

User Guide for GetQuote v1.1

Table of Contents

1 Release Notes.....	2
2 Introduction.....	2
3 Installation.....	2
4 GETQUOTE.....	3
4.1 Basic Usage.....	3
4.1.1 Financial Symbols.....	4
4.1.2 Quote Types.....	5
4.2 Advanced Usage.....	5
4.2.1 Recalculating Results.....	5
4.2.2 Array Function Call.....	6
4.3 Examples.....	6
4.3.1 Example 1.....	6
4.3.2 Example 2.....	7
4.3.3 Example 3.....	7
4.3.4 Example 4.....	8
4.3.5 Example 5.....	8
4.3.6 Example 6.....	9
5 GETHISTORY.....	10
5.1 Usage.....	10
5.2 Examples.....	11
5.2.1 Example 1.....	11
5.2.2 Example 2.....	11
6 Troubleshooting.....	14
6.1 Issue 1.....	14
6.2 Issue 2.....	14

1 Release Notes

Many new quote types have been added to the GETQUOTE function in this latest release. They are listed in Section 4.1.2 of the User Guide.

A new function for getting historical quotes has been added. It is described in Section 5 of the User Guide.

Installing the new add-in should replace the old one, but if it does not, (1) uninstall all previous versions of the add-in using the Extension Manager, (2) shut down all instances of OpenOffice (logging off and/or rebooting can't hurt), and then reinstall the add-in. Section 3 of the User Guide gives instructions for using the Extension Manager.

2 Introduction

GetQuote is an add-in to the OpenOffice.org Calc spreadsheet application. It creates two functions: GETQUOTE and GETHISTORY. The GETQUOTE function returns the most recent price, volume, yield, etc. of financial indexes and issues traded on the major global exchanges. The GETHISTORY function returns similar information but for a range of dates. Both functions use a web service provided by the Yahoo! Finance web site, <http://finance.yahoo.com>.

3 Installation

The GetQuote add-in is a file with the OXT extension. On most systems, double-clicking on this file will cause OpenOffice to start its “Extension Manager” and install the plug-in. If this does not work, take the following steps to start the Extension Manager and install the OXT.

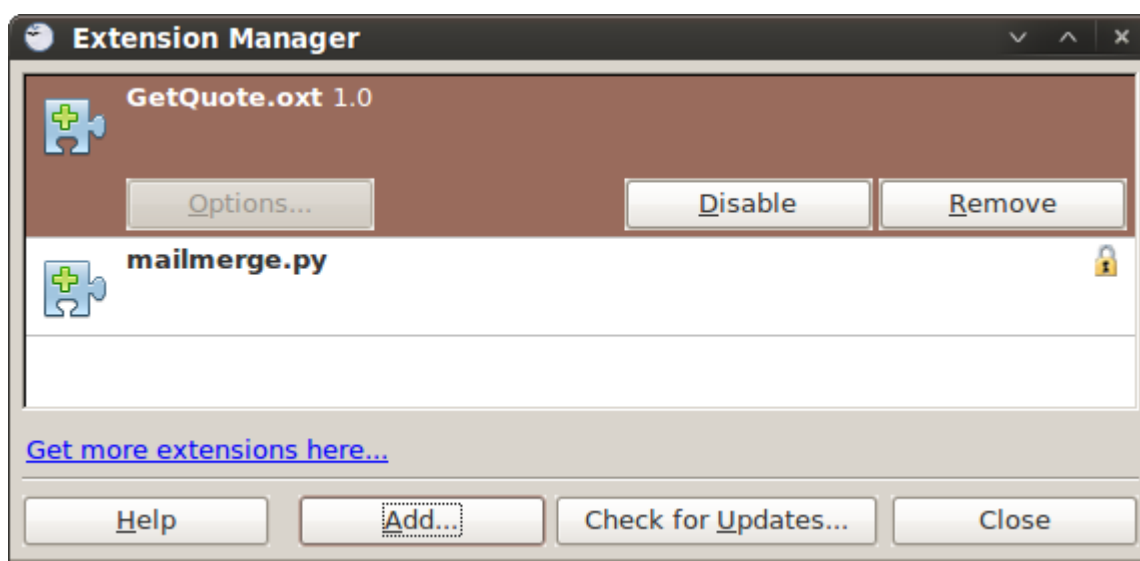


Illustration 1: Extension Manager

Step 1: Start OpenOffice and open the Extension Manager by picking “Extension Manager...” from the “Tools” menu as shown in Illustration 2.

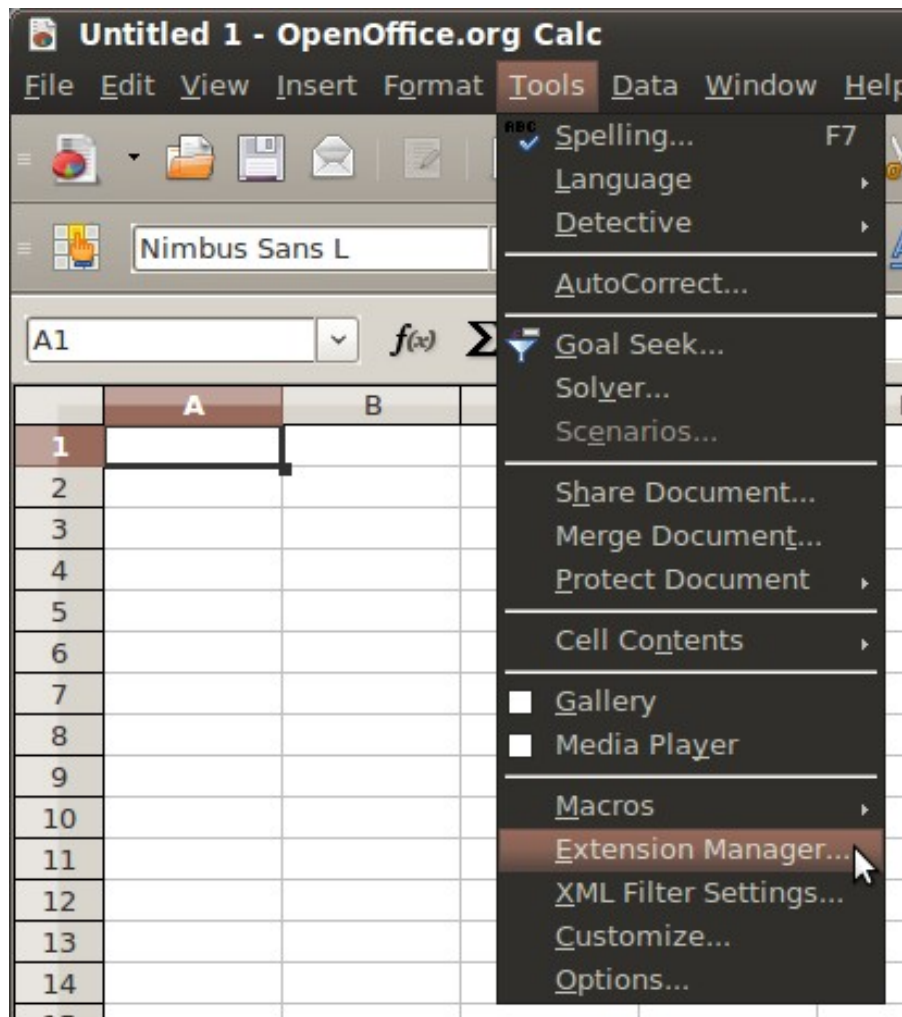


Illustration 2: Extension Manager Menu Item

Step 2: Uninstall any old version of GetQuote by selecting them and pressing the “Remove” button.

Step 3: Push the “Add...” button on the Extension Manager (shown at the bottom of Illustration 1) and select the “GetQuote.OXT” file.

Step 4: Push the “Close” button to close the Extension Manager and begin using the add-in.

If you encounter difficulty during installation, please read the Troubleshooting section of this document.

4 GETQUOTE

4.1 Basic Usage

A single call to the GETQUOTE function can return one quote for one symbol (Basic Use) or many quotes for many symbols (Advanced Use). The number of quotes returned is determined by the number and complexity of the arguments. Example 1 and Example 2 demonstrate basic use of the function.

The function accepts at most two arguments. The first argument is the financial symbol and is required. It is a short name that may or may not be identical to the ticker of the issue. The second argument is the quote type and is optional. It tells the function what kind of quote to fetch.

4.1.1 Financial Symbols

GETQUOTE accepts too many symbols to list in the user guide. The function can look up quotes for all the major exchanges, indexes, and currencies. The table below gives some examples to illustrate the kinds of symbols supported by the function.

Symbol	Description	Currency
HON	Honeywell International (New York Stock Exchange)	US \$
^NDX	NASDAQ-100 Index (NASDAQ Exchange)	Not Applicable
JPY=X	Currency exchange rate: Japanese Yen to the US Dollar (¥/\$)	¥
SGR.L	Shore Capital GRP (London Stock Exchange)	£
^HSI	HANG SENG Index (Hong Kong Stock Exchange)	Not Applicable
BUMI.JK	Bumi Resources Tbk (Indonesian Stock Exchange)	Rp
LOPO.PK	Low Power Technology (Pink Sheets)	US \$

Table 1: Symbol Examples

To look up a symbol, visit the web site <http://finance.yahoo.com>. In the upper left corner, type in the name of a company, index, or currency as shown in Illustration 3.

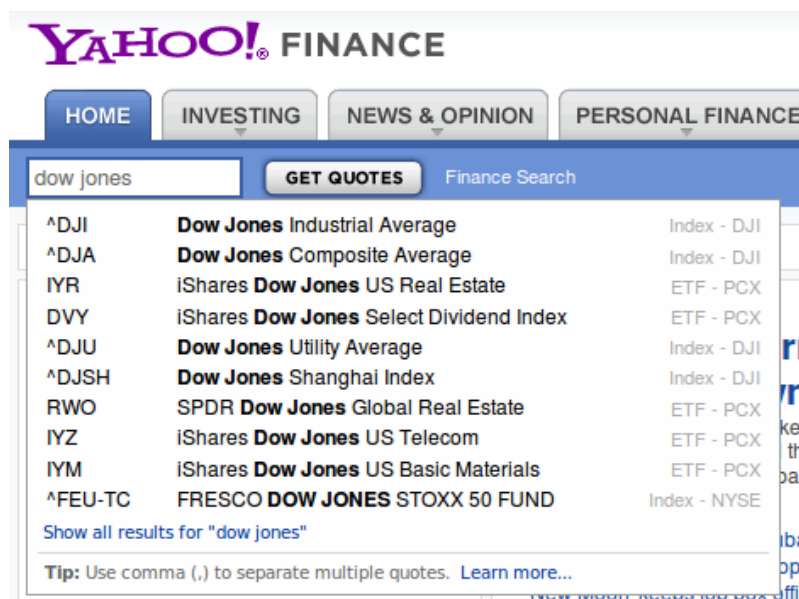


Illustration 3: Symbol Lookup

A symbol may be entered into its own cell (without quotes) and then referenced in the function call, or it may be surrounded by double quotation marks and entered directly into the function as shown in Example 1.

4.1.2 Quote Types

GETQUOTE can fetch 31 types of quotes. The types are numbered from 0 to 30. The function accepts these numbers as the second argument.

0 = Last traded price	16 = Change from 52-week low
1 = Change in price for the day	17 = 52-week high
2 = Opening price of the day	18 = Change from 52-week high
3 = High price of the day	19 = 50-day Moving Average
4 = Low price of the day	20 = 200-day Moving Average
5 = Volume	21 = Previous Close
6 = Average Daily Volume	22 = Price/Earning Ratio
7 = Ask Price	23 = Dividend Yield
8 = Bid Price	24 = Price/Sales
9 = Book Value	25 = Price/Book
10 = Dividend/Share	26 = PEG Ratio
11 = Earnings/Share	27 = Price/EPS Estimate Current Year
12 = Earnings/Share Estimate Current Year	28 = Price/EPS Estimate Next Year
13 = Earnings/Share Estimate Next Year	29 = Short Ratio
14 = Earnings/Share Estimate Next Quarter	30 = 1-year Target Price
15 = 52-week low	

Table 2: Quote Types

A quote type number may be entered into its own cell and then referenced in the function call, or it may be entered directly into the function as shown in Example 2.

4.2 Advanced Usage

4.2.1 Recalculating Results

To cause GETQUOTE to recalculate a result, the entire spreadsheet must be reloaded. Select the File | Reload menu item as shown in Illustration 4.

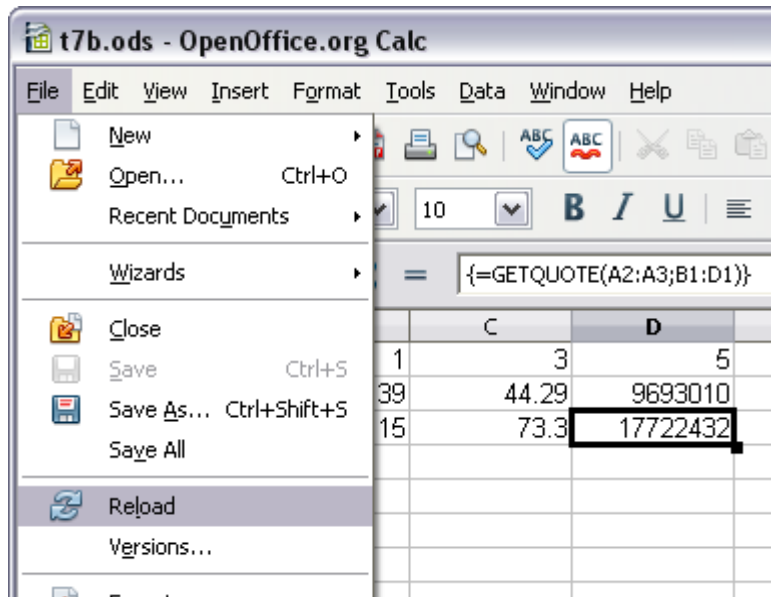


Illustration 4: Reload The Spreadsheet

4.2.2 Array Function Call

A single call to the GETQUOTE add-in is capable of returning an array of values when it receives multiple symbols and/or multiple quote types. The examples starting at Example 3 demonstrate this type of usage.

Using the add-in in this way can greatly reduce the time needed to load spreadsheets that contain many symbols and quote types.

4.3 Examples

The first two examples in this section demonstrate basic use of the function. The remaining examples demonstrate how to

4.3.1 Example 1

In this basic example, the function returns the last traded price of a symbol using only the first argument. Illustration 5 shows how to specify the symbol as a literal value. Illustration 6 shows how to specify the symbol as the contents of a cell.

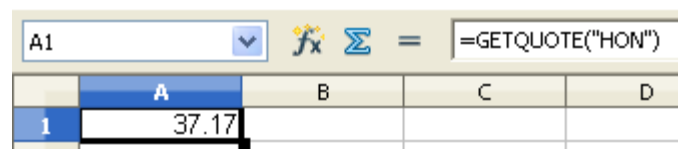


Illustration 5: Single Quoted Symbol

	A	B	C	D
1	HON	37.17		

Illustration 6: Single Cell Reference

4.3.2 Example 2

In this basic example, the function returns the last traded price of a symbol using both arguments. Illustration 7 shows how to specify the quote type and symbol as literal values. Illustration 8 shows how to specify the quote type and symbol as the contents of a cell.

	A	B	C	D
1	37.17			

Illustration 7: Two Literal Arguments

	A	B	C	D
1	HON	0	37.17	

Illustration 8: Two Cell References

4.3.3 Example 3

In this advanced example, the function returns a row of quotes for a single symbol. First, select the full range of cells that will contain the result--in this example, a row of five. Second, type the function without the outermost curly brackets:

`=GETQUOTE("HON"; { 0 ; 1 ; 2 ; 3 ; 4 ; 5 })`

When using literal values as shown in Illustration 9, the second argument is a list of types separated by semi-colons and surrounded by curly brackets. Third, press the Shift-Ctrl-Enter key combination. Illustration 10 shows how to specify the quote types as the contents of a range of cells from a single row.

	A	B	C	D	E	F
1	37.17	0.33	36.83	37.24	36.63	3588418

Illustration 9: Row of Quotes (Literal Arguments)

	A	B	C	D	E	F	G
1		0	1	2	3	4	5
2	HON	37.17	0.33	36.83	37.24	36.63	3588418
3							

Illustration 10: Row of Quotes (Cell References)

4.3.4 Example 4

In this advanced example, the function returns a column of quotes for two symbols. First, select the full range of cells that will contain the result--in this example, a column of two. Second, type the function without the outermost curly brackets:

=GETQUOTE({ "HON" | "BA" })

When using quoted symbols as shown in Illustration 11, the first argument is a list of quoted symbols separated by vertical bars and surrounded by curly brackets. Third, press the Shift-Ctrl-Enter key combination. Illustration 12 shows how to specify the symbols as the contents of a range of cells from a single column.

	A	B	C	D
1	37.17			
2	52.69			
3				

Illustration 11: Column of Last Traded Prices (Quoted Symbols)

	A	B	C	D
1	HON	37.17		
2	BA	52.69		
3				

Illustration 12: Column of Last Traded Prices (Cell Range)

4.3.5 Example 5

In this advanced example, the function returns a column of quotes for a single symbol. First, select the full range of cells that will contain the result--in this example, a column of three. Second, type the function without the outermost curly brackets:

=GETQUOTE(B1 ; { 1 | 2 | 3 })

When using literal values as shown in Illustration 13, the second argument is a list of quote types (numbers) separated by vertical bars and surrounded by curly brackets. Third, press the Shift-Ctrl-

Enter key combination. Illustration 14 shows how to specify the quote types as the contents of a range of cells from a single column.

	A	B	C	D
1	Symbol	HON		
2	Change	0.33		
3	Open	36.83		
4	High	37.24		

Illustration 13: Column of Quotes (Literal Arguments)

	A	B	C	D
1		HON		
2	1	0.33		
3	2	36.83		
4	3	37.24		

Illustration 14: Column of Quotes (Cell Range)

4.3.6 Example 6

In this advanced example, the function returns three quotes for each of two symbols (totaling 6 quotes). First select the full range of cells that will contain the result—in this example, a range with two rows and three columns. Second, type the function without the outermost curly brackets:

=GETQUOTE({ "HON" | "BA" } ; { 1 ; 3 ; 5 })

When using literal values as shown in Illustration 15, the first argument is the list of symbols separated by vertical bars since there is a row for each symbol. The second argument is the list of quote types separated by semicolons since there is a column for each quote type. Third, press the Shift-Ctrl-Enter key combination. Illustration 16 shows how to specify the arguments as the contents of ranges of cells.

	A	B	C	D	E
1	0.33	37.24	3588418		
2	0.39	52.75	2591224		
3					

Illustration 15: Array of Quotes (Literal Arguments)

	A	B	C	D
1		1	3	5
2	HON	0.33	37.24	3588418
3	BA	0.39	52.75	2591224

Illustration 16: Array of Quotes (Cell Ranges)

5 GETHISTORY

5.1 Usage

A single call to the GETHISTORY function returns a range of past and present quotes for a single symbol. It accepts four arguments: Symbol, Type, Start Date, and End Date.

The Symbol argument is similar to the GETQUOTE argument of the same name, documented under Section 4.1.1. Unlike GETQUOTE, it will only return prices for a single symbol.

The Type argument is a single number that controls what kind of values are returned in the columns of the result. It also has an effect on the number of rows in the results:

Type	Name	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
0	Dividends	Date	Amount per share	(Not Applicable)				
1	Daily	Date	Open	High	Low	Close	Volume	Adjusted Close
2	Weekly							
3	Yearly							

The Start and End dates define the range of dates the function will return. Both date arguments are optional. If a date is left blank, today's date will be used in its place. Both dates must be on or before today. If the start date comes after the end date, the rows will be returned in reverse chronological order.

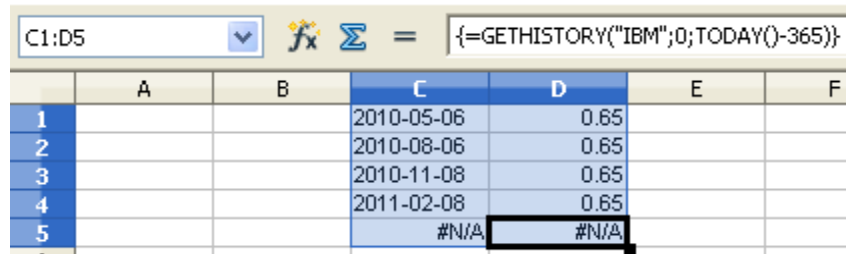
The GETHISTORY function must be invoked as an array function call. This entails selecting a range of cells in the worksheet and then either using the "Shift-Ctrl-Enter" key combination (when typing the function) or by checking the "Array" box in the lower left corner of the dialog (when using the Function Wizard).

All of these arguments may be either directly entered into the function or entered into cells referenced by the function.

5.2 Examples

5.2.1 Example 1

This is an example of fetching dividends issued over the past year. First select a range of cells that has two columns and as many rows as all the dates in the range. We think that the company usually issues quarterly dividends, but we aren't sure. We select 5 rows, expecting the last row to be invalid.



	A	B	C	D	E	F
1			2010-05-06	0.65		
2			2010-08-06	0.65		
3			2010-11-08	0.65		
4			2011-02-08	0.65		
5			#N/A	#N/A		

Illustration 17: Past Year's Dividends

Next, we type in the function with a type of 0 and start date 365 days before today:

```
=GETHISTORY("IBM"; 0; today()-365; )
```

Finally, pressing “Shift-Ctrl-Enter” fills in the selection with the four quarterly dividends issued over the past year to date. We know that we are viewing all the dividends of the past year because the last row is invalid.

5.2.2 Example 2

In this example, we wish to view the monthly prices for the last full calendar year and we want them in reverse chronological order.

First we select a range of cells that has 7 columns and 12 rows. Next, we bring up the Function Wizard, select the GETHISTORY function, and check the “Array” box in the lower left-hand corner:

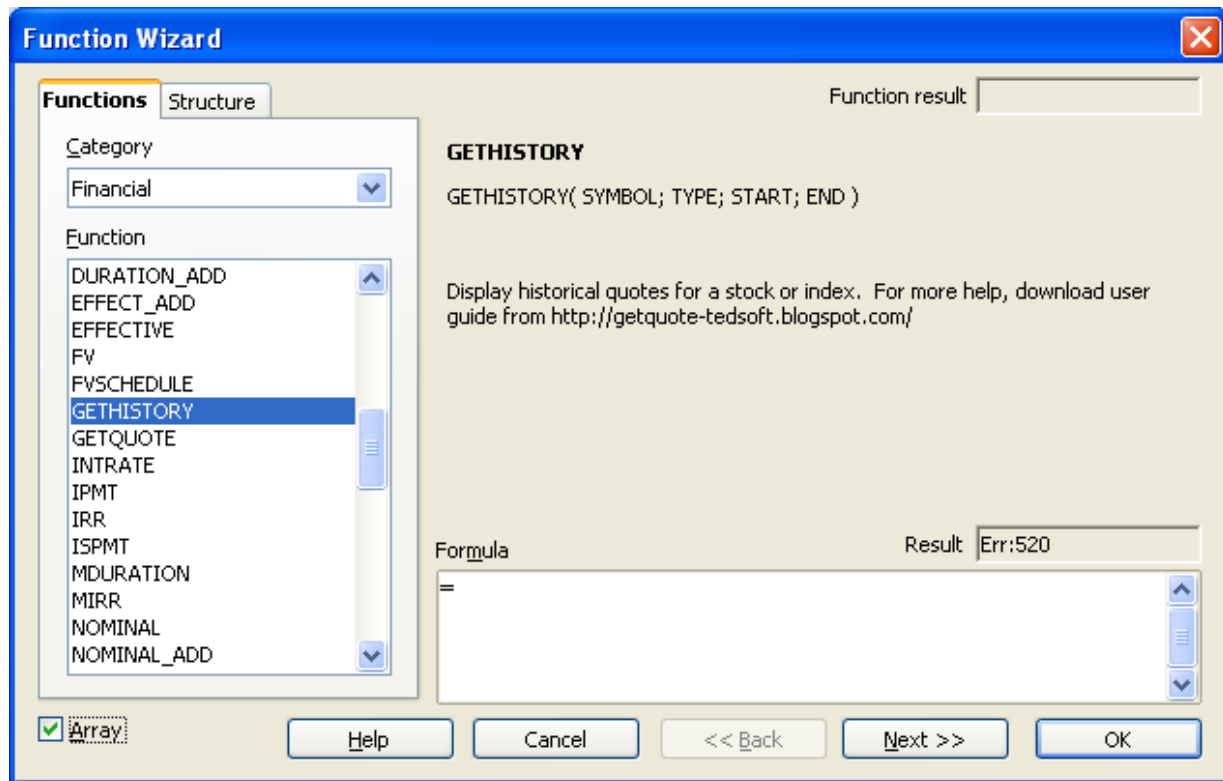


Illustration 18: Selecting GETHISTORY in Function Wizard

Next, we enter the quoted symbol, type 3, start date (the last day of the year) as quoted text, and end date (the first day of the year) using the DATE function:

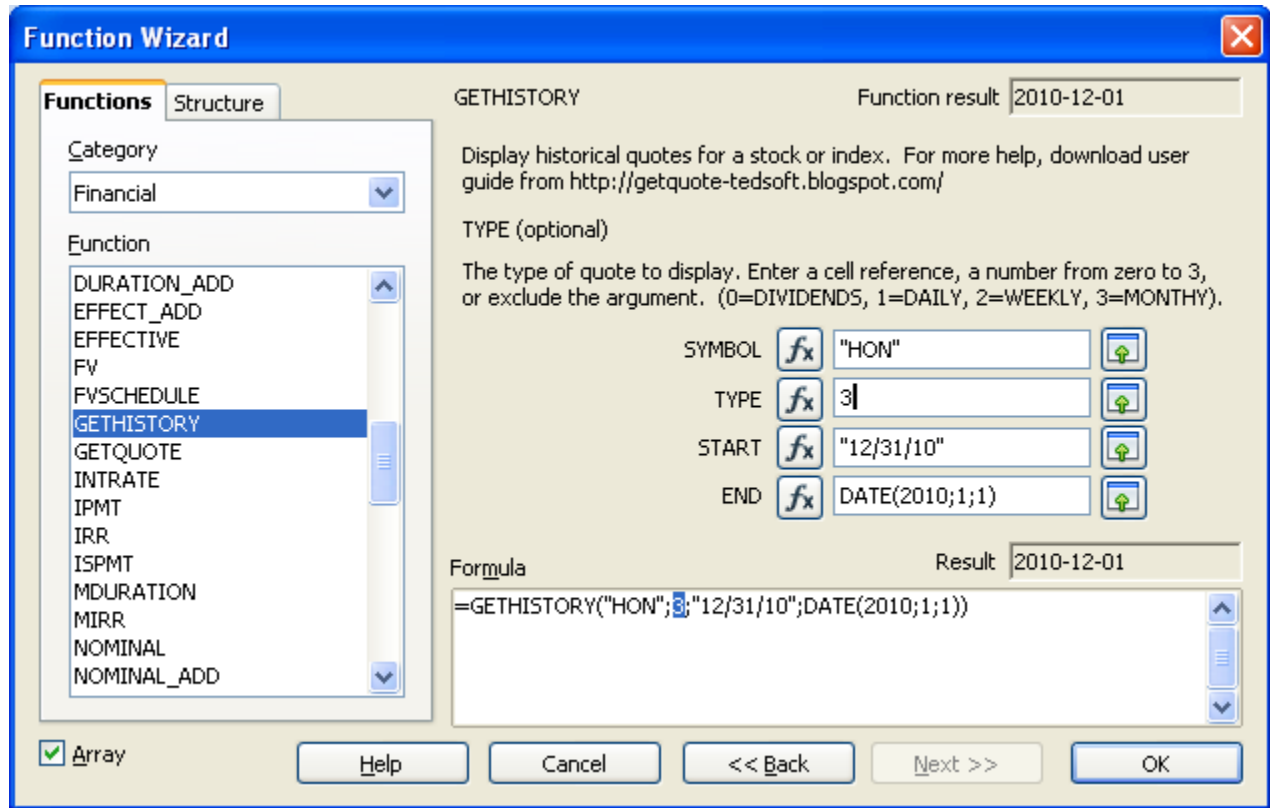


Illustration 19: Entering GETHISTORY Arguments in Function Wizard

Finally, we press “OK” and our selection is populated with the results:

	A	B	C	D	E	F	G	H	I
1	Date	Open	High	Low	Close	Volume	Adj. Close		
2	2010-12-01	50.41	53.74	50.41	53.16	3940600	52.85		
3	2010-11-01	46.93	50.44	46.56	49.71	5511400	49.42		
4	2010-10-01	44.22	48.28	43.41	47.11	5026100	46.54		
5	2010-09-01	39.59	44.89	39.45	43.94	5103100	43.41		
6	2010-08-02	43.66	44.17	38.08	39.06	4758500	38.59		
7	2010-07-01	38.99	44.3	37.89	42.86	5452100	42.04		
8	2010-06-01	42.24	46.56	38.54	39.03	6461300	38.29		
9	2010-05-03	47.81	48.53	39.98	42.77	7091100	41.95		
10	2010-04-01	45.51	48.63	44.72	47.47	5421100	46.25		
11	2010-03-01	40.25	45.78	40.25	45.27	5783600	44.11		
12	2010-02-01	38.92	40.85	36.68	40.16	5918300	39.13		
13	2010-01-04	39.61	43.21	37.58	38.64	6924800	37.36		

Illustration 20: Past Year's Monthly Prices (Reversed)

6 Troubleshooting

The troubleshooting section describes common issues and the steps that can be taken to resolve them.

6.1 Issue 1

When attempting to install the add-in, OpenOffice reports an error: “Could not create Java implementation loader”.



Illustration 21: Error Message 1

Try enabling Java in OpenOffice. First, select the Tools | Options... menu. Second, select “Java” on the left hand side of the dialog. Third, look on the right and make sure the “Use a Java run time environment” box is checked. Fourth, select a 1.5 or later version of Java. Fifth, push “OK”. Close OpenOffice, restart it, and try installing the add-in again.

6.2 Issue 2

When attempting to install the add-in after removing a previous version of it, OpenOffice reports an error “Rejecting types due to incompatibility!”:

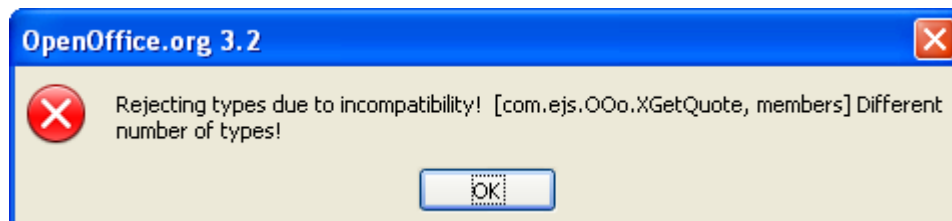


Illustration 22: Error Message 2

Shut down all instances of OpenOffice (all documents, spreadsheets, etc.). You might even log off or reboot to make sure all instances are truly closed. Then try to install the new version of the add-in again.