file and the second program creates a matching calendar/indices file. Both support the creation of IEEE binary formats that can be read by CRSP sample FORTRAN programs on various platforms.

See the CRSPAccess97 Stock File Programmers Guide for technical specifications of this format.

stk_dump_bin

This program creates a single CRSP binary stock data file from a CRSPAccess97 `CRSPDB` stock database. It contains options on the database location and type and output record options. To run the program, type the name of the program followed by parameter options at a command prompt. The parameters are

Parameter	Values
`CRSPDB` directory path	The directory where the database is stored. Standard environment names can be used such as `$CRSP_DSTK` on UNIX, `crsp_dstk:` on OpenVMS, or `%crsp_dstk%` for Windows NT.
Output file name	The name of the output file that will be created by the program. Make sure there is enough disk space in the location where this will be created. The output size will be roughly based on the size of the input database and the proportion of `PERMNO`s selected
Stock Setid	The database type. Use one of 10 if a daily stock database 20 if a monthly stock database
Port type 1	The portfolio type to load into the `PRTNUM(CAP,)` and `YRVAL(CAP,)` records. Check stock documentation for the numbers available for your product. Use 0 to load no portfolio information in the `CAP` fields.
Port type 2	The portfolio type to load into the `PRTNUM(SDEV,)` and `YRVAL(SDEV,)` records. Check stock documentation for the numbers available for your product. Use 0 to load no portfolio information in the `SDEV` fields.
Port type 3	The portfolio type to load into the `PRTNUM(BETA,)` and `YRVAL(BETA,)` records. Check stock documentation for the numbers available for your product. Use 0 to load no portfolio information in the BETA fields.
Byte-ordering flag	One of `keep` - keep the same byte-ordering as your current machine. `reverse` – reverse the byte-ordering of your current machine. Use this option if the output data intended usage will be on a machine with different byte-ordering than your current machine. Intel x86 and Digital Alpha computers are Little-Endian order and Sun Sparcstations are Big-Endian order. If the program is run on a Sun and the output will be used on a PC the reverse option should be used.
Record-delimiter type flag	One of `vms` - All records will contain four bytes of leading information, two bytes with a record length (excluding the 4-byte overhead) and two bytes with a segment id. These files will be compatible with OpenVMS Alpha systems `unix` - All records will contain four bytes of leading information and four bytes of trailing information. The record information includes the record length (excluding the 8-byte overhead). These files will be compatible with most UNIX systems. `none` – There is no additional record information. These file will be compatible with Windows NT and any other system where FORTRAN can read stream files.
Permlist file	The name of a file with a list of `PERMNO`s, one to a line. This parameter is optional. If it is used, only the `PERMNO`s in the input file will have data copied to the new database. If the parameter is not used, all `PERMNO`s in the input database will be copied.

ind_dump_bin

This program creates a single CRSP binary calendar/indices data file from a CRSP Access97 `CRSPDB` stock database. It contains options on the database location and type and output record options. To run the program, type the name of the program followed by parameter options at a command prompt. The parameters are

Parameter	Values
`CRSPDB` directory path	The directory where the database is stored. Standard environment names can be used such as `$CRSP_DSTK` on UNIX, `crsp_dstk:` on OpenVMS, or `%crsp_dstk%` for Windows NT. Use the same `CRSPDB` as in `stk_dump_bin` to create a compatible calendar/indices file
Output file name	The name of the output file that will be created by the program
Indices `Setid`	The database type. Use one of 460 if a daily indices database 420 if a monthly indices database
`Indno` 1	The `INDNO` of index loaded to the value-weighted returns and weight fields (`VWRETD`, `VWRETX`, `USDVAL`, and `TOTVAL`) of the calendar/indices files. See Stock or Indices documentation or run the *indsearch* utility to get a list of available `INDNO`s.
`Indno` 2	The `INDNO` of index loaded to the equal-weighted returns and count fields (`EWRETD`, `EWRETX`, `USDCNT`, and `TOTCNT`) of the calendar/indices files. See Stock or Indices documentation or run the *dindsearch* or *mindsearch* utility to get a list of available `INDNO`s.
`Indno` 3	The `INDNO` of index loaded to the S&P 500 index and returns fields (`SPINDX`, `SPRTRN`) of the calendar/indices files. See Stock or Indices documentation or run the *dindsearch* or *mindsearch* utility to get a list of available `INDNO`s.
Byte-ordering flag	One of `keep` - keep the same byte-ordering as your current machine. `reverse` - reverse the byte-ordering of your current machine. Use this option if the output data intended usage will be on a machine with different byte-ordering than your current machine. Intel x86 and Digital Alpha computers are Little-Endian order and Sun Sparcstations are Big-Endian order. If the program is run on a Sun and the output will be used on a PC the reverse option should be used.
Record-delimiter type flag	One of `vms` - All records will contain four bytes of leading information, two bytes with a record length (excluding the 4-byte overhead) and two bytes with a segment id. These files will be compatible with OpenVMS Alpha systems `unix` - All records will contain four bytes of leading information and four bytes of trailing information. The record information includes the record length (excluding the 8-byte overhead). These files will be compatible with most UNIX systems. `none` - There is no additional record information. These file will be compatible with Windows NT and any other system where FORTRAN can read stream files.

Examples

The following commands can be used in a command prompt window to create a daily PC binary file and matching index, assuming a CRSP AF product loaded on an Intel x86 PC running Windows NT:

```
stk_dump_bin %crsp_dstk% stkfl.bin 10 1 0 0 keep none

ind_dump_bin %crsp_dstk% calfl.bin 460 1000080 1000081 1000502 keep none
```

The following commands can be used in a command prompt window to create a monthly Sun binary file and matching index, assuming a CRSP AF product loaded on a Sun Sparcstation running Solaris:

```
stk_dump_bin $CRSP_MSTK stkfl.bin 20 1 0 0 keep unix

ind_dump_bin $CRSP_MSTK calfl.bin 420 1000080 1000081 1000502 keep unix
```

C.4 CRSP Database Utilities

CRSP provides several utilities that can be used to manipulate CRSPAccess97 databases.

stk_partial

This program creates a new CRSP Access97 `CRSPDB` stock database from an existing database or appends securities from one database to another. It can use a `permlist` or a data type restriction to subset the original database. It takes parameters on input and output databases, input and output set types, data wanted in the new database, and optionally a file containing `PERMNOs` to copy to the new database.

Parameter	Values
Input `CRSPDB` directory path	The directory where the database is stored. Standard environment names can be used such as `$CRSP_DSTK` on UNIX, `crsp_dstk:` on OpenVMS, or `%crsp_dstk%` for Windows NT.
Output `CRSPDB` directory path	The directory where the new database will be stored. This can be an empty directory or an existing directory. If it is an empty directory, a new database will be created. If there is already a `CRSPDB` in that directory, the selected `PERMNO`s will be added to that database.
Input Stock `Setid`	The database type. Use one of 10 if a daily stock database 20 if a monthly stock database
Output Stock `Setid`	The database type. Input and output stock `setid`s should be the same.
Set Wanted	A binary flag to determine the modules that will be supported in the new database. Use 32767 to support all current modules. A module that is not loaded at this time cannot be added later to that database.

Data Wanted	A binary flag to determine which modules will be copied to the new database. Use 32767 to copy all data to the new database. Data wanted must be a subset of set wanted. Individual wanted codes can be summed to load multiple modules. Individual modules codes are:
	`1 = headers`
	`2 = events (names, distributions, shares, delists, Nasdaq info)`
	`4 = lows`
	`8 = highs`
	`16 = prices`
	`32 = total returns`
	`64 = volumes`
	`128 = portfolios`
	`256 = Nasdaq bids`
	`512 = Nasdaq asks`
	`1024 = Returns without dividends`
	`2048 = spread`
	`4096 = Nasdaq number of trades or alternate price dates`
	`8192 = alternate prices`
	`16384 = groups`

Permlist file	The name of a file with a list of `PERMNOs`, one to a line. This parameter is optional. If it is used, only the `PERMNO`s in the input file will have data copied to the new database. If the parameter is not used, all `PERMNO`s in the input database will be copied.

ind_partial

This program creates a new CRSP Access97 CRSPDB index database from an existing database or appends indices from one existing database to another. It can use an indno list or a data type restriction to subset the original database. It takes parameters on input and output databases, input and output set identifiers, data wanted in the new database, and optionally a file containing INDNOs to copy to the new database. Standard stock databases contain stock and indices sets.

Parameter	Values
Input CRSPDB directory path	The directory where the database is stored. Standard environment names can be used such as $CRSP_DSTK on UNIX, crsp_dstk: on OpenVMS, or %crsp_dstk% for Windows NT.
Output CRSPDB directory path	The directory where the new database will be stored. This can be an empty directory or an existing directory. If it is an empty directory, a new database will be created. If there is already a CRSPDB in that directory, the selected INDNOs will be added to that database.
Input Index Setid	The database set type. Use one of 400 if a monthly series 420 if monthly groups 440 if daily series 460 if daily groups
Output Index Setid	The database set type. Input and output index setids should be the same.
Set Wanted	A binary flag to determine the index modules that will be supported in the new database. Use 8191 to support all current modules. A module that is not loaded at this time cannot be added later to that database.
Data Wanted	A binary flag to determine which modules will be copied to the new database. Use 8191 to copy all data to the new database. Data wanted must be a subset of set wanted. Individual wanted codes can be summed to load multiple modules. Individual modules codes are: 1 = headers 2 = rebalancing information for index groups 4 = issue lists 8 = portfolio used counts 16 = portfolio total eligible counts 32 = portfolio used weights 64 = portfolio eligible weights 128 = total returns 256 = capital appreciation returns 512 = income returns 1024 = total return index levels 2048 = capital appreciation index levels 4096 = income return index levels
Indno list file	The name of a file with a list of INDNOs, one to a line. This parameter is optional. If it is used, only the INDNOs in the input file will have data copied to the new database. If the parameter is not used, all INDNOs in the input database will be copied.

crsp_stk_headall

This program creates header files for a database. It is useful primarily for a subset database. If the files are created in the same directory as the database, and the CRSP_MSTK or CRSP_DSTK environment points to the database, the search utilities will function with that database.

Parameters are an input database and setid and four output files. The output files include header information, name history information, header PERMNO/CUSIP cross-reference, and historical PERMNO/CUSIP cross-reference.

Parameter	Values
Input CRSPDB directory path	The directory where the database is stored. Standard environment names can be used such as $CRSP_DSTK on UNIX, crsp_dstk: on OpenVMS, or %crsp_dstk% for Windows NT.
Input Stock Setid	The database type. Use one of 10 if a daily stock database 20 if a monthly stock database
Name history header file	A file name for the name history header file. A file with this name will be created with one line per name history event for each PERMNO. Each line contains PERMNO, PERMCO, name CUSIP, company name, ticker, exchange code, SIC code, and effective range of that name information. If the file is named headfile.dat in the database directory, the dstksearch or mstksearch utility can be used to search the file to find identifiers.
Header file	A file name for the name header file. A file with this name will be created with one line per PERMNO. Each line contains PERMNO, PERMCO, header CUSIP, latest company name, latest ticker, latest exchange code, latest SIC code, and date range.
PERMNO / CUSIP cross-reference file	A file name for a PERMNO/CUSIP historical cross-reference file. A file with this name will be created containing a row with CUSIP and PERMNO for every unique historical CUSIP assignment in the CRSP name history in the database.
PERMNO / header CUSIP cross-reference file	A file name for a PERMNO/CUSIP header file. A file with this name will be created containing a row with header CUSIP and PERMNO for every security in the database.

crsp_ind_headall

This program creates header files for an index database. It is useful primarily for a subset database. If the files are created in the same directory as the database, and the CRSP_MSTK or CRSP_DSTK environment points to the database, the index search utilities will function with that database.

Parameters are an input database and `setid` and one output file. The output file includes `indno`, `setid`, and index description.

Parameter	Values
Input `CRSPDB` directory path	The directory where the database is stored. Standard environment names can be used such as `$CRSP_DSTK` on UNIX, `crsp_dstk:` on OpenVMS, or `%crsp_dstk%` for Windows NT.
Input Stock Setid	The database set type. Use one of 400 if a monthly series 420 if monthly groups 440 if daily series 460 if daily groups
Index header file	A file name for the index header file. A file with this name will be created with one line per index, with `INDNO`, `SETID`, and index description. If the file is named `headind.dat` in the database directory, the *dindsearch* or *mindsearch* utility can be used to search the file to find identifiers.

crsp_stk_scd_load

This program creates secondary indexes for a database. It should be used any time a new subset database is created or edits are made to an existing database. The program can create indexes on multiple keys. The program automatically erases any keys previously stored in the database.

Parameters are an input database and `setid` and a code representing the keys wanted.

Parameter	Values
Input `CRSPDB` directory path	The directory where the database is stored. Standard environment names can be used such as `$CRSP_DSTK` on UNIX, `crsp_dstk:` on OpenVMS, or `%crsp_dstk%` for Windows NT.
Input Stock Setid	The database type. Use one of 10 if a daily stock database 20 if a monthly stock database
Input wanted flag	The data required to build the index. 1 if only header data are needed to build index 3 if header data and events data are needed to build index
Index wanted flag	A binary flag to select the indices to build. 1 = `PERMCO` (only header needed) 2 = header CUSIP (only header needed) 4 = historical CUSIP (header and names needed) 8 = historical SIC (header and names needed) 16 = header ticker; active securities at the cut date of the file (only header needed) Use 31 to build all secondary indices or add the flags for one or more types.
Name of permlist file (optional)	If this parameter is supplied, it must be the name of a text file containing `PERMNO`s, one per line. If the parameter is not used, all securities in the database will be used to create the secondary indexes. If the parameter is supplied, the indexes will only be based on the securities in the permlist and other securities will be unavailable using a secondary index read.

Examples

On Windows NT, the following commands can create a personal subset monthly database with only headers, events, prices, and returns for the PERMNOs in a file perms.txt, then create header files and secondary indexes for the database. The output directory must exist and the perms.txt file must exist.

```
stk_partial %crsp_mstk% c:\mydir\ 20 20 32767 51 perms.txt

crsp_stk_headall c:\mydir\ 20 c:\mydir\headfile.dat c:\mydir\cheadfile.dat
c:\mydir\permcusip.dat c:\mydir\cpermcusip.dat

crsp_stk_scd_load c:\mydir\ 20 3 31
```

If the environment variable crsp_mstk is set to c:\mydir\ in the user environment variable of the settings / control panel / system window, the standard utilities will use the new subset database as the standard monthly database.

On UNIX, the following commands provide the same result:

```
stk_partial $CRSP_MSTK /mydisk/mydir/ 20 20 32767 51 perms.txt

crsp_stk_headall /mydisk/mydir/ 20 /mydisk/mydir/headfile.dat /mydisk/mydir/cheadfile.dat
/mydisk/mydir/permcusip.dat /mydisk/mydir/cpermcusip.dat

crsp_stk_scd_load /mydisk/mydir 20 3 31

setenv CRSP_MSTK /mydisk/mydir    (csh)
or
CRSP_MSTK=/mydisk/mydir
export CRSP_MSTK              (ksh)
```

On OpenVMS, the following commands provide the same result:

```
stk_partial crsp_mstk: mydisk:[mydir] 20 20 32767 51 perms.txt

crsp_stk_headall mydisk:[mydir] 20 mydisk:[mydir]headfile.dat –
mydisk:[mydir]cheadfile.dat mydisk:[mydir]permcusip.dat mydisk:[mydir]cpermcusip.dat

crsp_stk_scd_load mydisk:[mydir] 20 3 31

define crsp_mstk mydisk:[mydir]
```

C.5. *ts_print* Maintenance

The *ts_print* data utility creates temporary files in the user log directory setup on the system during installation. It might become necessary to clean this directory periodically if circumstances arise where *ts_print* is unable to clean these files.

The log directory is defined during installation. The environment variable name on Unix or Windows NT systems is `CRSP_ENV_ULOG`. The logical directory name on OpenVMS is `CRSPDB_ULOGDIR:.` Any user with the proper permissions can erase all files in this directory if necessary.

D. DATA AVAILABILITY

D.1 CRSPAccess97 Products

CRSPAccess97 Products

Product Code	Description	INDNOs provided	PORTTYPEs provided
DA	Daily NYSE/AMEX/Nasdaq	1000080, 1000081, 1000502, 1000503	1
MA	Monthly NYSE/AMEX/Nasdaq	1000080, 1000081, 1000502, 1000503	1
DX	Daily NYSE/AMEX	1000040, 1000041, 1000502	2
MZ	Monthly NYSE/AMEX	1000040, 1000041, 1000502	2
IX	Indices File / Portfolio Assignments	All	All

D.2 Data Availability Chart

This table shows which types of data are present in the various CRSP stock files. In general, the information for securities on AMEX becomes available on July 2, 1962, the monthly information for NYSE securities on December 31, 1925, the daily information for NYSE securities on July 2, 1962, and the information for securities on Nasdaq becomes available on December 14, 1972. The legend and footnotes follow the table.

C Objects	FORTRAN Structures	Daily	Monthly
header_row	/HEADER/	√	√
names_arr	NAMES(,)	√	√
dists_arr	DISTS(,)	~	~
shares_arr	SHARES(,)	~	~
delist_arr	DELIST(,)	√	√
nasdin_arr	NASDIN(,)	~[1]	~[1]
bidlo_ts	BIDLO()	~[2]	~[3]
askhi_ts	ASKHI()	~[2]	~[3]
prc_ts	PRC()	√	√
vol_ts	VOL()	~[2, 1]	~[2, 1]
ret_ts	RET()	√	√
retx_ts	RETX()	√	√
spread_ts	PRC2()	•	~
port_ts[ ]	PRTNUM(,)	√	√
	YRVAL(,)	√	√
numtrd_ts	NUMTRD()	~[4]	~[5]
bid_ts	BID()	~[4]	~[4]
ask_ts	ASK()	~[4]	~[4]
altprc_ts	-	•	~[5]

The data is always present in the file for each security. See the version specific release notes for data ranges.

~ This data is present for a security if that security has that type of data. The data may not be available for a security if no events of that type occur for the security, or data of that type was not collected during the trading range of that security.

• The data is never present in the file.

1. Our Nasdaq data source changed for data beginning in November, 1982. Volumes and Nasdaq information structures are unavailable for any securities that stopped trading on Nasdaq prior to November, 1982. No volumes or Nasdaq information structures are available before that time for any Nasdaq securities.

2. NYSE/AMEX daily BIDLO, ASKHI, volumes and NYSE/AMEX monthly volumes are complete to July 2, 1962. There are no monthly VOL records for securities that stopped trading before that time. Only three NYSE/AMEX daily securities currently have no BIDLO, ASKHI, or volume data.

3. Monthly BIDLO and monthly ASKHI are derived from the daily data, which begins in July, 1962. Therefore monthly BIDLO and monthly ASKHI are unavailable for any security that stopped trading on NYSE prior to July, 1962 and all AMEX securities. No BIDLO or ASKHI data exists before that time for any NYSE monthly securities.

4. Nasdaq Bid, Ask and Number of trades began in November, 1982 for the National Market companies and are available for all Nasdaq companies as of June, 1992.

5. Alternate prices (altprc_ts) and dates of alternate prices (numtrd_ts) are derived from the daily data, which begins in July, 1962 on NYSE/AMEX and December, 1972 on Nasdaq. NUMTRD contains alternate price dates in monthly files only.

INDEX

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

Y