

# Datafeed Toolbox

For Use with MATLAB®

Computation

Visualization

Programming

User's Guide

*Version 1*



## How to Contact The MathWorks:



www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup



support@mathworks.com	Technical support
suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information



508-647-7000	Phone
--------------	-------



508-647-7001	Fax
--------------	-----



The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098	Mail
--	------

For contact information about worldwide offices, see the MathWorks Web site.

### *Datafeed Toolbox User's Guide*

© COPYRIGHT 1999 - 2000 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

**FEDERAL ACQUISITION:** This provision applies to all acquisitions of the Program and Documentation by or for the federal government of the United States. By accepting delivery of the Program, the government hereby agrees that this software qualifies as "commercial" computer software within the meaning of FAR Part 12.212, DFARS Part 227.7202-1, DFARS Part 227.7202-3, DFARS Part 252.227-7013, and DFARS Part 252.227-7014. The terms and conditions of The MathWorks, Inc. Software License Agreement shall pertain to the government's use and disclosure of the Program and Documentation, and shall supersede any conflicting contractual terms or conditions. If this license fails to meet the government's minimum needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to MathWorks.

MATLAB, Simulink, Stateflow, Handle Graphics, and Real-Time Workshop are registered trademarks, and Target Language Compiler is a trademark of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Printing History:	December 1999 First printing	New for MATLAB 5.3 (Release 11)
	June 2000 Online only	Revised for Version 1.2
	December 2000 Online only	Revised for Version 1.3

## Preface

---

<b>Using this Guide</b> .....	<b>iv</b>
Organization of the Document .....	<b>iv</b>
<b>Typographical Conventions</b> .....	<b>v</b>
<b>Related Products</b> .....	<b>vi</b>
Additional Software .....	<b>vii</b>
<b>Installation and Configuration</b> .....	<b>viii</b>

## Getting Started

---

**1**

<b>Introduction</b> .....	<b>1-2</b>
<b>Communicating with a Financial Data Server</b> .....	<b>1-3</b>
Communication Management .....	<b>1-3</b>
Verifying the Connection .....	<b>1-4</b>
Retrieving Connection Properties .....	<b>1-4</b>
Disconnecting from a Data Server .....	<b>1-5</b>
<b>Retrieving Data</b> .....	<b>1-6</b>
Example: Retrieving Bloomberg Data .....	<b>1-6</b>
<b>Datafeed Toolbox Graphical User Interface</b> .....	<b>1-13</b>
Datafeed Dialog Box .....	<b>1-13</b>
Securities Lookup Dialog Box (Bloomberg, FT Interactive Data) .....	<b>1-17</b>

## Function Reference

---

**2**

<b>Functions - By Category</b> .....	<b>2-2</b>
Bloomberg Functions .....	<b>2-2</b>
Hyperfeed Functions .....	<b>2-10</b>
FT Interactive Data Functions .....	<b>2-18</b>
Yahoo Functions .....	<b>2-25</b>

## Directory Structure

---

**A**

# Preface

---

<b>Using this Guide</b> . . . . .	iv
Organization of the Document . . . . .	iv
<b>Typographical Conventions</b> . . . . .	v
<b>Related Products</b> . . . . .	vi
Additional Software . . . . .	vii
<b>Installation and Configuration</b> . . . . .	viii

## Using this Guide

This book describes the Datafeed Toolbox for MATLAB®. The Datafeed Toolbox enables MATLAB users to obtain financial data from a financial data server and to enter this data into the MATLAB environment.

## Organization of the Document

<b>Chapter</b>	<b>Description</b>
Chapter 1. “Getting Started”	Describes how to connect to a data server and request various forms of data.
Chapter 2. “Function Reference”	Describes the specific functions used to connect to a data server and to request various forms of data.
Appendix A. “Directory Structure”	Shows the location of directories and files created after installation of the Datafeed Toolbox on your system.

# Typographical Conventions

This manual uses some or all of these conventions.

Item	Convention	Example
Example code	Monospace font	To assign the value 5 to A, enter <code>A = 5</code>
Function names, syntax, filenames, directory/folder names, and user input	Monospace font	The <code>cos</code> function finds the cosine of each array element. Syntax line example is <code>MLGetVar ML_var_name</code>
Buttons and keys	<b>Boldface</b> with book title caps	Press the <b>Enter</b> key.
Literal strings (in syntax descriptions in reference chapters)	<b>Monospace bold</b> for literals	<code>f = freqspace(n, 'whole')</code>
Mathematical expressions	<i>Italics</i> for variables Standard text font for functions, operators, and constants	This vector represents the polynomial $p = x^2 + 2x + 3$ .
MATLAB output	Monospace font	MATLAB responds with <code>A =</code> <code>5</code>
Menu and dialog box titles	<b>Boldface</b> with book title caps	Choose the <b>File Options</b> menu.
New terms and for emphasis	<i>Italics</i>	An <i>array</i> is an ordered collection of information.
Omitted input arguments	(...) ellipsis denotes all of the input/output arguments from preceding syntaxes.	<code>[c, ia, ib] = union(...)</code>
String variables (from a finite list)	<i>Monospace italics</i>	<code>sysc = d2c(sysd, 'method')</code>

## Related Products

The MathWorks provides several products relevant to the tasks you can perform with the Datafeed Toolbox.

For more information about any of these products, see either:

- The online documentation for that product, if it is installed or if you are reading the documentation from the CD
- The MathWorks Web site, at <http://www.mathworks.com>; see the “products” section

---

**Note** The toolboxes listed below all include functions that extend the capabilities of MATLAB.

---

<b>Product</b>	<b>Description</b>
Database Toolbox	Exchange data with relational databases
Excel Link	Use MATLAB with Microsoft Excel
Financial Derivatives Toolbox	Model and analyze fixed-income derivatives and securities
Financial Time Series Toolbox	Analyze and manage financial time series data
Financial Toolbox	Model financial data and develop financial analysis algorithms
GARCH Toolbox	Analyze financial volatility using univariate GARCH models
Optimization Toolbox	Solve standard and large-scale optimization problems
Statistics Toolbox	Apply statistical algorithms and probability models



## **Additional Software**

If you want to use the Datafeed Toolbox to retrieve data from Bloomberg, Hyperfeed, or FT Interactive Data Corporation data servers, you need to install client software available from each of these companies. Contact your data server sales representative for information.

Information about the services offered by these companies is available on the Web at:

<http://www.bloomberg.com>

[http://www.hyperfeed.com/f\\_main.html](http://www.hyperfeed.com/f_main.html)

<http://www.FTInteractiveData.com>

## **Installation and Configuration**

To install the Datafeed Toolbox, see the *MATLAB Installation Guide* for your computer system.

For information about installing Bloomberg, Hyperfeed, or FT Interactive Data Corporation software on your system, contact your sales representative from these companies.

# Getting Started

---

<b>Introduction</b> . . . . .	1-2
<b>Communicating with a Financial Data Server</b> . . . . .	1-3
Communication Management . . . . .	1-3
Verifying the Connection . . . . .	1-4
Retrieving Connection Properties . . . . .	1-4
Disconnecting from a Data Server . . . . .	1-5
<b>Retrieving Data</b> . . . . .	1-7
Example: Retrieving Bloomberg Data . . . . .	1-7
<b>Datafeed Toolbox Graphical User Interface</b> . . . . .	1-14
Datafeed Dialog Box . . . . .	1-14
Securities Lookup Dialog Box . . . . .	1-18

## **Introduction**

This document describes the Datafeed Toolbox for MATLAB®. The Datafeed Toolbox effectively turns your MATLAB workstation into a financial data acquisition terminal. Using the Datafeed Toolbox, you can download a wide variety of security data from financial data servers into your MATLAB workspace. Then, you can pass this data to MATLAB or to another toolbox, such as the Financial Time Series Toolbox, for further analysis.

## Communicating with a Financial Data Server

The Datafeed Toolbox supports connections to four financial data servers:

- Bloomberg (<http://www.bloomberg.com>)
- Hyperfeed ([http://www.hyperfeed.com/f\\_main.html](http://www.hyperfeed.com/f_main.html))
- FT Interactive Data Corporation (<http://www.FTInteractiveData.com>)
- Yahoo (<http://www.yahoo.com>)

Bloomberg, Hyperfeed, and FT Interactive Data all require that you install proprietary software on your PC. To connect to Yahoo, you need to have access to the Internet and to install a Web browser. The most commonly used browsers, Microsoft Internet Explorer (<http://www.microsoft.com>) and Netscape Communicator (<http://www.netscape.com>) are available for free download.

### Communication Management

For each of the supported financial data servers, the Datafeed Toolbox uses four commands to manage communication:

- `bloomberg`, `hyperfeed`, `idc`, or `yahoo`: establishes a connection to the appropriate data server.
- `isconnection`: verifies that a connection is working.
- `get`: retrieves connection properties.
- `close`: terminates the connection.

An additional function, `fetch`, obtains the desired data from the data server and transfers it to your PC.

### Example: The bloomberg Function

Connect to the Bloomberg data server using the `bloomberg` function. The connection requires a port number and an IP address.

The syntax for the `bloomberg` function is

```
connect = bloomberg(PortNumber, 'IPAddress')
```

The IP address is entered as a MATLAB string. For example, the command

```
c = bloomberg(8194, '123.456.54.123')
```

returns a Bloomberg connection object:

```
c =  
  
    connection: 84554360  
    ipaddress: '123.456.54.123'  
    port: 8194
```

The `connection` field within the object `c` contains the Bloomberg connection handle that will be used in processing future data requests.

If you want to accept the default port number and IP address provided when your Bloomberg software was installed, enter

```
c = bloomberg
```

with no arguments.

## Verifying the Connection

To verify that a data server connection is valid and open, use the `isconnection` command. For a connection object `c` previously created with one of the above connection commands,

```
x = isconnection(c)
```

returns `x = 1` if the connection is valid and open or `x = 0` if the connection is closed or invalid.

## Retrieving Connection Properties

To retrieve the properties of a connection object, use the command `get`. This command returns different values depending upon which data server is being used.

### Example: Retrieving Bloomberg Connection Properties

For the Bloomberg connection

```
c = bloomberg(8194, '123.456.54.123')
```

the command

```
p = get(c)
```

returns the list of all valid connection properties and their values associated with the connection object `c`:

```
p =  
  connection: 84554360  
  ipaddress: '123.456.54.123'  
  port: 8194  
  socket: 248  
  version: 1.8000
```

The `get` command can return specific properties of a connection object. For example, to obtain the port number and Bloomberg version for the connection object `c`, use the command

```
p = get(c,{'Port';'Version'})
```

which returns

```
p =  
  port: 8194  
  version: 1.8000
```

When returning a single property, for example, the connection handle, the command

```
p = get(c, 'Connection')
```

returns

```
p =  
  84554360
```

For a single returned property the output is not a structure.

## Disconnecting from a Data Server

To close a data server connection and disconnect, use the `close` command with the format

```
close(Connect)
```

You must have previously created the connection object with one of the connection commands.

## Retrieving Data

The `fetch` command controls data retrieval from a data server connection. `fetch` returns different information depending upon which data server is being accessed. See the version of `fetch` appropriate for your data server for further information.

### Example: Retrieving Bloomberg Data

This section illustrates the use of the `fetch` command to retrieve data from a Bloomberg data server. Versions of the `fetch` command that retrieve data from other data servers work similarly.

#### Retrieving Header (Bloomberg Default) Data

A header (default) data request to Bloomberg returns a fixed set of field data. Not all fields in the header data are relevant for a specific security.

**Determining Header Fields.** The list of valid header fields is stored in the file `@bloomberg/bbfields.mat`. Use the command

```
load @bloomberg/bbfields
```

to load this file. The variable `headerfieldnames` contains the list of header field names.

**Obtaining Data.** To retrieve header data from the Bloomberg connection, use `fetch` with the syntax

```
data = fetch(Connect, Security, 'HEADER', Flag)
```

where

- `Connect` is a Bloomberg connection object established with the `bloomberg` command.
- `Security` is the list of securities for which data is requested.
- The `'HEADER'` argument is entered literally.
- `Flag` denotes the dates for which data can be retrieved. `Flag` has three possible values:
  - `DEFAULT` fills all fields with data from the most recent date with a bid, ask, or trade.



- TODAY fills the fields with data from today only.
- ENHANCED fills the fields with data for the most recent event for each individual field. In this case, for example, the bid and ask group fields could come from different dates.

Commands of the form

```
data = fetch(Connection, Security)
data = fetch(Connection, Security, 'HEADER')
data = fetch(Connection, Security, 'HEADER', 'DEFAULT')
```

are equivalent.

The returned data has a fixed set of fields. For example, a header inquiry for the security IBM US Equity returns data of the form:

```

                Status:0
                OpenPrice:93
    TodaysOpenPrice:93
                HighPrice:93.1875
    TodaysHighPrice:93.1875
                LowPrice:89
    TodaysLowPrice:89
                LastPrice:90.9375
    TodaysLastPrice:0
                SettlePrice:NaN
                BidPrice:0
    TodaysBidPrice:NaN
                AskPrice:0
    TodaysAskPrice:NaN
                YieldBid:NaN
    TodaysYieldBid:NaN
                YieldAsk:NaN
    TodaysYieldAsk:NaN
                LimitUp:NaN
                LimitDown:NaN
                OpenInterest:3359000
    LastPriceYesterday:95
                Scale:1
                LastPriceTime:0.4993
    LastTradeExchange:7
                TickDirection:-1
```

```
        BidSize:0
        TodaysBidSize:NaN
        AskSize:NaN
        TodaysAskSize:0
        BidCondition:NaN
        AskCondition:NaN
        LastTradeCondition:NaN
        LastMarketCondition:NaN
        Monitorable:1
        TotalVolume:60018500
        TodaysTotalVolume:0
        TotalNumberOfTicks:63318
        TodaysTotalNumberOfTicks:63318
        SessionStartTime:0.3958
        SessionEndTime:0.6875
        Currency:538989397
        Format:0
        SecurityKey: {'IBM US Equity'}
        AsOfDate:730441
        TodaysAsOfDate:730441
```

Not all fields are applicable to IBM US Equity, the security about which we inquired.

## Retrieving Field Data

The `fetch` command with the `GETDATA` argument obtains Bloomberg field data. The entire set of field data provides statistics for all possible securities but does not apply universally to any one security.

**Determining Field Names.** The complete list of valid field names is stored in the file `@bloomberg/bbfields.mat`. Use the command

```
load @bloomberg/bbfields
```

to load this file. The variable `bbfieldnames` contains the list of field names. This list includes the header field names plus numerous others.

**Obtaining Data.** To obtain data for specific fields of a given security, use the `fetch` command with the syntax

```
d = fetch(Connect, Security, 'GETDATA', Fields)
```

For example, use the `bloomberg` command to establish a connection `c1` to a Bloomberg data server.

```
c1 = bloomberg(8234, '123.457.78.999')
```

Then

```
d = fetch(c1, 'IBM US Equity', 'GETDATA',
         {'OpenPrice'; 'LastPrice'})
```

returns

```
d =
    OpenPrice: 126.2500
    LastPrice: 125.1250
```

### Retrieving Time Series Data

The `fetch` command with the `TIMESERIES` argument returns price and volume data for a particular security on a specified date. Time series data for a given security and a specific date are returned using the syntax

```
data = fetch(Connection, Security, 'TIMESERIES', Date)
```

Date may be a MATLAB date string or serial date number.

To obtain time series data for the current day, you can use the alternate forms of the command

```
data = fetch(Connection, Security, 'TIMESERIES', now)
```

or

```
data = fetch(Connection, Security, 'TICKS').
```

To obtain time series data for IBM using an existing connection `c1`, enter the command

```
data = fetch(c1, 'IBM US Equity', 'TIMESERIES', '11/16/99')
```

The result will look like this:

```
data =
    31.00    730440.31    130.00    1000.00
    32.00    730440.31    130.00     200.00
    32.00    730440.35    129.50   10000.00
```

31.00	730440.35	129.50	100.00
32.00	730440.35	129.50	100.00
1.00	730440.56	129.25	4000.00
31.00	730440.56	129.38	1500.00
32.00	730440.56	129.50	500.00
1.00	730440.56	129.63	5000.00
31.00	730440.56	129.63	400.00
32.00	730440.56	129.63	200.00
1.00	730440.56	129.69	5000.00
31.00	730440.56	129.69	500.00
32.00	730440.56	129.69	500.00
31.00	730440.56	129.75	100.00
32.00	730440.56	130.00	100.00
1.00	730440.56	130.00	5000.00
1.00	730440.56	129.88	5000.00
31.00	730440.56	129.88	300.00

Column 1 contains the tick type flag, column 2 contains the time stamp in MATLAB serial date number format, column 3 contains the tick value, and column 4 contains the number of shares in the transaction.

### Retrieving Historical Data

Use the `fetch` command with the `HISTORY` argument to obtain historical data for a specific security.

For a specified field of a particular security use the syntax

```
d = fetch(Connect,Security,'HISTORY',Field,FromDate,ToDate)
```

to obtain historical data. Data for the field is returned for the date range from `FromDate` to `ToDate`. See “Determining Field Names” on page 1-8 for instructions on determining valid field names.

For example, to obtain the closing price for IBM for the dates July 15, 1999 to August 2, 1999 using the connection `c1`, enter

```
data = fetch(connection, 'IBM US Equity', 'HISTORY',...
'LastPrice', '07/15/99', '08/02/99')
```

```
data =
```

```
730316.00    136.31
```

730317.00	136.25
730320.00	134.63
730321.00	128.25
730322.00	129.00
730323.00	123.88
730324.00	124.81
730327.00	123.00
730328.00	126.25
730329.00	128.38
730330.00	125.38
730331.00	125.69
730334.00	122.25

Column 1 is the date represented as a MATLAB date number, and column 2 is the last price.

### Finding Ticker Symbols

You can use the `fetch` command with the `LOOKUP` argument to find a ticker symbol when you are uncertain what the symbol might be. Use the syntax

```
data = fetch(Connect, SearchString, 'LOOKUP', Market)
```

to locate a specific ticker symbol.

The `SearchString` argument is the comparison string used in the lookup operation, and `Market` indicates the type of security (the market in which the security trades). The allowable values for `Market` are

- `Comdty` (Commodities)
- `Corp` (Corporate Bonds)
- `Curncy` (Currencies)
- `Equity` (Equities)
- `Govt` (Government Bonds)
- `Index` (Indexes)
- `M-Mkt` (Money Market Securities)
- `Mtge` (Mortgage-backed Securities)
- `Muni` (Municipal Bonds)
- `Pfd` (Preferred Stocks)

For example, using `fetch` with the connection `c1` to look up the ticker symbol for New Zealand government bonds returns

```
data = fetch(c1, 'New', 'LOOKUP', 'Govt')
```

returns a list of possible values:

```
data =
```

```
'NZTB   New Zealand Treasury Bill NZGB   New Zealand Governme'  
'NZGB   New Zealand Government Bond NZ   New Zealand Govern '  
'NZ     New Zealand Government International Bond HCNZ   Hous '  
'ECNZ   Electric Corporation of New Zealand Bond NZTB NZGB NZ H'
```

## Datafeed Toolbox Graphical User Interface

The Datafeed Toolbox provides a graphical user interface (GUI) consisting of two dialog boxes. The **Datafeed** dialog box consists of two tabbed dialogs, one to establish a data server connection, and the second to retrieve data from the server. The second dialog box, the **Securities Lookup** dialog box, enables you to find the ticker symbol for a specific security when you know at least part of the name of the security.

For additional information about the **Datafeed** dialog box, see

- “Connecting to a Data Server” on page 1-13
- “Data Retrieval” on page 1-15

For additional information about the **Securities Lookup** dialog box, see

- “Securities Lookup Dialog Box (Bloomberg, FT Interactive Data)” on page 1-17

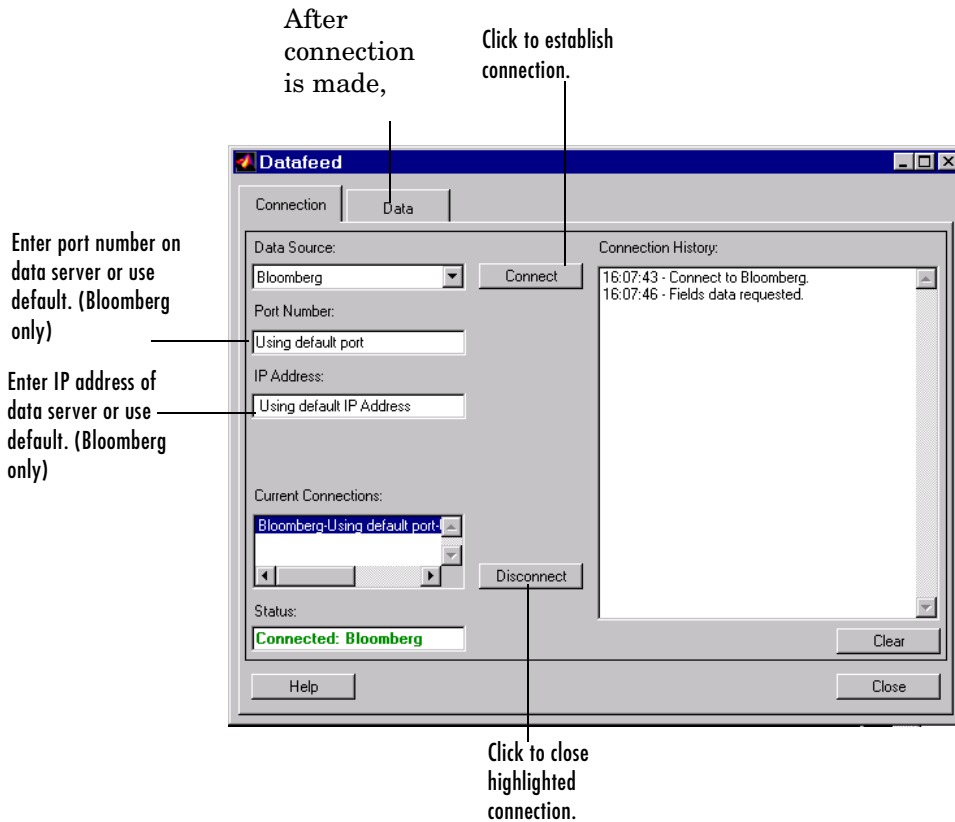
### Datafeed Dialog Box

The **Datafeed** dialog box establishes the connection with the data server and manages the retrieval of data. Enter the command `dftool` to display the **Datafeed** dialog box on your screen. The **Datafeed** dialog box consists of two tabbed dialogs:

- The **Connection** tab establishes communication with a data server. (See “Connecting to a Data Server” on page 1-13.)
- The **Data** tab specifies the data request. (See “Data Retrieval” on page 1-15.)

#### Connecting to a Data Server

The **Connection** tab establishes a connection to one or more data servers. For Yahoo and FT Interactive Data connections, choose the data server from the **Data Source** choices and click on the **Connect** button. For a Bloomberg connection, you can specify a specific IP address and port number on the Bloomberg server, or alternatively, just click on the **Connect** button and accept the default values provided when the Bloomberg software was installed on your machine.



- 1 (Bloomberg only) Enter the port number on the data server in the **Port Number** box (or use default).
- 2 (Bloomberg only) Enter the IP address of the data server in the **IP Address** box (or use default).
- 3 Click the **Connect** button to establish the connection.
- 4 When the Connected message appears in the **Status** box, click on the **Data** tab to begin the process of retrieving data from the data server. (For information on the **Data** tab, see “Data Retrieval” on page 1-15).



- 5 Click the **Disconnect** button to terminate the session highlighted in the **Current Connections** box.

### **Data Retrieval**

The **Data** tab manages the retrieval of data from the data server.

Enter security symbol if known. Click **Get Data** button to retrieve data. Click **Add** button to add security to **Selected Securities** list.

(Bloomberg only) Use to find security symbol if not known. Displays **Securities Lookup** dialog box.

Type of data to be retrieved from data server.

Security fields.

Click to retrieve data.

Fields with data retrieved from the connection.

Variable in MATLAB workspace.

Field	Value
LowPrice	110.06
Monitorable	1
OpenInterest	337000
OpenPrice	110.50
Scale	1
SecurityKey	IBM Equity
SessionEndTime	16:30:00
SessionStartTime	09:30:00
SettlePrice	NaN
Status	0

1 Enter security symbol in the **Enter Security** box.

- 2 Indicate the type of data you are seeking in the **Data Selection** panel.
- 3 Indicate whether you want the default or full set of data in the **Fields** panel.
- 4 Click the **Get Data** button to retrieve data from the data server.

---

**Note** If you do not know the symbol for a security, you can use the **Lookup** button to find the name of the security. (See “Securities Lookup Dialog Box (Bloomberg, FT Interactive Data)” on page 1-17.)

---

## **Securities Lookup Dialog Box (Bloomberg, FT Interactive Data)**

Click on the **Lookup** button of the **Datafeed** dialog box **Data** tab to display the **Securities Lookup** dialog box. See “Data Retrieval” on page 1-15 for information about the **Data** tab.

The **Securities Lookup** dialog box provides a means to obtain the ticker symbol for a particular security when you know part of the name. You can then enter the ticker symbol into the **Enter Security** field on the **Data** tab. It is essential that you enter the ticker symbol as specified; otherwise, the data server may provide no data or provide data for some other security.

Alternatively, you can highlight one or more securities in the list and click **Select**. The selected securities are added to the **Selected Securities** list on the **Data** tab.

The screenshot shows a dialog box titled "Datafeed Securities Lookup". It has a "Lookup:" text box containing "FORD" and a "Choose Market:" dropdown menu set to "Equity". A "Submit" button is located below the market selection. To the right is a table with two columns: "Security" and "Symbol". The table lists several entries for "FORD MOTOR CO" with different ticker symbols and market codes. The entry "FORD MOTOR CO (F US)" is highlighted in blue. A "Select" button is positioned below the table. At the bottom of the dialog are "Help" and "Close" buttons.

Enter lookup search string.

Indicate choice of market from Market list.

Click to send request to data server.

Security	Symbol
FORD MOTOR CO	(7657 JP)
FORD MOTOR CO	(FRD LN)
FORD MOTOR CO	(FU NA)
FORD MOTOR CO	(F SW)
FORD MOTOR CO	(F US)
FORD MOTOR CO-CT	(FMC GR)
FORD MOTOR CO-NY	(10107 NY)

Search results. Displays all possible values of company name and ticker symbol. Select desired securities from list.

Enter selected securities on Data tab.

# Function Reference

---

<b>Bloomberg Function Summary . . . . .</b>	<b>2-3</b>
<b>Hyperfeed Function Summary . . . . .</b>	<b>2-11</b>
<b>FT Interactive Data Function Summary . . . . .</b>	<b>2-19</b>
<b>Yahoo Function Summary . . . . .</b>	<b>2-26</b>

---

## Functions - By Category

Datafeed Toolbox functions are categorized by the data service that provides financial data to MATLAB.

- “Bloomberg Functions”
- “Hyperfeed Functions”
- “FT Interactive Data Functions”
- “Yahoo Functions”

### **Bloomberg Functions**

This section provides detailed descriptions of the Bloomberg functions in the Datafeed Toolbox.

<code>bloomberg</code>	Connect to Bloomberg
<code>close</code>	Close connection
<code>fetch</code>	Request data
<code>get</code>	Get connection properties
<code>isconnection</code>	True if valid connection

<b>Purpose</b>	Connect to Bloomberg				
<b>Syntax</b>	<code>Connect = bloomberg(PortNumber, 'IPAddress')</code> <code>Connect = bloomberg</code>				
<b>Arguments</b>	<table><tr><td>PortNumber</td><td>Port on machine where connection is being made.</td></tr><tr><td>IPAddress</td><td>A MATLAB string containing the internet address of machine where connection is being made.</td></tr></table>	PortNumber	Port on machine where connection is being made.	IPAddress	A MATLAB string containing the internet address of machine where connection is being made.
PortNumber	Port on machine where connection is being made.				
IPAddress	A MATLAB string containing the internet address of machine where connection is being made.				
<b>Description</b>	<p><code>Connect = bloomberg(PortNumber, IPAddress)</code> establishes a connection to a Bloomberg data server using the port number, PortNumber, and the internet address, IPAddress.</p> <p><code>Connect = bloomberg</code> establishes a connection to a Bloomberg data server using port number 8194 and the default internet address provided when the Bloomberg software was installed on your machine.</p>				
<b>Examples</b>	<pre>c = bloomberg(8194, '111.222.33.444')</pre> <p>makes a connection to the Bloomberg server on port 8194 of the machine with internet address 111.222.33.444.</p>				
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>get</code> , <code>isconnection</code>				

# close

---

<b>Purpose</b>	Close Bloomberg connection		
<b>Syntax</b>	<code>close(Connect)</code>		
<b>Arguments</b>	<table><tr><td><code>Connect</code></td><td>Bloomberg connection object created with the <code>bloomberg</code> command.</td></tr></table>	<code>Connect</code>	Bloomberg connection object created with the <code>bloomberg</code> command.
<code>Connect</code>	Bloomberg connection object created with the <code>bloomberg</code> command.		
<b>Description</b>	<code>close(Connect)</code> closes the connection to the Bloomberg data server.		
<b>Examples</b>	<pre>c = bloomberg(8194, '111.222.33.444')</pre> <p>establishes a Bloomberg connection, <code>c</code>.</p> <pre>close(c)</pre> <p>closes this connection.</p>		
<b>See Also</b>	<code>bloomberg</code>		



**Purpose** Request data from Bloomberg

**Syntax**

```

data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'HEADER', 'Flag')
data = fetch(Connect, 'Security', 'GETDATA', 'Fields')
data = fetch(Connect, 'Security', 'TICKS')
data = fetch(Connect, 'Security', 'TIMESERIES', 'Date')
data = fetch(Connect, 'Security', 'HISTORY', 'Field', 'FromDate',
            'ToDate')
ticker = fetch(Connect, 'SearchString', 'LOOKUP', 'Market')

```

**Arguments**

Connect	Bloomberg connection object created with the bloomberg command.
Security	A MATLAB string containing the name of a security in a format recognizable by the Bloomberg server. (Note: For header data only, Security may be a cell array of strings containing a list of securities.)
<i>Flag</i>	<p>A MATLAB string indicating the dates from which data is to be retrieved. Possible values are:</p> <p>DEFAULT: Data from most recent bid, ask, or trade. If a Flag value is not specified, 'DEFAULT' is assumed.</p> <p>TODAY: Today's data only.</p> <p>ENHANCED: Data from most recent date of each individual field.</p>
<i>Fields</i>	A MATLAB string or cell array of strings indicating specific fields for which data is to be provided. Valid field names are in the file @bloomberg/bbfields.mat. The variable bbfieldnames contains the list of field names.
Date	Date string or serial date number indicating date for the time series. Specify now for today's time series data.
FromDate	Beginning date for historical data.

# fetch

---

ToDate	End date for historical data.
Market	A MATLAB string indicating the market in which a particular security trades. <i>Market</i> values are:
	Comdty (Commodities)
	Corp (Corporate bonds)
	Curncy (Currencies)
	Equity (Equities)
	Govt (Government bonds)
	Index (Indexes)
	M-Mkt (Money Market securities)
	Mtge (Mortgage-backed securities)
	Muni (Municipal bonds)
	Pfd (Preferred stocks)

## Description

For a given security, `fetch` returns header (default), field, time series, and historical data via the Bloomberg connection

`data = fetch(Connect, 'Security')` fills the header fields with data from the most recent date with a bid, ask, or trade.

`data = fetch(Connect, 'Security', 'HEADER', 'Flag')` returns data based upon the value of *Flag*.

- If *Flag* is `DEFAULT`, `fetch` fills the header fields with data from the most recent date with a bid, ask, or trade. (This is the equivalent of `data = fetch(Connect, Security)`).
- If *Flag* is `TODAY`, `fetch` returns the header field data with data from today only.
- If *Flag* is `ENHANCED`, `fetch` returns the header field data for the most recent date of each individual field. In this case, for example, the bid and ask group fields could come from different dates.

`data = fetch(Connect, 'Security', 'GETDATA', 'Fields')` returns the data for the specified fields of the indicated security.

`data = fetch(Connect, 'Security', 'TICKS')` returns today's tick data.

`data = fetch(Connect, 'Security', 'TIMESERIES', 'Date')` returns the tick data for a security for the specified date. For today's tick data, specify

```
data = fetch(Connect, 'Security', 'TIMESERIES', now)
data = fetch(Connect, 'Security', 'TICKS') is an equivalent command.
```

`data = fetch(Connect, 'Security', 'HISTORY', 'Field', 'FromDate', 'ToDate')` returns historical data for the specified field for the date range FromDate to ToDate.

`ticker = fetch(Connect, 'SearchString', 'LOOKUP', 'Market')` uses SearchString to find the ticker symbol for a security trading in a designated market. The output ticker is a column vector of possible ticker values.

## Examples

For a United States equity with ticker ABC:

```
D = fetch(C, 'ABC US Equity')
```

returns the header data.

```
D = fetch(C, 'ABC US Equity', 'GETDATA', {'LastPrice'; 'OpenPrice'})
```

returns the opening and closing prices.

```
D = fetch(C, 'ABC US Equity', 'TICKS') and
D = fetch(C, 'ABC US Equity', 'TIMESERIES', now)
```

return today's time series.

```
D = fetch(C, 'ABC US Equity', 'HISTORY', 'LastPrice', '8/01/99', ...
'8/10/99')
```

returns the closing price for the given dates.

## See Also

bloomberg, close, get, isconnection

# get

---

**Purpose** Get Bloomberg connection properties

**Syntax** `value = get(Connect, 'PropertyName')`

**Arguments** `Connect` Bloomberg connection object created with the `bloomberg` command.

`PropertyName` (optional) A MATLAB string or cell array of strings containing property names. Property names are:

`Connection`  
`IPAddress`  
`Port`  
`Socket`  
`Version`

**Description** `value = get(Connect, 'PropertyName')` returns a MATLAB structure containing the value of the specified properties for the Bloomberg connection object.

`value = get(Connect)` returns the value for all properties.

**Examples** `c = bloomberg(8194, '111.222.33.444')`

establishes a Bloomberg connection, `c`.

The command

```
p = get(c, {'Port', 'IPAddress'})
```

returns

```
p =  
    port: 8194  
  ipaddress: 111.222.33.444
```

**See Also** `bloomberg`, `close`, `fetch`, `isconnection`

<b>Purpose</b>	True if valid Bloomberg connection
<b>Syntax</b>	<code>x = isconnection(Connect)</code>
<b>Arguments</b>	<code>Connect</code> Bloomberg connection object created with the <code>bloomberg</code> command.
<b>Description</b>	<code>x = isconnection(Connect)</code> returns <code>x = 1</code> if the connection is a valid Bloomberg connection, and <code>x = 0</code> if it is not.
<b>Examples</b>	<p>The command</p> <pre>c = bloomberg(8194, '111.222.33.444')</pre> <p>establishes a Bloomberg connection, <code>c</code>.</p> <p>Then</p> <pre>x = isconnection(c) x = 1</pre> <p>indicates that <code>c</code> is a valid Bloomberg connection.</p>
<b>See Also</b>	<code>bloomberg</code> , <code>close</code> , <code>fetch</code> , <code>get</code>

---

## Hyperfeed Functions

This section provides detailed descriptions of the Hyperfeed functions in the Datafeed Toolbox.

<code>close</code>	Close connection
<code>fetch</code>	Request data
<code>get</code>	Get connection properties
<code>hyperfeed</code>	Connect to Hyperfeed
<code>isconnection</code>	True if valid connection

<b>Purpose</b>	Close Hyperfeed connection		
<b>Syntax</b>	<code>close(Connect)</code>		
<b>Arguments</b>	<table><tr><td><code>Connect</code></td><td>Hyperfeed connection object created with the hyperfeed command.</td></tr></table>	<code>Connect</code>	Hyperfeed connection object created with the hyperfeed command.
<code>Connect</code>	Hyperfeed connection object created with the hyperfeed command.		
<b>Description</b>	<code>close(Connect)</code> closes the connection to the Hyperfeed data server.		
<b>See Also</b>	<code>hyperfeed</code>		

# fetch

---

## Purpose

Request data from Hyperfeed

## Syntax

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
data = fetch(Connect, 'Security', 'Fields', 'Date')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')
```

## Arguments

Connect	Hyperfeed connection object created with the hyperfeed command.
Security	A MATLAB string or cell array of strings containing the names of a securities in a format recognizable by the Hyperfeed server.
<i>Fields</i>	A MATLAB string or cell array of strings indicating the data fields for which data is to be retrieved. Some possible values are: Symbol Last Date Time Change Open High Low Volume
Date	Date string or serial date number indicating date for the requested data. If today's date is entered, yesterday's data is returned.
FromDate	Beginning date for historical data.



`ToDate`                      End date for historical data.

`Period`                      Period within date range. *Period* values are:

                                  d (daily)

                                  w (weekly)

                                  m (monthly)

                                  v (dividends)

**Description**

`data = fetch(Connect, 'Security')` returns data for all fields from Hyperfeed’s web site for the indicated securities.

`data = fetch(Connect, 'Security', 'Fields')` returns data for the specified fields.

`data = fetch(Connect, 'Security', 'Date')` returns all security data for the requested date.

`data = fetch(Connect, 'Security', 'Fields', 'Date')` returns security data for the specified fields on the requested date.

`data = fetch(Connect, 'Security', 'FromDate', 'ToDate')` returns security data for the date range `FromDate` to `ToDate`.

`data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')` returns security data for the specified fields for the date range `FromDate` to `ToDate`.

`data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')` returns security data for the date range `FromDate` to `ToDate` with the indicated period.

**Examples**

Obtain the closing price for Coca Cola on April 6, 2000.

```
c = hyperfeed('History');

ClosePrice = fetch(c,'ko','Close','Apr 6 00')

ClosePrice =

          730582.00          45.75
```

# fetch

---

## **See Also**

close, get, hyperfeed, isconnection (Hyperfeed functions)

---

<b>Purpose</b>	Get Hyperfeed connection properties				
<b>Syntax</b>	<code>value = get(Connect, 'PropertyName')</code>				
<b>Arguments</b>	<table><tr><td><code>Connect</code></td><td>Hyperfeed connection object created with the hyperfeed command.</td></tr><tr><td><code>PropertyName</code></td><td>(optional) A MATLAB string or cell array of strings containing property names. Property names are: Connection IPAddress Port Socket Version</td></tr></table>	<code>Connect</code>	Hyperfeed connection object created with the hyperfeed command.	<code>PropertyName</code>	(optional) A MATLAB string or cell array of strings containing property names. Property names are: Connection IPAddress Port Socket Version
<code>Connect</code>	Hyperfeed connection object created with the hyperfeed command.				
<code>PropertyName</code>	(optional) A MATLAB string or cell array of strings containing property names. Property names are: Connection IPAddress Port Socket Version				
<b>Description</b>	<p><code>value = get(Connect, 'PropertyName')</code> returns the value of the specified properties for the Hyperfeed connection object.</p> <p><code>value = get(Connect)</code> returns a MATLAB structure where each field name is the name of a property of <code>Connect</code>, and each field contains the value of that property.</p>				
<b>Examples</b>	<p>Use the hyperfeed command to establish a connection to Hyperfeed.</p> <pre>c = hyperfeed('Price')</pre> <p>Now use the get command to retrieve the connection property value.</p> <pre>h = get(c, Connection)</pre> <pre>h=</pre> <pre>connection: 3 table: 'Price'</pre>				
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>hyperfeed</code> , <code>isconnection</code> (Hyperfeed functions)				

# hyperfeed

---

<b>Purpose</b>	Connect to Hyperfeed
<b>Syntax</b>	<code>Connect = hyperfeed(<i>Table</i>)</code>
<b>Arguments</b>	<i>Table</i> Indicates the Hyperfeed table (database) to access. Possible values are: 'Price' (Default) 'Profile' 'Timeseries' 'History'
<b>Description</b>	<code>Connect = hyperfeed(<i>Table</i>)</code> connects to the indicated Hyperfeed table.
<b>Examples</b>	<pre>c = hyperfeed('Price ')</pre> connects to the Hyperfeed Price table.
<b>See Also</b>	<code>close, fetch, get, isconnection</code> (Hyperfeed functions)

<b>Purpose</b>	True if valid Hyperfeed connection
<b>Syntax</b>	<code>x = isconnection(Connect)</code>
<b>Arguments</b>	<code>Connect</code> Hyperfeed connection object created with the hyperfeed command.
<b>Description</b>	<code>x = isconnection(Connect)</code> returns <code>x = 1</code> if the connection is a valid Hyperfeed connection, and <code>x = 0</code> if it is not.
<b>Examples</b>	<p>The command</p> <pre>    c = hyperfeed</pre> <p>establishes a Hyperfeed connection, <code>c</code>, to the Price table.</p> <p>Then</p> <pre>    x = isconnection(c);</pre> <pre>    x = 1</pre> <p>indicates that <code>c</code> is a valid Hyperfeed connection.</p>
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>get</code> , <code>hyperfeed</code> (Hyperfeed functions)

---

## **FT Interactive Data Functions**

This section provides detailed descriptions of the FT Interactive Data functions in the Datafeed Toolbox.

<code>close</code>	Close connection
<code>fetch</code>	Request data
<code>get</code>	Get connection properties
<code>idc</code>	Connect to IDC
<code>isconnection</code>	True if valid connection

**Purpose** Close FT Interactive Data connection

**Syntax** `close(Connect)`

**Arguments** Connect FT Interactive Data connection object created with the `idc` command.

**Description** `close(Connect)` closes the connection to the FT Interactive Data server.

**Examples**

```
c = idc
establishes an FT Interactive Data connection, c.

close(c)
closes this connection.
```

**See Also** `idc`

# fetch

---

**Purpose** Request data from FT Interactive Data

**Syntax**

```
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate',
            'Period')
data = fetch(Connect, 'String', 'Lookup', 'Type', 'Market',
            NumRecords, StartRecord)
data = fetch(Connect, '', 'Lookup', 'Category')
data = fetch(Connect, '', 'GUILookup', 'GUICategory')
```

**Arguments**

Connect	FT Interactive Data connection object created with the <code>idc</code> command.
Security	A MATLAB string containing the name of a security in a format recognizable by the FT Interactive Data server.
<i>Fields</i>	A MATLAB string or cell array of strings indicating specific fields for which data is to be provided. Valid field names are in the file <code>@idc/idcfields.mat</code> . The variable <code>bbfieldnames</code> contains the list of field names.
FromDate	Beginning date for historical data.
ToDate	End date for historical data.
<i>Period</i>	Period within date range.
String	Search string.
<i>Type</i>	Lookup type. Possible values are: F (Fields) S (Securities)
<i>Market</i>	Market to search.
NumRecords	Number of record to fetch.
StartRecord	Starting record for fetch.



<i>Category</i>	Lookup category. Possible values are: F (All valid field categories) S (All valid security categories)
<i>GUICategory</i>	GUI category. Possible values are: F (All valid field categories) S (All valid security categories)

**Description**

`data = fetch(Connect, 'Security', 'Fields')` returns data for the indicated fields of the designated securities.

`data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')` returns historical data for the indicated fields of the designated securities.

`data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate', 'Period')` returns historical data for the indicated fields of the designated securities with the designated period.

`data = fetch(Connect, 'String', 'Lookup', 'Type', 'Market', NumRecords, StartRecord)` returns data of the requested type by searching within the designated market for the string.

`data = fetch(Connect, '', 'Lookup', 'Category')` returns all valid field or security categories.

`data = fetch(Connect, '', 'GUILookup', 'GUICategory')` opens the F'T Interactive Data dialog for selecting fields or securities.

**Examples**

```
d = fetch(c,'ford','lookup','s','equity',4,1)
```

returns the first four securities containing the string 'ford' starting with the first record.

**See Also**

close, get, idc, isconnection

# get

---

## Purpose

Get FT Interactive Data connection properties

## Syntax

```
value = get(Connect, 'PropertyName')  
value = get(Connect)
```

## Arguments

<i>Connect</i>	FT Interactive Data connection object created with the <code>idc</code> command.
<i>PropertyName</i>	(optional) A MATLAB string or cell array of strings containing property names. Property names are: Connected Connection Queued

## Description

`value = get(Connect, 'PropertyName')` returns the value of the specified properties for the FT Interactive Data connection object. `'PropertyName'` is a string or cell array of strings containing property names.

`value = get(Connect)` returns a MATLAB structure. Each field name is the name of a property of `Connect`, and each field contains the value of that property.

## See Also

`close`, `fetch`, `idc`, `isconnection` (FT Interactive Data functions)

---

<b>Purpose</b>	Connect to FT Interactive Data
<b>Syntax</b>	<code>Connect = idc</code>
<b>Description</b>	<code>Connect = idc</code> connects to the FT Interactive Data Corporation data server. <code>Connect</code> is a connection handle used by other functions to obtain data.
<b>Examples</b>	<pre>c = idc</pre> <p>makes a connection to the FT Interactive Data server.</p>
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>get</code> , <code>isconnection</code> (FT Interactive Data functions)

# isconnection

---

<b>Purpose</b>	True if valid FT Interactive Data connection		
<b>Syntax</b>	<code>x = isconnection(Connect)</code>		
<b>Arguments</b>	<table><tr><td><code>Connect</code></td><td>FT Interactive Data connection object created with the <code>idc</code> command.</td></tr></table>	<code>Connect</code>	FT Interactive Data connection object created with the <code>idc</code> command.
<code>Connect</code>	FT Interactive Data connection object created with the <code>idc</code> command.		
<b>Description</b>	<code>x = isconnection(Connect)</code> returns <code>x = 1</code> if the connection is a valid FT Interactive Data connection, and <code>x = 0</code> if it is not.		
<b>Examples</b>	<p>The command</p> <pre>c = idc</pre> <p>establishes an FT Interactive Data connection, <code>c</code>.</p> <p>Then</p> <pre>x = isconnection(c) x = 1</pre> <p>indicates that <code>c</code> is a valid FT Interactive Data connection.</p>		
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>get</code> , <code>idc</code> (FT Interactive Data functions)		

---

## **Yahoo Functions**

This section provides detailed descriptions of the Yahoo functions in the Datafeed Toolbox.

<code>close</code>	Close connection
<code>fetch</code>	Request data
<code>get</code>	Get connection properties
<code>isconnection</code>	True if valid connection
<code>yahoo</code>	Connect to Yahoo

# close

---

**Purpose** Close Yahoo connection

**Syntax** `close(Connect)`

**Arguments** Connect                      Yahoo connection object created with the yahoo command.

**Description** `close(Connect)` closes the connection to the Yahoo data server.

**See Also** yahoo

**Purpose** Request data from Yahoo

**Syntax**

```

data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
data = fetch(Connect, 'Security', 'Fields', 'Date')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')

```

**Arguments**

Connect	Yahoo connection object created with the yahoo command.
Security	A MATLAB string or cell array of strings containing the names of a securities in a format recognizable by the Yahoo server.
<i>Fields</i>	A MATLAB string or cell array of strings indicating the data fields for which data is to be retrieved. Possible values are:  Symbol Last Date Time Change Open High Low Volume
Date	Date string or serial date number indicating date for the requested data. If today's date is entered, yesterday's data is returned.
FromDate	Beginning date for historical data.

# fetch

---

<code>ToDate</code>	End date for historical data.
<code>Period</code>	Period within date range. <i>Period</i> values are: d (daily) w (weekly) m (monthly) v (dividends)

## Description

`data = fetch(Connect, 'Security')` returns data for all fields from Yahoo's web site for the indicated securities.

`data = fetch(Connect, 'Security', 'Fields')` returns data for the specified fields.

`data = fetch(Connect, 'Security', 'Date')` returns all security data for the requested date.

`data = fetch(Connect, 'Security', 'Fields', 'Date')` returns security data for the specified fields on the requested date.

`data = fetch(Connect, 'Security', 'FromDate', 'ToDate')` returns security data for the date range FromDate to ToDate.

`data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')` returns security data for the specified fields for the date range FromDate to ToDate.

`data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')` returns security data for the date range FromDate to ToDate with the indicated period.

## Examples

Obtain the closing price for Coca Cola on April 6, 2000.

```
c = yahoo;  
  
ClosePrice = fetch(c, 'ko', 'Close', 'Apr 6 00')  
  
ClosePrice =  
  
730582.00      45.75
```



**See Also**

close, get, isconnection, yahoo (Yahoo functions)

# get

---

**Purpose** Get Yahoo connection properties

**Syntax** `value = get(Connect, 'PropertyName')`

**Arguments**

<code>Connect</code>	Yahoo connection object created with the yahoo command.
<code>PropertyName</code>	(optional) A MATLAB string or cell array of strings containing property names. Currently the only property name recognized is <code>url</code> .

**Description** `value = get(Connect, 'PropertyName')` returns the value of the specified properties for the Yahoo connection object.

`value = get(Connect)` returns a MATLAB structure where each field name is the name of a property of `Connect`, and each field contains the value of that property.

**Examples** Use the yahoo command to establish a connection to Yahoo.

```
c = yahoo
```

```
c =
```

```
url: 'http://quote.yahoo.com'
```

Now use the `get` command to retrieve the connection property value.

```
get(c, url )
```

```
ans =
```

```
url: 'http://quote.yahoo.com'
```

**See Also** `close`, `fetch`, `isconnection`, `yahoo` (Yahoo functions)

<b>Purpose</b>	True if valid Yahoo connection		
<b>Syntax</b>	<code>x = isconnection(Connect)</code>		
<b>Arguments</b>	<table><tr><td>Connect</td><td>Yahoo connection object created with the yahoo command.</td></tr></table>	Connect	Yahoo connection object created with the yahoo command.
Connect	Yahoo connection object created with the yahoo command.		
<b>Description</b>	<code>x = isconnection(Connect)</code> returns <code>x = 1</code> if the connection is a valid Yahoo connection, and <code>x = 0</code> if it is not.		
<b>Examples</b>	<p>The command</p> <pre>c = yahoo</pre> <p>establishes a Yahoo connection, <code>c</code>.</p> <p>Then</p> <pre>x = isconnection(c) x = 1</pre> <p>indicates that <code>c</code> is a valid Yahoo connection.</p>		
<b>See Also</b>	<code>close</code> , <code>fetch</code> , <code>get</code> , <code>yahoo</code> (Yahoo functions)		

# yahoo

---

## Purpose

Connect to Yahoo

## Syntax

```
Connect = yahoo
Connect = yahoo('URL', 'IPAddress', PortNumber)
```

## Arguments

URL	Must be 'http://quote.yahoo.com'
IPAddress	A MATLAB string containing the internet address of proxy server machine.
PortNumber	Port number on proxy server.

## Description

Connect = yahoo verifies that the URL `http://quote.yahoo.com` is accessible and creates a connection handle.

Connect = yahoo('URL', 'IPAddress', PortNumber) connects to Yahoo through a proxy server using the IP address and port number provided. This form of the yahoo command may be required when connecting to Yahoo from behind an internal firewall.

## Examples

Use the yahoo command to establish a connection to the Yahoo data server.

```
Connect = yahoo

Connect =

    url: 'http://quote.yahoo.com'
```

Use the yahoo command to establish a connection to the Yahoo data server, providing an IP address and port number on a proxy server.

```
Connect = yahoo('http://quote.yahoo.com', '111.222.33.444', 5678)

Connect =

    url: 'http://quote.yahoo.com'
    ip: '111.222.33.444'
    port: 5678
```

## See Also

close, fetch, get, isconnection (Yahoo functions)

# Directory Structure

---

**Table A-1: <matlab>/datafeed/datafeed/@bloomberg**

<b>File</b>	<b>Purpose</b>
bloomberg	Connect to Bloomberg data server
close	Close connection to Bloomberg data server
fetch	Request data from Bloomberg data server
get	Get Bloomberg connection properties
isconnection	True if valid Bloomberg connection

**Table A-2: <matlab>/datafeed/datafeed/@hyperfeed**

<b>File</b>	<b>Purpose</b>
close	Close connection to Hyperfeed data server
fetch	Request data from Hyperfeed data server
get	Get Hyperfeed connection properties
hyperfeed	Connect to Hyperfeed data server
isconnection	True if valid Hyperfeed connection

**Table A-3: <matlab>/datafeed/datafeed/@idc**

<b>File</b>	<b>Purpose</b>
close	Close connection to FT Interactive Data data server
fetch	Request data from FT Interactive Data data server
get	Get FT Interactive Data connection properties
idc	Connect to FT Interactive Data data server
isconnection	True if valid FT Interactive Data connection

---

**Table A-4: <matlab>/datafeed/datafeed/@yahoo**

<b>File</b>	<b>Purpose</b>
close	Close connection to Yahoo data server
fetch	Request data from Yahoo data server
get	Get Yahoo connection properties
isconnection	True if valid Yahoo connection
yahoo	Connect to Yahoo data server

**Table A-5: <matlab>/datafeed/dfgui**

<b>File</b>	<b>Purpose</b>
dftool	Datafeed graphical user interface





## B

Bloomberg  
    connection handle 1-4  
    connection object 1-4  
bloomberg 1-3, **2-3**

## C

close 1-5  
    Bloomberg **2-4**  
    FT Interactive Data **2-19**  
    hyperfeed **2-11**  
    yahoo **2-26**  
connecting 1-3  
connection handle 1-4  
connection object 1-4  
**Connection** tab 1-13

## D

data  
    default 1-6  
    field 1-8  
    header 1-6  
    historical 1-10  
    time series 1-9  
**Data** tab 1-15  
**Datafeed** dialog box 1-13  
default data 1-6  
dftool **1-13**  
disconnecting 1-5

## F

fetch 1-6  
    Bloomberg **2-5**  
    FT Interactive Data **2-20**

    hyperfeed **2-12**  
    yahoo **2-27**  
field data 1-8  
field names 1-8  
Flag values 1-6

## G

get 1-4, 1-5  
    Bloomberg **2-8**  
    FT Interactive Data **2-22**  
    hyperfeed **2-15**  
    yahoot **2-30**  
GETDATA argument 1-8  
graphical user interface 1-13

## H

HEADER argument 1-6  
header data 1-6  
header fields 1-6  
historical data 1-10  
HISTORY argument 1-10  
hyperfeed **2-16**

## I

idc **2-23**  
IP address 1-3  
isconnection 1-4  
    Bloomberg **2-9**  
    FT Interactive Data **2-24**  
    hyperfeed **2-17**  
    yahoo **2-31**

## **L**

LOOKUP argument 1-11

## **M**

markets 1-11, 2-6

MATLAB root directory A-1

## **P**

port number 1-3

## **R**

retrieve properties 1-4

root directory

    MATLAB A-1

## **S**

**Securities Lookup** dialog box 1-13, 1-17

## **T**

ticker symbols 1-11

time series data 1-9

TIMESERIES argument 1-9

typographical conventions (table) v

## **V**

verification 1-4

## **Y**

yahoo **2-32**