TMX - CFMRC

Toronto
Stock
Exchange
-
Canadian
Financial
Markets
Research
Center

SUMMARY INFORMATION DATABASE PRODUCTS

Version 1.3a
July 17, 2008

TORONTO STOCK EXCHANGE - CANADIAN FINANCIAL MARKETS RESEARCH CENTER

SUMMARY INFORMATION DATA PRODUCTS

USER'S MANUAL

August 3,1993
Revised: July 17, 2008

Table of Contents

I. Introduction

The Toronto Stock Exchange - Canadian Financial Markets Research Center (TMX-CFMRC) Summary Information Databases are a complete amalgamation and substantial enhancement of the TMXUWO database products. Highlights of the improvements are:

1) expanded daily data set includes opening as well as closing data for: prices; bids; asks; trades; and volumes, amongst other information.

2) expanded monthly data set includes: betas; earnings per share; volume; and transactions, amongst other data.

3) expanded daily and monthly index data sets which include: a selection of interest and exchange rates; TMX Group indices; new under- and over- $2.00 indices, as well as other data.

4) a convenient, easy-to-use, DOS user interface program.

5) collection of five ASCII data files conveniently written on a single CD-ROM disk. These disks are compatible with any ISO-9660 CD-ROM capable system.

Acknowledgments

Data on interest and foreign exchange rates has been made available through the generosity and kind assistance of the Bank of Canada and Statistics Canada. Selected data from Statistics Canada's CANSIM data bank are published with authority of Statistics Canada. CANSIM is Statistics Canada's official Time Series data bank service. For further information about the services contact your nearest Statistics Canada office or call 613-951-8200. Where appropriate, the CANSIM data series number is noted.

Disclaimer of Warranties

NEITHER CFMRC NOR ANY THIRD-PARTY SUPPLIERS OF DATA OR SOFTWARE TO CFMRC MAKE ANY WARRANTY, EXPRESS OR IMPLIED, AS TO RESULTS TO BE OBTAINED BY SUBSCRIBER OR OTHERS FROM THE USE OF THE DATA BASES OR ANY SOFTWARE PROVIDED HEREUNDER, AND THERE ARE NO EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE. The information contained in the Data Bases shall be obtained from sources considered by CFMRC to be reliable, but the accuracy and completeness thereof are not guaranteed and CFMRC and its third-party suppliers shall have no liability for errors or omissions with respect to the Data Bases or their delivery, regardless of the cause or source of such error or omission.

I.1 Using the CFMRC Summary Information Databases

There are two ways to access the CFMRC Databases:

1) using the DOS-based, menu-driven CFMRC software provided on the CD-ROM.

2) writing specialized programs to access the databases.

The latter alternative is for users on non-DOS platforms or users who wish to tailor their computations. Most users will probably opt for the ease and simplicity of the DOS interface.

The files included are:

DAILY.IDX	- Daily calendar/TSX index file
DATA.TSE	- Main data file containing all security specific data
MONTHLY.IDX	- Monthly calendar/TSX index file
TICKERS.DAT	- List of tickers/names and their offsets in the main data file

Each has information on the same list of tickers. For example, if AB is in the file DATA.TSE, then its name and offset is in TICKERS.DAT.

The databases contain daily and monthly Toronto Stock Exchange (TSX) trading information about specific securities as well as information on "price adjustments" such as dividends, stock splits, recapitalizations, etc. (in DATA.TSE). The daily and monthly "index" files (DAILY.IDX and MONTHLY.IDX) contain information on daily and monthly index levels as well as selected other financial markets information.

Installation instructions for DOS systems are outlined in section II. Section III discusses and defines the various data fields included in the CFMRC Summary Information Databases. Section IV outlines the format of the data files which will be very important to those users writing their own code. Appendix A contains code fragments in C to read these files.

II. DOS Interface Program

The CFMRC Summary Information Database products come with an easy to use, menu driven interface program for use on DOS computers. The program is called CFMRC.EXE and it and its various support files can be found in the DOS subdirectory on the CD-ROM. The files and their purposes are:

CFMRC.EXE	- executable DOS interface program
CFMRC.HLP	- HELP file for CFMRC.EXE (ASCII text)
DAILY.BIN	- DOS binary form of DAILY.IDX
DICTION.BIN	- required index file
MONTHLY.BIN	- DOS binary form of MONTHLY.IDX
TICKERS.BIN	- DOS binary form of TICKERS.DAT

Files ending in .BIN are DOS binary files that contain exactly the same information as their respective ASCII database files. Because they are binary however, the program CFMRC.EXE can read them much more quickly.

II.1 Installation

Installing the interface program is quite easy. First, determine the drive letter or directory path of your CD-ROM reader - suppose it is drive G:\. Issue the command

```
SET CFMRC=G:\
```

to tell DOS that the "Environment Variable" CFMRC should point to the G: drive (your CD-ROM). When you start the CFMRC.EXE program, it will ask DOS about this environment variable and use it to find the raw and binary data files. You can automatically set this variable every time you turn your computer on by adding the above command to your AUTOEXEC.BAT file (see your DOS manual for more details).

To start the interface program, change to the directory containing the program (G:\DOS in this example) and type the command

```
CFMRC
```

Alternatively, you can copy the program CFMRC.EXE to a directory in your program search path (see PATH in your DOS manual) or you can add G:\DOS to your search path.

II.2 Alternate Installation

Modern hard disks are much faster than CD-ROM drives and the CFMRC.EXE program can take advantage of this extra speed to improve its performance.

To do this, create a separate directory on your hard disk for the CFMRC DOS files - lets say it's C:\PROGRAMS\CFMRC. Copy all of the DOS files from the CD-ROM DOS directory to the new program directory using the command

```
COPY G:\DOS\*.* C:\PROGRAMS\CFMRC
```

In addition to the SET command given above, you must also do

```
SET CFMRC_DOS_DIR=C:\PROGRAMS\CFMRC
```

This tells the interface program to look for its DOS files in the new directory (by default it will look in the subdirectory named DOS under the directory named in the SET CFMRC command). G:\ is still searched for the ASCII data files.

To take full advantage of the flexibility of DOS and the interface program, you should add C:\PROGRAMS\CFMRC to your program search path and both SET commands to your AUTOEXEC.BAT file.

II.3 Support

If you have any questions, comments, suggestions, or criticisms, please feel free to contact the TSX at:

TSX Market Data Services
Exchange Tower
2 First Canadian Place
Toronto, Ontario
M5X 1J2

or E-mail to:

marketdata@TSX.com

Phone: (416) 947-4452

III. Data Description

There are six different types of data in the CFMRC databases:

1) Daily index data
2) Monthly index data
3) Stock header information
4) Daily trading data
5) Monthly trading data
6) Price adjustment data

Each is discussed below.

Missing Data

Unless otherwise noted, missing data is flagged with -9.0 for floating point (exceptions are monthly closing prices which use 0.0, and EPS which uses -9999.0) data or -9 for integer data. Data is missing if there is no information available. Thus, if there were no trades in a day for a certain stock, the number of trades would be set to 0 and the closing price (for example) would be 0.00 - that is, there is information. Only if there is no trading information at all would trades be set to -9 and closing price to -9.00.

Data Errors

While every effort has been made to eliminate data errors, suspect or erroneous data still creeps into the database. In cases where data is clearly wrong or suspect, data that is only meaningful when positive (shares outstanding or prices, for example) has been flagged by multiplying by -1.0. The user is free to decide on the usefulness of these data and can recover them by multiplying by -1.0 again. Again, monthly closing prices are an exception. See below.

III.1 Daily Index Data (Daily Calendar)

The CFMRC databases use relative calendars. That is, each trading day or month is given a sequential day or month number called its "relative calendar" number.

Daily data in the CFMRC databases begins on January 2, 1975 and so 750102 has a relative daily calendar number of 1. 750103 has a relative daily calendar number of 2 but January 4 &

5, 1975 fell on a Saturday and Sunday so these are non-trading days. Relative day 3 is 750106. To determine the number of trading days between two calendar dates (counting both dates), subtract the relative daily calendar number of the start date from the relative daily calendar number of the end date and add 1. Because of weekends and holidays, there is no simple algorithm for mapping to/from relative calendar number from/to Julian calendar date.

Data available for each day in the relative calendar is:

Date - the Julian calendar date corresponding to this relative daily calendar number.

S&P/TSX Composite Price Index - the closing S&P/TSX Composite Price Index. This index is based solely on security prices. See the TSX Review (green pages) for securities included in this index and their weights. Daily data for TSX Indices begins on 770103 (relative day number 506).

S&P/TSX Composite Total Return Index - the closing S&P/TSX Composite Total Return Index based on prices and distributions (stocks, dividends, etc.). See the TSX Review (green pages) for securities included in this index and their weights. Daily data for TSX Indices begins on 770103 (relative day number 506).

S&P/TSX Sector Indices: Price & Total Return - daily closing values for all ten S&P/TSX Sector based on closing prices (Price) or including distributions (Total Return). See the TSX Review (green pages) for securities included in each sector and their weights. Daily data for S&P/TSX Sectors begins on 871231. The present sectors are:

Energy Sector
Materials Sector
Industrials Sector
Consumer Discretionary Sector
Consumer Staples Sector
Health Care Sector
Financials Sector
Information Technology Sector
Telecommunications Services Sector
Utilities Sector

Also included as of 020603 are the following S&P/TSX Indices (Price and Total Return):

S&P/TSX 60 Index
S&P/TSX Canadian Midcap Index
S&P/TSX Canadian Smallcap Index
S&P/TSX Venture Composite Index

CFMRC Equal Weighted Index - the average daily return for all domestic common equities in the CFMRC database. This is the sum of all defined common equity returns divided by the number of valid equity returns (Canadian based firms only). Returns used in this index are fully adjusted for distributions. See Daily Returns below for the definition of a valid return.

CFMRC Value Weighted Index - the market value weighted average daily return for all domestic common equities in the CFMRC database. This is the sum of all products of defined common equity returns with their market weights. A security's market weight is defined as its market value at the beginning of the current month (shares outstanding times closing price on the last trading day in the previous month) divided by the market value of all securities included in the index (Canadian firms only). Returns used in this index are fully adjusted for distributions. See Daily Returns below for the definition of a valid return.

CFMRC Under $2 Equal Weighted Index - similar to the CFMRC Equal Weighted Index but includes only those securities for which prices fell below $2.00 on either the current or previous trading day.

CFMRC Under $2 Value Weighted Index - similar to the CFMRC Value Weighted Index but includes only those securities for which prices fell below $2.00 on either the current or previous trading day.

CFMRC Over $2 Equal Weighted Index - similar to the CFMRC Equal Weighted Index but includes only those securities for which prices did not fall below $2.00 on either the current or previous trading day.

CFMRC Over $2 Value Weighted Index - similar to the CFMRC Value Weighted Index but includes only those securities for which prices did not fall below $2.00 on either the current or previous trading day.

Call Loan Rate - the Canadian over-night wholesale money market interest rate. This is CANSIM series B114011.

Foreign Exchange Rate (CA$/US$) - the daily closing spot exchange rate expressed as Canadian dollars per U.S. dollar. This is CANSIM series B100014.

Historical Daily Index Data

The old Daily Calendar file is included with the database in a flat file named OLD_DAILY.IDX. All daily data fields described above are present in this file except for the S&P/TSX Sector Indices which are replaced with the old S&P/TSX Composite Group Indices, as listed below. This file cannot be viewed with the browser.

S&P/TSX Group Indices: Price & Total Return - daily closing values for all fourteen S&P/TSX Group Indices based on closing prices (Price) or including distributions (Total Return). See the TSX Review (green pages) for securities included in each group and their weights. Daily data for TSX Indices begins on 770103 and ends on 021231. The groups are:

Metals & Minerals
Gold & Silver
Oil & Gas
Paper & Forest Products
Consumer Products
Industrial Products
Real Estate & Construction
Transportation & Environmental
Pipelines
Utilities
Communications & Media
Merchandising
Financial Services
Conglomerates

III.2 Monthly Index Data (Monthly Calendar)

The CFMRC databases use relative calendars. That is, each trading day or month is given a sequential day or month number called its "relative calendar" number.

Monthly data in the CFMRC databases begins with January, 1950 and so 5001 has a relative monthly calendar number of 1. Unlike the daily calendar, relative monthly calendar numbers can be easily computed. If yymm (where yy is the last two digits of the year and mm is the month number - i.e. 7208 for Aug/78) is the month in question, then

month # = ((INT(yymm/100)-50)*12)+(yymm-(INT(yymm/100)*100))

The CFMRC databases actually contain price observations (no other data) for many securities for December, 1949. Using the above formula, 4912 has a relative monthly calendar number of 0.

Data available for each month in the relative calendar is:

Month End - the Julian calendar date of the last trading day in the month. For months prior to

7501, the date is given as yymm00 (i.e. no day).

S&P/TSX Composite Price Index - the closing S&P/TSX Composite Price Index. This index is based solely on security prices. See the TSX Review (green pages) for securities included in this index and their weights. Monthly data for TSX Indices begins on 5601 (relative month number 73).

S&P/TSX Composite Total Return Index - the closing S&P/TSX Composite Total Return Index based on prices and distributions (stocks, dividends, etc.). See the TSX Review (green pages) for securities included in this index and their weights. Monthly data for TSX Indices begins on 5601 (relative month number 73).

S&P/TSX Sector Indices: Price & Total Return - monthly closing values for all ten S&P/TSX Sector based on closing prices (Price) or including distributions (Total Return). See the TSX Review (green pages) for securities included in each sector and their weights. Monthly data for S&P/TSX Sectors begins on 871231. The present sectors are:

Energy Sector
Materials Sector
Industrials Sector
Consumer Discretionary Sector
Consumer Staples Sector
Health Care Sector
Financials Sector
Information Technology Sector
Telecommunications Services Sector
Utilities Sector

Also included as of 020603 are the following S&P/TSX Indices (Price and Total Return):

S&P/TSX 60 Index
S&P/TSX Canadian Midcap Index
S&P/TSX Canadian Smallcap Index
S&P/TSX Venture Composite Index

CFMRC Equal Weighted Index - the average monthly return for all domestic common equities in the CFMRC database. This is the sum of all defined common equity returns divided by the number of valid equity returns (Canadian based firms only). Returns used in this index are fully adjusted for distributions. See Monthly Returns below for the definition of a valid return.

CFMRC Value Weighted Index - the market value weighted average monthly return for all domestic common equities in the CFMRC database. This is the sum of all products of defined

common equity returns with their market weights. A security's market weight is defined as its market value at the beginning of the current month (shares outstanding times closing price on the last trading day in the previous month) divided by the market value of all securities included in the index (Canadian firms only). Returns used in this index are fully adjusted for distributions. See Monthly Returns below for the definition of a valid return.

T-Bill Rate - the average yield on 91 day Government of Canada T-bills based on the weekly auctions. This is CANSIM series B14007.

Long Term Government Bond Rate - the average yield on a portfolio of 10+ year Government of Canada (GOC) Bonds. This is CANSIM series B14013.

Corporate Bond Rate - the average yield on a portfolio of high grade, long term corporate bonds. This is CANSIM series B14048.

Foreign Exchange Rate (CA$/US$) - the average noon spot exchange rate expressed as Canadian dollars per U.S. dollar. CANSIM series B3400.

T-Bill Return - the return on a 91 day T-bill purchased at the end of last month and sold at the end of this month. If r(t) is the yield (in percent) at the end of month t, then the price last month, P(t-1), of a T-bill with 91 days to maturity is

$$P(t-1) = 100/[100 + r(t-1)*91/365]$$

and the price today of that same bill with today's yield and only 61 days to maturity is

$$P(t) = 100/[100 + r(t)*61/365]$$

The return, R(t), is

$$R(t) = [P(t) - P(t-1)]/P(t-1)$$

Long Term Government Bond Return - the return on a long term GOC bond with an assumed term to maturity of 17 years purchased at the end of last month (at par or P(t-1) = 100) and sold at the end of this month.

If Y(t) is the yield at the end of month t, then the price, P(t) is simply the present value of the future semi-annual interest payments plus the discounted terminal or par value.

$$P(t) = [Y(t-1)/2]/[1+\{Y(t)/100\}/2]^{**}1 + [Y(t-1)/2]/[1+\{Y(t)/100\}/2]^{**}2 + \ldots + [Y(t-1)/2]/[1+\{Y(t)/100\}/2]^{**}2M + 100/[1+Y(t)/100/2]^{**}2M$$

where M is the number of years to maturity (=17 in this case). The interest received in month t is Y(t-1)/12 so the total return is

$$R(t) = [P(t) - 100 + \{Y(t-1)/12\}]/100$$

Historical Monthly Index Data

The old Monthly Calendar file is included with the database in a flat file named OLD_MONTHLY.IDX. All monthly data fields described above are present in this file except for the S&P/TSX Sector Indices which are replaced with the old S&P/TSX Composite Group Indices, as listed below. This file cannot be viewed with the browser.

TSX Group Indices: Price & Total Return - monthly closing values for all fourteen TSX Group Indices based on closing prices (Price) or including distributions (Total Return). See the TSX Review (green pages) for securities included in each group and their weights. The present groups are:

Metals & Minerals
Gold & Silver
Oil & Gas
Paper & Forest Products
Consumer Products
Industrial Products
Real Estate & Construction
Transportation & Environmental
Pipelines
Utilities
Communications & Media
Merchandising
Financial Services
Conglomerates

III.3 Stock Header Information

Stock Name - the most recent name of the firm issuing this security while this security was still listed. If a firm changed its name after delisting one of its securities, the change will not be reflected in the database record for that delisted security. Any other securities issued by the firm that were actively listed when the name change occurred will reflect the name change.

Ticker - the most recent ticker symbol for this security.

Usage Number - over time, ticker symbols are reused by the TSX. To uniquely identify securities, the CFMRC has assigned a usage number to each use of a particular ticker which indicates the number of times the ticker has been used before in the CFMRC databases.

For example, between 450711 and 730704, Atlantic Sugar Refineries, Inc. used the ticker AC on the TSX. Since this was the first usage of that symbol in the database, it was assigned the usage number 0. So, AC (0) refers to Atlantic Sugar Refineries, Inc. In 1988, the same ticker was issued to Air Canada but since it had been used before, Air Canada was given the usage number 1 so the unique ticker symbol is AC (1).

Business - a brief description, defined by the TSX, of the type of business this firm is involved in.

Foreign Based - a flag to identify foreign-based versus domestic firms. If this is a domestic firm, this flag is '0' (the character zero). If it is '1', the firm is foreign based and will be excluded from all CFMRC index calculations.

Starting Relative Calendar Day - the relative daily calendar number of the first day of data for this security.

Number of Days - number of daily data records.

Starting Relative Calendar Month - the relative monthly calendar number of the first month of data for this security.

Number of Months - number of monthly data records.

Number of Price Adjustments - number of price adjustment records (dividends, splits, etc.).

III.4 Daily Trading Data

Daily data in the CFMRC databases begins with January 2, 1975. However, data for the time between 750102 and 770228 inclusive is based on round lot trades only. As of 770301, data is based on ALL trades (including odd lots). Information reported in the TSX Review is based on round lots only.

Opening Price - the price of the first transaction after the market opened. The TSX opened at 10:00 A.M. before 850930 but has opened at 9:30 A.M. since.

Opening Bid - the last bid price before the market opened. If there were no quotes prior to the market opening, this is the first bid price since the open. The TSX opened at 10:00 A.M. before 850930 but has opened at 9:30 A.M. since. This data is not available prior to 770301.

Opening Ask - the last ask price before the market opened. If there were no quotes prior to the market opening, this is the first ask price since the open. The TSX opened at 10:00 A.M. before 850930 but has opened at 9:30 A.M. since. This data is not available prior to 770301.

Closing Price - the price of the last transaction of the day.

Closing Bid - the bid price for the last quote of the day before the market closes. This data is not available prior to 770301.

Closing Ask - the ask price for the last quote of the day before the market closes. This data is not available prior to 770301.

High - the highest price at which this security traded during the day.

Low - the lowest price at which this security traded during the day.

Return - the fully adjusted daily return calculated as if the security was purchased at the close yesterday and sold at the close today. Let P(t) be today's closing price, D(t) be the cash or cash equivalent dividend (in $) paid today (that is, today is the ex-dividend date) and S(t) be the stock split factor for a stock dividend or split today. The return is

$$R(t) = \{[P(t)+D(t)]*S(t) - P(t-1)\}/P(t-1)$$

If there was no cash dividend, D(t)=0.0 and if there was no stock dividend or split, S(t)=1.0. If a security was reclassified at t or t-1, the return will be recorded as -9.0 for both. Since open and closing prices are now available in the database, the user can compute quasi-split factors if desired.

Transactions - the number of transactions in this security on day t.

Volume - total volume transacted in this stock on day t.

Quotes - the number of valid market quotes posted throughout the day for this stock. This data is not available prior to 770301.

Quote Changes - number of times the valid market quoted prices changed throughout the day for this security. Since prices do not always change when a new quote is posted, this number will always be less than or equal to Quotes (above). This data is not available prior to 770301.

III.5 Monthly Trading Data

Closing Price - the last trade price from the last two days of trading in the given month. If a security did not trade within the last two trading days of the month, -1 times the mean of the last bid and ask quote in the last two days of trading is reported here. If there were no trades or quotes in the last two trading days in the month, 0.0 is reported.

Prices are arbitrarily limited to the last two trading days to ensure that monthly prices are not "stale" and that monthly returns are calculated over a one month period. This also has the effect of eliminating infrequently traded stocks from the CFMRC indices since they rarely have monthly returns.

Return - the fully adjusted return based on purchasing a share at last month's close and selling it at this month's close. If either month's closing price is less than or equal to 0.0 or an undefined dividend/recapitalization occurred within the month, the return is undefined (-9.0).

EPS - trailing twelve months earnings per share as reported by the TSX. Missing data is flagged with -9999.0.

Beta - the rolling average beta from the CAPM. Given $R_f(t)$ and $R_m(t)$ the T-bill return and return on the TSX 300 Total Return Index (see above) and the security return, R(t), beta is calculated as:

$$\log[R(t) - R_f(t)] = a + \text{Beta}*\log[R_m(t) - R_f(t)]$$

When there is no valid return due to missing prices, the mean of the bid and ask is used as a proxy (-1 times the monthly closing price, if valid). A minimum of 24 months of returns over the past 60 months are required before a beta is calculated.

Transactions - total number of transactions in this month.

Total Volume - total volume transacted this month in 100's of shares.

Shares Outstanding - total shares outstanding at month end for this security in 100's of shares.

Dividend Flag - a flag to indicate whether a price adjustment occurred this month. 0 if there were no price adjustments and 1 if there were.

III.6 Price Adjustments Data

Dividend - a decimal amount whose meaning depends on the contents of the Dividend Type flags.

Dividend Type - three flags that indicate the type of dividend and the format of the Dividend field.

The first flag identifies the type of dividend or capital change:

0 - Cash dividend - Regular
1 - Cash dividend - Extra
2 - Stock dividend
3 - Stock in lieu of cash
4 - Stock right
5 - Stock split
6 - Recapitalization
9 - Meaningless data. Record should be ignored.

The second flag identifies the currency in which the dividend was paid:

0 - Canadian dollars
1 - United States dollars
2 - Other foreign currency

The last flag identifies the format of the data in the Dividend field:

0 - Dollar value
1 - Decimal - fraction of a share or share multiple

If the last flag is 1, the type of number in the Dividend field depends on the type of dividend as indicated by flag one. If this is a stock dividend, stock in lieu dividend or stock right, then the number represents a fraction of a share. Otherwise, the number in the Dividend field is a ratio (i.e. 2:1 is 2.0).

Ex-dividend Date - the date upon which the security began trading ex-dividend or ex-split.

Recapitalizations - for recapitalizations and reclassifications, the dividend field is -9.0. Since Since opening and closing prices are now given in the database, the user can compute quasi-split factors if desired.

IV. File Formats

For convenience, Fortran FORMAT statement notation will be used to summarize the format of each data record in the various data files. The field specifiers are:

Ann	- nn spaces of character data
Fx.y	- x spaces of floating point data which includes a decimal point ('.') and y decimal places.
Inn	- nn spaces of integer data
nnX	- nn spaces are skipped (but may contain meaningless characters).

Format specifiers are separated by commas and my be grouped with parentheses '( )'. Any specifier or group of specifiers may be repeated y times by preceding it with y. For example, the format string

(A30,2I5,3(1X,F10.4))

defines a 30 character long string, 2 five digit integers and 3 floating point numbers of ten digits each (counting the decimal point and four decimal places) each preceded by one space.

In the data files, numbers are right justified within their fields and strings are left justified. Leading zeros are suppressed.

IV.1 Daily Calendar/Index File - DAILY.IDX and OLD_DAILY.IDX

Each record in the daily index file represents one day and contains 360 data characters and a terminating linefeed (hex 0A, decimal 10) for a total of 361 characters per record. Records are organized sequentially starting with relative daily calendar number one. Each record contains:

1) Relative day number
2) Date
3) S&P/TSX Composite Daily Price Index
4) S&P/TSX Composite Daily Total Return Index
5) Sector 10/Group 1 Daily Price Index
6) Sector10/Group 1 Daily Total Return Index
7) Sector 15/Group 2 Daily Price Index
8) Sector 15/Group 2 Daily Total Return Index
9) Sector 20/Group 3 Daily Price Index
10) Sector 20/Group 3 Daily Total Return Index
11) Sector 25/Group 4 Daily Price Index

12) Sector 25/Group 4 Daily Total Return Index
13) Sector 30/Group 5 Daily Price Index
14) Sector 30/Group 5 Daily Total Return Index
15) Sector 35/Group 6 Daily Price Index
16) Sector 35/Group 6 Daily Total Return Index
17) Sector 40/Group 7 Daily Price Index
18) Sector 40/Group 7 Daily Total Return Index
19) Sector 45/Group 8 Daily Price Index
20) Sector 45/Group 8 Daily Total Return Index
21) Sector 50/Group 9 Daily Price Index
22) Sector 50/Group 9 Daily Total Return Index
23) Sector 55/Group 10 Daily Price Index
24) Sector 55/Group 10 Daily Total Return Index
25) S&P/TSX 60/Group 11 Daily Price Index
26) S&P/TSX 60/Group 11 Daily Total Return Index
27) S&P/TSX Mid Cap/Group 12 Daily Price Index
28) S&P/TSX Mid Cap/Group 12 Daily Total Return Index
29) S&P/TSX Small Cap/Group 13 Daily Price Index
30) S&P/TSX Small Cap/Group 13 Daily Total Return Index
31) S&P/TSX Venture/Group 14 Daily Price Index
32) S&P/TSX Venture/Group 14 Daily Total Return Index
33) Call Loan Interest Rate
34) Daily Foreign Exchange Rate
35) CFMRC Daily Equal Weighted Index
36) CFMRC Daily Value Weighted Index
37) CFMRC Equal Weighted Under $2 Index
38) CFMRC Value Weighted Under $2 Index
39) CFMRC Equal Weighted Over $2 Index
40) CFMRC Value Weighted Over $2 Index

The format is:

```
(I5,1X,I6,2(1X,F8.2),14(1X,F8.2,1X,F8.2),2(1X,F8.5),3(1X,F9.6,1X,F9.6))
```

IV.2 Data File - DATA.DAT

Records in the main data file are 112 characters long including the trailing linefeed. There is one header record followed by one group of data records for each security in the database. The file header is simply a 97 character string identifying the version and date of the database. The format is (A97).

Each group contains all the data records for a specific security. The order of the data records is:

1) 2 security header records
2) variable number of daily data records
3) variable number of monthly data records
4) variable number of price adjustment records

IV.2.1 Security Header Records

The first header record contains

1) Ticker
2) Usage Number
3) Name
4) Business
5) Foreign Flag

in the following format:

(A8,1X,I1,11X,A45,1X,A24,1X,I1)

The second header record describes the rest of the group specifying

1) Daily data start date relative calendar number
2) Number of daily data records
3) Monthly data start date relative calendar number
4) Number of monthly data records
5) Number of price adjustments

The format is: (5(I5,1X))

IV.2.2 Daily Data Records

The first record immediately following the second security header line is the first daily data record if the number of daily data records in the second header line is greater than 0. The date for this record can be found by looking up the Julian calendar date for the daily data start date relative calendar number in the relative calendar/index file. Subsequent daily data records have sequential daily relative calendar numbers.

Each daily record contains

1) Opening price
2) Opening bid
3) Opening ask
4) Closing price
5) Closing bid
6) Closing ask
7) High
8) Low
9) Transactions
10) Volume
11) Quotes
12) Quote changes
13) Return

in the following format

(8F9.3,I8,I9,2I6,F10.6)

IV.2.3 Monthly Data Records

The first record immediately following the last daily data record is the first monthly data record if the number of monthly data records in the second header line is greater than 0. As with the daily data records, the dates associated with monthly records can be found by looking them up in the monthly calendar/index file.

Each monthly record contains

1) Closing price
2) Total transactions
3) Total volume
4) Shares outstanding
5) Earnings per share
6) Beta
7) Monthly return
8) Dividend flag

The format is

(F9.3,I9,I10,I11,F13.6,F7.3,F11.6,1X,I1)

IV.2.4 Price Adjustment Data Records

Each price adjustment record contains from one to five data clusters and each cluster contains data on one price adjustment. Clusters contain

1) Dividend
2) Dividend flag 1
3) Dividend flag 2
4) Dividend flag 3
5) Ex-dividend date

formatted as (F11.6,1X,3I1,1X,I6)

so the format for a complete record is

(5(F11.6,1X,3I1,1X,I6))

Note that there is one cluster for every price adjustment indicated in the second stock header record. If this number, say N, is greater than 0, there will be INT((N-1)/5)+1 price adjustment records for this security.

IV.3 Monthly Calendar/Index File - MONTHLY.IDX and OLD_MONTHLY.IDX

Each record in the monthly index file represents one day and contains 357 data characters and a terminating linefeed (hex 0A, decimal 10) for a total of 358 characters per record. Records are organized sequentially starting with relative monthly calendar number one. Each record contains:

1) Relative month number
2) End of Month Date
3) 91 Day T-Bill Rate
4) 30 day Return on T-Bills
5) Long Term Government Bond Rate
6) Corporate Bond Rate
7) Foreign Exchange Rate (Cdn$/US$)
8) S&P/TSX Composite Monthly Price Index
9) S&P/TSX Composite Monthly Total Return Index
10) Sector 10/Group 1 Monthly Price Index
11) Sector 10/Group 1 Monthly Total Return Index

12) Sector 15/Group 2 Monthly Price Index
13) Sector 15/Group 2 Monthly Total Return Index
14) Sector 20/Group 3 Monthly Price Index
15) Sector 20/Group 3 Monthly Total Return Index
16) Sector 25/Group 4 Monthly Price Index
17) Sector 25/Group 4 Monthly Total Return Index
18) Sector 30/Group 5 Monthly Price Index
19) Sector 30/Group 5 Monthly Total Return Index
20) Sector 35/Group 6 Monthly Price Index
21) Sector 35/Group 6 Monthly Total Return Index
22) Sector 40/Group 7 Monthly Price Index
23) Sector 40/Group 7 Monthly Total Return Index
24) Sector 45/Group 8 Monthly Price Index
25) Sector 45/Group 8 Monthly Total Return Index
26) Sector 50/Group 9 Monthly Price Index
27) Sector 50/Group 9 Monthly Total Return Index
28) Sector 55/Group 10 Monthly Price Index
29) Sector 55/Group 10 Monthly Total Return Index
30) S&P/TSX 60/Group 11 Monthly Price Index
31) S&P/TSX 60/Group 11 Monthly Total Return Index
32) S&P/TSX Mid Cap/Group 12 Monthly Price Index
33) S&P/TSX Mid Cap/Group 12 Monthly Total Return Index
34) S&P/TSX Small Cap/Group 13 Monthly Price Index
35) S&P/TSX Small Cap/Group 13 Monthly Total Return Index
36) S&P/TSX Venture/Group 14 Monthly Price Index
37) S&P/TSX Venture/Group 14 Monthly Total Return Index
38) CFMRC Monthly Equal Weighted Index
39) CFMRC Monthly Value Weighted Index
40) 30 Day Return on Long Term GOC Bonds

The format is:

```
(I4,1X,I6,1X,F8.5,1X,F9.6,3(1X,F8.5),15(1X,F8.2,1X,F8.2),3(1X,F9.6))
```

IV.4 Tickers & Names File - TICKERS.DAT

The ticker file contains a list of all securities included in the current version of the CFMRC database (DECyy.TSE). Only securities listed in this file are in DECyy.TSE and vice versa. There is one 98 character record for each security which, as with other files, is terminated with a linefeed for a 99 character record. Each record contains:

1) Ticker
2) Usage Number
3) Header Record Number
4) Name
5) Business

The format is:

```
(A8,1X,I1,1X,I9,1X,A45,1X,A31)
```

Appendix A - C Subroutines

The following C structure and variable definitions and subroutines demonstrate the format and decoding of the various data files in the CFMRC database. In addition, they form a core of useful routines around which it should be relatively easy to write programs for research.

This code is ANSI C and does not rely on any machine specifics. However, some machines have "extensions" that might cause the compiler to treat variable definitions differently (i.e. some RISC machines use int as long or double precision and long as quad-precision). As a rule, this will not affect these subroutines except in print statements. These compilers usually have a command line switch to switch them to ANSI mode if this is a concern.

```
#define MAX_DAYS        5000    /* maximum number of days in the daily calendar */
#define MAX_MONTHS      600     /* maximum number of months in the monthly calendar
*/
#define MAX_TICKS       3400    /* maximum number of different tickers in the dictionary
*/
#define MAX_SYM         3000    /* maximum number of securities in the ticker file */
#define RECL            112     /* record length in main data file including trailing LF */
#define RECLl           112l    /* same as RECL but a long int */


struct daily_calendar {         /* daily calendar record structure */
   long date;
   float price_idx;
   float total_idx;
   float price_group_idx[14];
   float total_group_idx[14];
   float goc1mo;
   float fx;
   float cfmrc_equal;
   float cfmrc_value;
   float cfmrc_u2equal;
   float cfmrc_u2value;
   float cfmrc_o2equal;
   float cfmrc_o2value;
   } d_c[MAX_DAYS];


struct monthly_calendar {       /* monthly calendar record structure */
   long date;
   float tb91d;
   float tb30d;
   float goc_lt;
   float corp_lt;
   float fx;
   float price_idx;
   float total_idx;
   float price_group_idx[14];
   float total_group_idx[14];
   float cfmrc_equal;
   float cfmrc_value;
   float goc1mo_return;
   } m_c[MAX_MONTHS];
```

```
struct tickers {                /* ticker name file data structure */
   char tic[9];
   char usage;
   long rec;
   } symbols[MAX_SYM];


struct daily {                  /* daily records - one per record */
     double oprice;
     double obid;
     double oask;
     double cprice;
     double cbid;
     double cask;
     double high;
     double low;
     long xacts;
     long volume;
     long quotes;
     long changes;
     double ret;
} dr;


struct monthly {                /* monthly records - 1 per record */
     double price;
     double eps;
     double beta;
     long xacts;
     long volume;
     long s_os;
     double ret;
     int divflag;
} mr;


struct header {                 /* header records - one per record */
     char *tic;
     char *use;
     int foreign;
     char *name;
     char *n_bus;
     char *cusip;
     int dstart;
     int ndays;
     int mstart;
     int nmons;
     int npradj;
} hr;


struct pradj {                  /* price adjustment records - 5 per record */
     double div;
     long exdate;
     int type;
} pr;


FILE *d,*m,*fopen();
char buffer[1000],secheader[RECL];
double atof();
int NTIC,NSEC,datafil;
```

```
read_daily_calendar()
/*
 *      Reads the CFMRC daily calendar and returns the number
 *      of days found.  Uses the global file pointer d and the
 *      global array d_c (daily calendar).
 *
 *      Although standard C practise is to use an origin of 0,
 *      this routine places data for relative day number 1 in
 *      array element d_c[1].  There is no relative day 0.
 */
{
   int N_DAYS,i;

   fprintf(stderr,"Reading daily calendar...");
   d = fopen("DAILY.IDX","r");
   for (N_DAYS=1; fgets(buffer,500,d) != NULL; N_DAYS++) {
      if (N_DAYS >= MAX_DAYS) {
         fprintf(stderr,"Too many days in the daily calendar!");
         fclose(d);
         return(MAX_DAYS);
      }
      d_c[N_DAYS].date = atoi(buffer+5);
      d_c[N_DAYS].price_idx = atof(buffer+12);
      d_c[N_DAYS].total_idx = atof(buffer+21);
      for (i=0; i<14; i++) {
         d_c[N_DAYS].price_group_idx[i] = atof(buffer+30+(i*18));
         d_c[N_DAYS].total_group_idx[i] = atof(buffer+39+(i*18));
      }
      d_c[N_DAYS].goc1mo = atof(buffer+282);
      d_c[N_DAYS].fx = atof(buffer+291);
      d_c[N_DAYS].cfmrc_equal = atof(buffer+300);
      d_c[N_DAYS].cfmrc_value = atof(buffer+310);
      d_c[N_DAYS].cfmrc_u2equal = atof(buffer+320);
      d_c[N_DAYS].cfmrc_u2value = atof(buffer+330);
      d_c[N_DAYS].cfmrc_o2equal = atof(buffer+340);
      d_c[N_DAYS].cfmrc_o2value = atof(buffer+350);
   }
   fprintf(stderr," %d days found.\n",N_DAYS);
   fclose(d);
   return(N_DAYS);
}
```

```
read_monthly_calendar()
/*
 *      Reads the CFMRC monthly calendar and returns the number
 *      of months found.  Uses the global file pointer m and the
 *      global array m_c (monthly calendar).
 *
 *      Data for relative month number 0 (4912) is placed in
 *      array element m_c[0].
 */
{
   int N_DAYS,i;

   fprintf(stderr,"Reading monthly calendar...");
   m = fopen("MONTHLY.IDX","r");
   m_c[0].date = 491200;
   for (N_MONTHS=1; fgets(buffer,500,m) != NULL; N_MONTHS++) {
      if (N_MONTHS >= MAX_MONTHS) {
         fprintf(stderr,"Too many months in the monthly calendar!");
         fclose(m);
         return(MAX_MONTHS);
      }
      m_c[N_MONTHS].date = atoi(buffer+4);
      m_c[N_MONTHS].tb91d = atof(buffer+11);
      m_c[N_MONTHS].tb30d = atof(buffer+20);
      m_c[N_MONTHS].goc_lt = atof(buffer+30);
      m_c[N_MONTHS].corp_lt = atof(buffer+39);
      m_c[N_MONTHS].fx = atof(buffer+48);
      m_c[N_MONTHS].price_idx = atof(buffer+57);
      m_c[N_MONTHS].total_idx = atof(buffer+66);
      for (i=0; i<14; i++) {
         m_c[N_MONTHS].price_group_idx[i] = atof(buffer+75+(i*18));
         m_c[N_MONTHS].total_group_idx[i] = atof(buffer+84+(i*18));
      }
      m_c[N_MONTHS].cfmrc_equal = atof(buffer+327);
      m_c[N_MONTHS].cfmrc_value = atof(buffer+337);
      m_c[N_MONTHS].goc1mo_return = atof(buffer+347);
   }
   fclose(m);
   fprintf(stderr," %d months found.\n",N_MONTHS);
   return(N_MONTHS);
}
```

```
read_ticker_index()
/*
 *      Reads the ticker names/index file into the global array
 *      symbols.  This can be used to determine the presence of
 *      a ticker in the database and find the record number of
 *      its header records in the main file.
 *
 *      Returns the number of tickers found and sets the global
 *      variable NSEC.
 */
{
   int sec;
   char *p;

   fprintf(stderr,"Reading ticker index...");
   d = fopen("TICKERS.DAT","r");
   for (sec=0; sec<MAX_SYM && fgets(buffer,500,d) != NULL; sec++) {
      for (*((p=strncpy(symbols[sec].tic,buffer,8))+8) = '\0'; *p != ' ' && *p != '\0'; p++) ;
      *p = '\0';
      symbols[sec].usage = buffer[9] - '0';
      symbols[sec].rec = atol(buffer+10);
   }
   fprintf(stderr," %d securities found.\n",sec);
   fclose(d);
   return(NSEC=sec);
}
```

```
void read_header(sec)
/*
 *      Reads the header records for security sec into the global
 *      structure hr.
 *
 *      Assumes that read_ticker_index() has already been called
 *      to set NSEC and that the data file DECyy.TSE is open under
 *      the file handle datafil (opened with open()).
 */
{
   long offs;

   if (sec >= NSEC) {
      fprintf(stderr,"Security is beyond the end of the index.\n");
      return;
   }
   offs = (symbols[sec].rec-1l)*RECLl;
   if (lseek(datafil,offs,0) == -1l) {
      fprintf(stderr,"Error seeking to requested header.\n");
      return;
   }
   if (read(datafil,secheader,RECL) != RECL) {
      fprintf(stderr,"Error reading requested header.\n");
      return;
   }
   (hr.tic=secheader)[8]='\0';
   (hr.use=secheader+9)[1]='\0';
   (hr.name=secheader+21)[45]='\0';
   (hr.n_bus=secheader+67)[24]='\0';
   hr.foreign=atoi(secheader+92);
   if (read(datafil,buffer,RECL) != RECL) {
      fprintf(stderr,"Error reading requested header.\n");
      return;
   }
   hr.dstart = atoi(buffer);
   hr.ndays = atoi(buffer+5);
   hr.mstart = atoi(buffer+11);
   hr.nmons = atoi(buffer+17);
   hr.npradj = atoi(buffer+23);
}
```

```
read_pradj(sec,n)
/*
 *      Reads price adjustment n for security sec into the global
 *      structure pr.
 *
 *      Assumes that read_header(sec) has already been called and
 *      that the data file DECyy.TSE is already open under the
 *      file handle datafil.
 *
 *      Returns -1 on error and 0 otherwise.
 */
{
   char dum[15];
   long offs;
   int p1;

   if (sec >= NSEC) {fprintf(stderr,"Security is beyond the end of the index.\n"); return(-1);}
   if (n > hr.npradj) {fprintf(stderr,"End of Price Adjustments.\n"); return(-1);}
   offs = (symbols[sec].rec+hr.ndays+hr.nmons+((n+4)/5))*RECLl;
   if (lseek(datafil,offs,0) == -1l) {
      fprintf(stderr,"Error seeking to requested price adjustment record.\n");
      return(-1);
      }
   if (read(datafil,buffer,RECL) != RECL) {
      fprintf(stderr,"Error reading requested price adjustment record.\n");
      return(-1);
      }
   if (n%5 == 0) p1 = 88;              /* where in the buffer does this pradj start */
   if (n%5 == 1) p1 = 0;
   if (n%5 == 2) p1 = 22;
   if (n%5 == 3) p1 = 44;
   if (n%5 == 4) p1 = 66;              /* offset in buffer of record n */

   (strncpy(dum,buffer+p1,11))[11]='\0';
   pr.div=atof(dum);                   /* may not be any white space between numbers */
   (strncpy(dum,buffer+p1+11,4))[4]='\0';
   pr.type=atoi(dum);
   (strncpy(dum,buffer+p1+15,7))[7]='\0';
   pr.exdate=atol(dum);
   return(0);
}
```

```
read_monthly(sec,n)
/*
 *      Reads monthly record n for security sec into the global
 *      structure mr.
 *
 *      Assumes that read_header(sec) has already been called and
 *      that the data file DECyy.TSE is already open under the
 *      file handle datafil.
 *
 *      Returns -1 on error and 0 otherwise.
 */
{
   char dum[15];
   long offs;

   if (sec >= NSEC) {fprintf(stderr,"Security is beyond the end of the index.\n"); return(-1);}
   if (n > hr.nmons) {fprintf(stderr,"End of Monthly Data.\n"); return(-1);}
   offs = (symbols[sec].rec+hr.ndays+n)*RECLl;
   if (lseek(datafil,offs,0) == -1l) {
      fprintf(stderr,"Error seeking to requested monthly record.\n");
      return(-1);
   }
   if (read(datafil,buffer,RECL) != RECL) {
      fprintf(stderr,"Error reading requested monthly record.\n");
      return(-1);
   }

   (strncpy(dum,buffer,9))[9]='\0';
   mr.price=atof(dum);                  /* may not be any white space between numbers */
   (strncpy(dum,buffer+9,9))[9]='\0';
   mr.xacts=atol(dum);
   (strncpy(dum,buffer+18,10))[10]='\0';
   mr.volume=atol(dum);
   (strncpy(dum,buffer+28,11))[11]='\0';
   mr.s_os=atol(dum);
   (strncpy(dum,buffer+39,13))[13]='\0';
   mr.eps=atof(dum);
   (strncpy(dum,buffer+52,7))[7]='\0';
   mr.beta=atof(dum);
   (strncpy(dum,buffer+59,11))[11]='\0';
   mr.ret=atof(dum);
   mr.divflag=atoi(buffer+70);
   return(0);
}
```

```
read_daily(sec,n)
/*
 *      Reads daily record n for security sec into the global
 *      structure dr.
 *
 *      Assumes that read_header(sec) has already been called and
 *      that the data file DECyy.TSE is already open under the
 *      file handle datafil.
 *
 *      Returns -1 on error and 0 otherwise.
 */
{
   char dum[15];
   long offs;

   if (sec >= NSEC) {fprintf(stderr,"Security is beyond the end of the index.\n"); return(-1);}
   if (n > hr.ndays) {fprintf(stderr,"End of Daily Data.\n"); return(-1);}
   offs = (symbols[sec].rec+n)*RECLl;
   if (lseek(datafil,offs,0) == -1l) {
      fprintf(stderr,"Error seeking to requested daily record.\n");
      return(-1);
   }
   if (read(datafil,buffer,RECL) != RECL) {
      fprintf(stderr,"Error reading requested daily record.\n");
      return(-1);
   }
   (strncpy(dum,buffer,9))[9]='\0';
   dr.oprice=atof(dum);                 /* may not be any white space between numbers */
   (strncpy(dum,buffer9,9))[9]='\0';
   dr.obid=atof(dum);
   (strncpy(dum,buffer+18,9))[9]='\0';
   dr.oask=atof(dum);
   (strncpy(dum,buffer+27,9))[9]='\0';
   dr.cprice=atof(dum);
   (strncpy(dum,buffer+36,9))[9]='\0';
   dr.cbid=atof(dum);
   (strncpy(dum,buffer+45,9))[9]='\0';
   dr.cask=atof(dum);
   (strncpy(dum,buffer+54,9))[9]='\0';
   dr.high=atof(dum);
   (strncpy(dum,buffer+63,9))[9]='\0';
   dr.low=atof(dum);
   (strncpy(dum,buffer+72,8))[8]='\0';
   dr.xacts=atol(dum);
   (strncpy(dum,buffer+80,9))[9]='\0';
   dr.volume=atol(dum);
   (strncpy(dum,buffer+89,6))[6]='\0';
   dr.quotes=atol(dum);
   (strncpy(dum,buffer+95,6))[6]='\0';
   dr.changes=atol(dum);
   (strncpy(dum,buffer+101,10))[10]='\0';
   dr.ret=atof(dum);
   return(0);
}
```