Economic Events Collection for MultiCharts

Created by Jura @ BMT (JoshM on MC forum), with support from Nicolas11 @ BMT
BigMikeTrading.com Multicharts.com/Discussion/

The ‘Economic Events Collection’ is a collection of MultiCharts functions that make it possible to very easily work with future and historical economic events.

The Economic Events Collection was designed for flexibility and usability – giving it endless possibilities while keeping the amount of code required to an absolute minimum.

For whom might the Economic Events Collection for MultiCharts be useful?

- If you’re an automated trader and want to close open positions 10 minutes before a high impact economic event is released;
- If you’re a discretionary trader and don’t want to be caught off guard with important economic releases or speeches and summits about market moving topics;
- If you’re an option trader and want to open volatility-based option strategies (like straddles and strangles) in the 30 minutes before important, market moving news;
- If you’re a backtester and want to backtest the market reactions on certain historical economic events, or just don’t want to hold a position when a historical economic event happened (like a Non-Farm Payroll release).

How to start using the Economic Events Collection

The steps below detail the actions that need to be undertaken before you can work with the Economic Events Collection.

If you already have the ELCollections.DLL, you can skip the first step.

1. Get the ELCollections.DLL.
    a. Go to here to find the 32bit version or here to find the 64bit version.
    b. Download and extract the .zip file with the ELCollections.DLL.
    c. Make sure that MultiCharts and the accompanying processes are *not* running.
    d. Place the ELCollections.DLL in the MultiCharts Program Files folder (which in most cases will be C:\Program Files\MultiCharts\ or C:\Program Files\TS Support\MultiCharts\).

   e. ***Note:*** Though the ELCollections.zip comes with a collection of functions (in the ELCollections.ELD file), these aren't necessary to install in order to use the Economic Events Collection.

2. Import the Economic Events Collection functions pack.

   a. **Tip:** Check the box "Compile on Import" to prevent having to manually compile the 50+ functions.

   b. The 'Economic Events' functions *include* the ELCollections functions that are used by the Economic Events Collection. If you have already imported these at an earlier point, you can skip importing them. However, all Economic Events Collection functions (starting with "EE.") *will* need to be imported.

3. Make a directory on your pc and extract the Java executable in the "EconomicEventsDownloader.zip" to this folder.

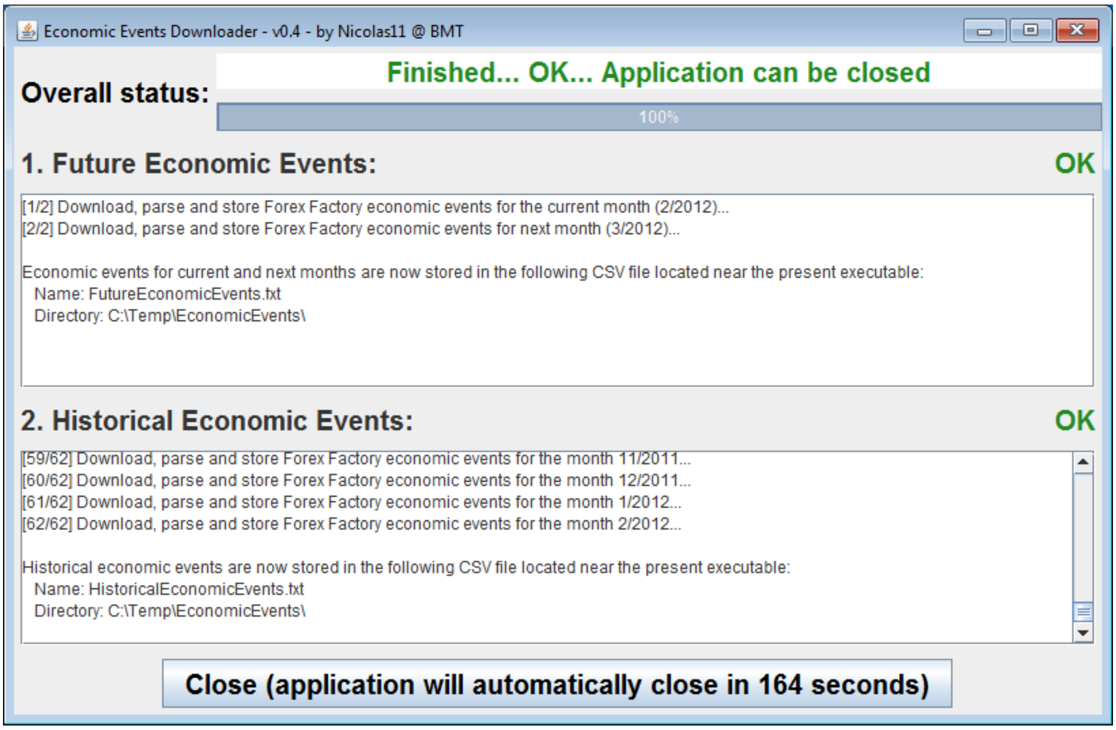
   a. *Note:* If you're concerned about this executable, extract the "EconomicEventsDownloader_sources_v*.zip" to verify the Java code and compile it yourself.

4. Double-click on the "EconomicEventsDownloader.jar" file in the directory and the Economic Events Calendar from the ForexFactory website will be downloaded and saved in two text files.

   a. **Tip*:*** Since the program doesn't require any user input, you can schedule it with Windows Task Scheduler to run, for instance, every morning or every weekend. That way you'll have the most recent ForexFactory Economic Events calendar available for use in MultiCharts, and will your Historical Economic Events file contain the most historical economic events for use in backtesting.

   b. If Java is not installed on your pc, it can be downloaded for free from Java.com.

5. The download process will create two text files in the directory where the Java program is placed in: a "FutureEconomicEvents.txt" file that holds the economic events for the current and next month, and a "HistoricalEconomicEvents.txt" that contains the historical events from 2007 till yesterday. If the download process went okay, you'll see a screen similar to the one below.



Overall status: Finished... OK... Application can be closed

100%

1. Future Economic Events: OK

```
[1/2] Download, parse and store Forex Factory economic events for the current month (2/2012)...
[2/2] Download, parse and store Forex Factory economic events for next month (3/2012)...

Economic events for current and next months are now stored in the following CSV file located near the present executable:
  Name: FutureEconomicEvents.txt
  Directory: C:\Temp\EconomicEvents\
```

2. Historical Economic Events: OK

```
[59/62] Download, parse and store Forex Factory economic events for the month 11/2011...
[60/62] Download, parse and store Forex Factory economic events for the month 12/2011...
[61/62] Download, parse and store Forex Factory economic events for the month 1/2012...
[62/62] Download, parse and store Forex Factory economic events for the month 2/2012...

Historical economic events are now stored in the following CSV file located near the present executable:
  Name: HistoricalEconomicEvents.txt
  Directory: C:\Temp\EconomicEvents\
```

6. Open the function ‘EE.UserSettings’ from the Economic Events Collection in the PowerLanguage Editor and specify your user settings (such as directory and time zone). When in doubt, follow the instructions in the comments in the file. After entering you settings, **Compile** (F3) the EE.UserSettings file.

7. Read the rest of the manual and *then* start coding!

General notes on using the Economic Events Collection

Some general notes and tips about using the Economic Events Collection:

- Though it might not be mandatory to assign the output of a function to a variable, it’s recommended to do so and generally considered ‘good programming practice’. In the examples in this document, the variable *Value1* is used for this.

- Practically every function of the Economic Events Collection needs a string variable called *NameOfList*. If you create an economic events list, you’ll need to remember (by variable or typing the name manually) the name that you gave to that list so you’ll can use it for further reference.

- The Economic Events list are *shared* lists, meaning that the same list can be accessed by other indicators or signals other than the one that created it. Keep in mind though that you’ll need to use the same *NameOfList* string variable when referencing a specific list.

- Multiple Economic Events list can be made as long as each one has an unique name.

- *Understanding of the following is **not** needed to use the Economic Events Collection, but technically speaking*: The Economic Events Collection is build using the ELCollection functions to read the data and store these in lists. It uses a shared combined list of lists consisting out of three numerical lists (for date, time, and impact) and two string lists (for region and event). This data import, conversion to variables, storing them in a list is done in the EE.ReadFile function.

Functions of the Economic Events Collection

Creating an Economic Events list

```
Value1 = EE.GetFutureEvents(NameOfList (string));
```

This function creates an economic events list with all future events as found in the text file with the future economic events, including this month's economic events that have already passed.

To make an economic events list called "myList", type EE.GetFutureEvents("myList").

Note: Use the EE.CleanUp function to remove already passed events from a list.

```
Value1 = EE.GetNextDaysEvents(NameOfList (string), NumberOfDays (numeric));
```

This function creates a list with the future economic events for the next few days, including the current day.

For example, EE.GetNextDaysEvents("myList", 2) will create an economic events list named *myList* with the events for today and tomorrow. Using EE.GetNextDayEvents with 1 day will return only the economic events for today.

Notes about the EE.GetNextDaysEvents function:

- The economic events list will not contain passed economic events from previous days.
- The function uses *calendar days* instead of working days.
- This function will also add economic events for today to the list that might have already happened. Use the EE.CleanUp function to removes these.

```
Value1 = EE.GetHistoricalEvents(NameOfList (string), FromDate (numeric),
ToDate (numeric));
```

This function creates a list with the historical events loaded from the "HistoricalEconomicEvents.txt" file and specified by the FromDate and ToDate (both in *YYYMMdd* format).

For example, to create an economic events list with the historical events from January 1st, 2008, to and including January 1st, 2009, type EE.GetHistoricalEvents("myList", 1080101, 1090101).

> **Tip**: Use the EE.StrToDate function to make the usage of the EE.GetHistoricalEvents easier. That way, instead of having to provide the dates in *YYYMMdd* format, you could type: EE.GetHistoricalEvents(theList, EE.StrToDate("01/01/2008"), EE.StrToDate("01/01/2009")).

To create an economic events list for the past 100 days, starting from the date of the current bar, type EE.GetHistoricalEvents("myList", JulianToDate(DateToJulian(Date) - 100), Date).

```
Value1 = EE.LoadStoredList(NameOfList (string), FileName (string));
```

Use this function to load a *custom* created Economic Events list from the specified text file. This loaded economic events list will be given the name specified in the NameOfList string.

For example, to load the previously stored economic events list from the "myCustomList.txt" file in the "newList" economic events list, type EE.LoadStoredList("newList", "myCustomList.txt").

This functions loads a list previously stored with the EE.StoreListInFile function; see the section *"General functions for working with the list of Economic Events"* further down in this document for specifics about that function.

See the examples section at the end of this document for an example of using both functions to store a custom economic events list.

Manipulating a Economic Events list

Working with dates of the Economic Events list

```
Value1 = EE.RemoveDate(NameOfList (string), DateToRemove (numeric));
```

This function removes the specified date (in EasyLanguage *YYYMMdd* format) from the specified list.

For example, to remove all economic events from today from a list, type EE.RemoveDate("myList", CurrentDate).

To remove all economic events happening on the day after tomorrow, type EE.RemoveDate("myList", JulianToDate(DateToJulian(CurrentDate) + 2)).

To remove all economic events happening on July 14th, 2009, type EE.RemoveDate("myList", EE.StrToDate("14/07/2009")).

```
Value1 = EE.KeepDate(NameOfList (string), DateToKeep (numeric));
```

This function only keeps the specified date (in EasyLanguage *YYYMMdd* format) in the list and removes all other dates.

For example, to only keep the current date on the list, type EE.KeepDate("myList", CurrentDate).

```
Value1 = EE.RemoveDateRange(NameOfList (string), FromDate (numeric), ToDate
(numeric));
```

This function removes the specified date range (in EasyLanguage *YYYMMdd* format) from the specified list. This includes both the *From* and *To* date.

For example, to remove the date range of today, tomorrow, and the day after tomorrow, type EE.RemoveDateRange("myList", CurrentDate, JulianToDate(DateToJulian(CurrentDate) + 2)).

```
Value1 = EE.KeepDateRange(NameOfList (string), FromDate (numeric), ToDate
(numeric));
```

This function removes all dates, except those specified by the date range (in EasyLanguage *YYYMMdd* format) from the specified economic events list.

This includes both the *From* and *To* date, so if your specified date range comprises three days, only the middle day will remain.

Working with <u>times</u> of the Economic Events list

Note: the functions below use the time of the economic event in the user his Local time zone.

```
Value1 = EE.RemoveTime(NameOfList (string), TimeToRemove (numeric));
```

This function removes all events with a certain time from the specified economic events list, where the time is in 24-hour *HHmm* format.

For example, to remove all economic events occurring at 10 am, type EE.RemoveTime("myList", 1000).

```
Value1 = EE.KeepTime(NameOfList (string), TimeToKeep (numeric));
```

This function removes all economic events from the specified list except events with the time as specified by the user (in 24-hour *HHmm* format).

For example, EE.KeepTime("myList", 1430) will remove all economic events from the *myList* list except those occurring at 14:30.

```
Value1 = EE.RemoveTimeRange(NameOfList (string), TimeFrom (numeric), TimeTo
(numeric));
```

This function removes all economic events from the specified list that fall in a certain by the user specified time range (in 24-hour *HHmm* format).

Note that if you specify '800' and '1200' here, economic events occurring at 8 am and those at 12 noon will also be deleted. If you want to keep those, you'll need to specify '801' and '1159', like EE.RemoveTimeRange("myList", 801, 1159).

```
Value1 = EE.KeepTimeRange(NameOfList (string), TimeFrom (numeric), TimeTo
(numeric));
```

This function removes all economic events from the specified list except those that fall inside a certain specified time range (in 24-hour *HHmm* format).

See the examples section further down in this document for an example.

Working with impact of Economic Events

Note: the Economic Events Impact is specified by a number as given by ForexFactory, where:

- 0 = Non-economic (for example a bank holiday),
- 1 = Low impact expected,
- 2 = Medium impact expected,
- 3 = High impact expected.

```
Value1 = EE.RemoveImpact(NameOfList (string), ImpactToRemove (numeric));
```

This function removes all economic events with a certain impact from the specified list.

For example, EE.RemoveImpact("myList", 2) will remove all 'Medium Impact Expected' events from the *myList* economic events list.

```
Value1 = EE.KeepImpact(NameOfList (string), ImpactToKeep (numeric));
```

This function removes all economic events except the ones with the impact as specified by the user from a specified list.

For example, to only keep 'High Impact Expected' events on the *myList* economic events list, type EE.KeepImpact("myList", 3).

Working with regions of the Economic Events list

```
Value1 = EE.RemoveRegions(NameOfList (string), RegionsToRemove (string));
```

This function removes the specified region(s) from the specified economic events list.

For example, EE.RemoveRegions("myList", "JPY") will remove all Japanese economic events from the *myList* economic events list.

If you want to remove multiple regions in one go, separate them by a separator like a comma (",") or semicolon (";").

For example, to remove all USD, CHF and AUD regions, type EE.RemoveRegions("myList", "USD,CHF,AUD"). Also see the example in the examples section of this document.

Note: the actual separator that you use won’t matter, but keep in mind the following rules when using this function:

- Each region symbol needs to be separated by one separator (for example, this isn’t allowed: “EUR,,USD”),
- A separator at the end of the string isn’t allowed (for example, “JPY,AUD,”),
- Spaces between the regions and/or separator isn’t allowed (for example, “JPY, USD, GBP”).

```
Value1 = EE.KeepRegions(NameOfList (string), RegionsToKeep (string));
```

This function only keeps the specified region(s) in the specified economic events list, and all other regions are removed.

A trader only interested in American events can type EE.KeepRegions(“myList”, “USD”) to only keep American economic events on his *myList* economic events list.

Or, to only keep USD and EUR events in an economic events list, type EE.KeepRegions(“myList”, “EUR,USD”).

Note: the same comments as those from the EE.RemoveRegions function apply here.

Working with the <u>events</u> of the Economic Events list

```
Value1 = EE.RemoveEvents(NameOfList (string), EventsToRemove (string));
```

This function removes economic events from the specified list that are *similar* as the EventsToRemove string specified by the user.

For example, to remove all “manufacturing” economic release from the *myList* economic events list, type EE.RemoveEvent(“myList”, “manufacturing”).

With this function it’s also possible to remove multiple economic events from the specified list in one go. For example, if you can’t stand to hear anything about the/an euro crisis, type EE.RemoveEvents(myList, "ECB,French,German,Draghi").

Note: Keep in mind the following important notes when using this function:

- The input is **not** case sensitive; in the string comparisons performed by the function “Draghi” equals “draghi” and “EcB” equals “ECB”.
- The only allowed separator is a comma (“**,**”) - don’t use any other separator.
- You can use spaces in the EventsToRemove string, but don’t use spaces between the separators.

This will work: “ECB Press Conference,Bond,Bernanke Speaks” while this won’t work: “ECB Press Conference, Bond, Bernanke Speaks” (note the spaces after the comma’s).

- The function looks for *similar* strings, so if you specify to remove “German”, it will remove economic events like “German Consumer Confidence”, “German Bond Auction” and “German ZEW Index” from the list.

- **Warning**: since it’s not known to the function how many events you specify, it uses a while loop to determine the number of events and place these in an array. If you enter an erroneous input, you’ll run the risk of getting stuck in an infinite while loop, which will inevitably crash MultiCharts and/or the Power Language Editor. However, the chances of this occurring are pretty slim (the while loop won’t be entered for example if you use a wrong separator), but you’ll need to be aware of this since it *can* happen.

```
Value1 = EE.KeepEvents(NameOfList (string), EventsToKeep (string));
```

This function works the same as the EE.RemoveEvents function, except that this function only keeps the event in the specified economic events list that are similar as the string provided by the user.

For example, to only keep PPI data in the *myList* list, type EE.KeepEvents(“myList”, “PPI”).

With this function it’s also possible to specify multiple events that needs to be kept on the list. For example, to only keep some macro-economic events related to consumers on the list, type EE.KeepEvents(“myList”, “Employment,Unemployment,Housing,Confidence”).

See the examples section further down in this document for another example.

Note: the same notes as those of the EE.RemoveEvents function apply here.

Miscellaneous operations on a list of economic events

```
Value1 = EE.RemoveList(NameOfList (string));
```

This function removes the *whole* specified economic events list from the computer memory. There’s no undoing from this, so if you accidentally delete the list, you’ll need to reload it.

Note: the actual text file with the economic events is **not** deleted, only the by the user created list in the working memory.

If you reference the list *after* you have removed it, you’ll trigger an “List index is out of bounds” runtime error.

```
Value1 = EE.CleanUp(NameOfList (string));
```

This function removes all the past events (i.e. the economic events that happened prior to the current date and current time) from the specified list.

General functions for working with the list of Economic Events

```
EE.IndexNumberNextEvent(NameOfList (string), GivenDate (numeric), GivenTime
(numeric));
```

This function returns the economic events list IndexNumber for the next event given the specified date (in *YYYMMdd* format) and time (in 24 hour *HHmm* format).

The 'next event' is here specified as the first event in the economic events list that is going to happen given that specified date and time.

For example, to get the IndexNumber of the first next future economic event, type EE.IndexNumberNextEvent("myList", CurrentDate, CurrentTime).

To get the IndexNumber for the next historical event relative to the bar's date and time the function is called on, type EE.IndexNumberNextEvent("myList", Date, Time).

This is a *very* useful function, since knowing the IndexNumber of the next event, you can use the EE.DateList through EE.EventList functions (see the section below) to retrieve that specific event and it's properties from your list.

By incrementing this IndexNumber you can also perform actions on the future economic events that happen after the first next event.

Tip: to prevent having MultiCharts call this function every bar or tick, combine it with the EE.MinutesToNextEvent function described below and only call it conditionally.

```
EE.MinutesToNextEvent(NameOfList (string), GivenDate (numeric), GivenTime
(numeric));
```

This function returns the number of minutes prior to the next event given the specified date (in *YYYMMdd* format) and time (in 24 hour *HHmm* format).

For example, to get the number of minutes to the next future economic event, type EE.MinutesToNextEvent("myList", CurrentDate, CurrentTime).

Or, to get the number of minutes to the next historical event relative to the bar's date and time, type EE.MinutesToNextEvent("myList", Date, Time).

Tip: use this function conditionally to ensure better performance, instead of calling it every tick or bar close.

```
EE.DateTimeOfEvent(NameOfList (string), IndexNumber (numeric));
```

This function returns the numeric DateTime value for a certain by the user specified economic event on the specified list.

For example, EE.DateTimeOfEvent("myList", 1) will return 40946.50 if the first economic event from the *myList* economic events list occurs on February 7th, 2012 at 12:00.

```
EE.EventsOnDate(NameOfList (string), GivenDate (numeric));
```

This function returns the number of economic events from a specified economic events list that happen on a certain specified date (in *YYYMMdd* format).

For example, to get the economic events that happen today, type EE.EventsOnDate("myList", CurrentDate).

To get the number of economic events that happen on the current bar date, type EE.EventsOnDate("myList", Date).

To get the number of economic events that happen tomorrow, type EE.EventsOnDate("myList", JulianToDate(DateToJulian(CurrentDate) + 1)).

```
Value1 = EE.StoreListInFile(NameOfList (string), FileName (string));
```

Use this function to store your manually created Economic Events list in an external text file in the file directory as specified in the EE.UserSettings function.

This has the benefit of having only to create a historical economic events list once, and after performing your filtering operations you can save this list in a file, and load it the next time when needing it.

For example, to store your custom *myList* Economic Events list in the text file "myCustomList", type EE.StoreListInFile("myList", "myCustomList.txt").

Some notes about using this function:

- EE.StoreListInFile uses FileAppend, so if you specifiy a file name that already exists your Economic Events list gets mingled up with the already existing contents of the file.

- Also, if you create an economic events list it will remain in memory till you close MultiCharts or call EE.RemoveList. So, you don't need to store and load your custom list every time you want to use it – but only the first time you start up MultiCharts for that day.

- When the list is stored in the text file, the list in the working memory of the pc is **not** deleted. You'll need to manually call the EE.RemoveList function to do this.

Functions to get specific values of one of the Economic Events lists:

Note: the lists with economic events are non-zero based, so the first value will be '1' and not zero. See the examples section in this document for an example of how to use these function to loop through an economic events list.

```
EE.DateList(NameOfList (string), IndexNumber (numeric));
```

Returns the date (in *YYYMMdd* format) of the specified economic events list accompanying with that index number.

For example, EE.DateList(“myList”, 2) will return the date for the second economic event on the *myList* economic events list.

```
EE.TimeList(NameOfList (string), IndexNumber (numeric));
```

Returns the time (in 24-hour *HHmm* format) of an economic event on the list accompanying with that index number.

For example, EE.TimeList(“myList”, 5) will return the time of the fifth economic event on the *myList* economic events list.

```
EE.ImpactList(NameOfList (string),  IndexNumber (numeric));
```

Returns the (numeric) impact of an economic event on the list accompanying that index number.

For example, EE.ImpactList(“myList”, 3) will return ‘1’ for the third event on the *myList* economic events list if this event has a “Low Impact Expected”.

> Note: you can use the EE.ImpactToString function to convert this numeric value to its meaning, for example EE.ImpactToString(EE.ImpactList(“myList”, 3)) would return the string “Low”, signaling that the third event on the specified list has a low expected impact.

```
EE.RegionList(NameOfList (string), IndexNumber (numeric));
```

Returns the name of the region of an economic event on the list given the index number.

For example, EE.RegionList(“myList”, 10) will return “USD” if the tenth economic event on the *myList* economic events list is an American economic event.

```
EE.EventList(NameOfList (string), IndexNumber (numeric));
```

Returns the name of the specific economic event on the list given the index number.

For example, EE.EventList(“myList”, 4) will return “Manufacturing PMI” if the fourth economic event on the “myList” economic events list is the Manufacturing PMI.

Other Economic Events functions

```
Value1 = EE.UserSettings(FileDirectory (string ref), FileNameFutureEvents
(string ref), FileNameHistoricalEvents (string ref), TimeZoneOffSet
(numeric ref));
```

In this function, the user stores his or her settings so that these can be retrieved ‘behind the scenes’ by the other functions. This has the benefit of only having to change one file for the settings.

```
Value1 = EE.ReadFile(FileName (string), FromDate (numeric), ToDate
(numeric), NameOfList (string), TimeZoneOffSet (numeric));
```

This function is the main function of the Economic Events Collection and reads the text file, converts it to variables, and put these in the lists. This is the only function that reads and converts the data.

```
EE.GetMaxIndex(NameOfList (string));
```

This function returns the numeric length of the specified economic events list. For example, this might be useful if the user wants to loop through the economic events list.

See the examples in the example section of this document.

```
EE.GetListID(NameOfList (string));
```

This function returns the numeric list ID for the shared combined list of list of the ELCollections. This function is called by the other functions 'behind the scenes', and normally won't be needed by the user.

```
EE.ImpactToString(NumericImpact (numeric));
```

This function converts the numeric 'Expected Impact' value to a string.

For example, EE.ImpactToString(1) will return "Low" and EE.ImpactToString(3) gives "High".

Or to convert the impact number of the fifth economic event on the *myList* list to a string, type EE.ImpactToString(EE.ImpactList("myList", 5)).

Note: if the EE.ImpactToString function gives you "N/A" as a return, the number lays outside of the range 0 - 3.

```
Value1 = EE.PrintToOutputLog(NameOfList (string));
```

This function prints the specified list to the Output Log of the PowerLanguage Editor.

When constructing an economic events list, the displaying of the current contents of the economic events list is a frequent reoccurring task – by using this function the user doesn't have to manually create a loop every time.

```
EE.StrToDate(DateString (string));
```

This function translates the by the user specified date string to a numeric date in *YYYMMdd* format. See the example in the description of the EE.GetHistoricalEvents function about how it might be used.

This function accepts dates in *dd/MM/yy** and *dd/MM/yyyy** format, for example "09/02/12" and "09/02/2012".

*: the actual input for this function is dependent on your regional date format. If you're pc is set to US regional date format, you'll need to enter *MM/dd/yy* here – if you're in Europe you'll need to use the *dd/MM/yy* format.

Examples of using the Economic Events Collection

Looping through a list of future economic events

To loop through the first five values of the “myListName” list:

```
Variables:
       x(0), myList("myListName");

value1 = EE.GetFutureEvents(myList);

for x = 1 to 5 begin

                    Print("Date: ", EE.DateList(myList, x),
                    "   Time: ", EE.TimeList(myList, x),
                    "   Impact: ",EE.ImpactList(myList, x),
                    "   Region: ", EE.RegionList(myList, x),
                    "   Event: ", EE.EventList(myList, x)
                    );
end;
```

Which returns:

```
Date: 1120201.00   Time:  630.00   Impact:    2.00   Region: AUD   Event: Commodity Prices y/y
Date: 1120201.00   Time:  800.00   Impact:    2.00   Region: GBP   Event: Nationwide HPI m/m
Date: 1120201.00   Time:  915.00   Impact:    3.00   Region: CHF   Event: Retail Sales y/y
Date: 1120201.00   Time:  930.00   Impact:    2.00   Region: CHF   Event: SVME PMI
Date: 1120201.00   Time: 1000.00   Impact:    1.00   Region: EUR   Event: Final Manufacturing PMI
```

Looping through the list of future economic events for the next three days (including today)

For example:

```
Variables:
       x(0), myList("anExampleList");

value1 = EE.GetNextDaysEvents(myList, 3);

for x = 1 to EE.GetMaxIndex(myList) begin

       Print(
              "Date: ", FormatDate("dd-MM-yyyy",
ELDateToDateTime(EE.DateList(myList, x))),
              "   Time: ", FormatTime("HH:mm",
ELTimeToDateTime(EE.TimeList(myList, x))),
              "   Impact: ",EE.ImpactList(myList, x),
              "   Region: ", EE.RegionList(myList, x),
              "   Event: ", EE.EventList(myList, x)
              );
end;
```

Will return:

```
Date: 05-02-2012   Time: 00:30   Impact:    1.00   Region: AUD   Event: MI Inflation Gauge m/m
Date: 05-02-2012   Time: 01:30   Impact:    3.00   Region: AUD   Event: Retail Sales m/m
Date: 05-02-2012   Time: 01:30   Impact:    2.00   Region: AUD   Event: ANZ Job Advertisements m/m
Date: 06-02-2012   Time: 09:00   Impact:    3.00   Region: CHF   Event: Foreign Currency Reserves
Date: 06-02-2012   Time: 10:30   Impact:    2.00   Region: EUR   Event: Sentix Investor Confidence
Date: 06-02-2012   Time: 12:00   Impact:    2.00   Region: EUR   Event: German Factory Orders m/m
Date: 06-02-2012   Time: 16:00   Impact:    3.00   Region: CAD   Event: Ivey PMI
Date: 06-02-2012   Time: 22:45   Impact:    3.00   Region: NZD   Event: Labor Cost Index q/q
Date: 06-02-2012   Time: 23:30   Impact:    1.00   Region: AUD   Event: AIG Construction Index
Date: 06-02-2012   Time: 01:01   Impact:    2.00   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 06-02-2012   Time: 04:30   Impact:    3.00   Region: AUD   Event: Cash Rate
Date: 06-02-2012   Time: 04:30   Impact:    3.00   Region: AUD   Event: RBA Rate Statement
Date: 07-02-2012   Time: 06:00   Impact:    1.00   Region: JPY   Event: Leading Indicators
Date: 07-02-2012   Time: 08:45   Impact:    1.00   Region: EUR   Event: French Trade Balance
```

```
Date: 07-02-2012   Time: 12:00   Impact:    2.00   Region: EUR   Event: German Industrial Production m/m
Date: 07-02-2012   Time: 14:05   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 07-02-2012   Time: 14:30   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 07-02-2012   Time: 16:00   Impact:    3.00   Region: USD   Event: Fed Chairman Bernanke Testifies
Date: 07-02-2012   Time: 16:00   Impact:    1.00   Region: USD   Event: IBD/TIPP Economic Optimism
Date: 07-02-2012   Time: 21:00   Impact:    1.00   Region: USD   Event: Consumer Credit m/m
Date: 07-02-2012   Time: 00:30   Impact:    2.00   Region: AUD   Event: Westpac Consumer Sentiment
Date: 07-02-2012   Time: 00:50   Impact:    1.00   Region: JPY   Event: Bank Lending y/y
Date: 07-02-2012   Time: 00:50   Impact:    1.00   Region: JPY   Event: Current Account
Date: 07-02-2012   Time: 01:01   Impact:    1.00   Region: GBP   Event: BRC Shop Price Index y/y
```

Removing a specific date from an economic events list

To remove a specific date from a list:

```
Variables:
      x(0), myList("anExampleList"), dateToRemove(1120207);

value1 = EE.GetNextDaysEvents(myList, 3);

Print("Before removing of  date ", dateToRemove, "....");

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ",EE.ImpactToString(EE.ImpactList(myList, x)),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;

value1 = EE.RemoveDate(myList, dateToRemove);

Print("After removing date...");

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ",EE.ImpactToString(EE.ImpactList(myList, x)),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;
```

Which returns:

```
Before removing of  date 1120207.00....
Date: 1120205.00   Time:   30.00   Impact: Low   Region: AUD   Event: MI Inflation Gauge m/m
Date: 1120205.00   Time:  130.00   Impact: High   Region: AUD   Event: Retail Sales m/m
Date: 1120205.00   Time:  130.00   Impact: Medium   Region: AUD   Event: ANZ Job Advertisements m/m
Date: 1120206.00   Time:  900.00   Impact: High   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact: Medium   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact: Medium   Region: EUR   Event: German Factory Orders m/m
Date: 1120206.00   Time: 1600.00   Impact: High   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact: High   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact: Low   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact: Medium   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact: High   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact: High   Region: AUD   Event: RBA Rate Statement
Date: 1120207.00   Time:  600.00   Impact: Low   Region: JPY   Event: Leading Indicators
Date: 1120207.00   Time:  845.00   Impact: Low   Region: EUR   Event: French Trade Balance
Date: 1120207.00   Time: 1200.00   Impact: Medium   Region: EUR   Event: German Industrial Production m/m
Date: 1120207.00   Time: 1405.00   Impact: Low   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact: High   Region: CAD   Event: Building Permits m/m
Date: 1120207.00   Time: 1600.00   Impact: High   Region: USD   Event: Fed Chairman Bernanke Testifies
Date: 1120207.00   Time: 1600.00   Impact: Low   Region: USD   Event: IBD/TIPP Economic Optimism
Date: 1120207.00   Time: 2100.00   Impact: Low   Region: USD   Event: Consumer Credit m/m
Date: 1120207.00   Time:   30.00   Impact: Medium   Region: AUD   Event: Westpac Consumer Sentiment
Date: 1120207.00   Time:   50.00   Impact: Low   Region: JPY   Event: Bank Lending y/y
Date: 1120207.00   Time:   50.00   Impact: Low   Region: JPY   Event: Current Account
Date: 1120207.00   Time:  101.00   Impact: Low   Region: GBP   Event: BRC Shop Price Index y/y
After removing date...
Date: 1120205.00   Time:   30.00   Impact: Low   Region: AUD   Event: MI Inflation Gauge m/m
```

```
Date: 1120205.00   Time:  130.00   Impact: High   Region: AUD   Event: Retail Sales m/m
Date: 1120205.00   Time:  130.00   Impact: Medium   Region: AUD   Event: ANZ Job Advertisements m/m
Date: 1120206.00   Time:  900.00   Impact: High   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact: Medium   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact: Medium   Region: EUR   Event: German Factory Orders m/m
Date: 1120206.00   Time: 1600.00   Impact: High   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact: High   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact: Low   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact: Medium   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact: High   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact: High   Region: AUD   Event: RBA Rate Statement
```

Filter out all economic events falling outside a certain time range

For example, if the trader only trades from 9:00 till 15:00, then only the economic events that happen during this time are relevant. The EE.KeepTimeRange function is used in the example below to perform this kind of filtering:

```
Variables:
      x(0), myList("anExampleList"), timeFrom(900), timeTo(1500);

// Get the economic events for the next 7 days
value1 = EE.GetNextDaysEvents(myList, 7);

Print("Before removing of time range from ", timeFrom, "  to ", timeTo);

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;

value1 = EE.KeepTimeRange(myList, timeFrom, timeTo);

Print(NewLine, "After filtering out the economic events that fall outside of
the range..");

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;
```

Which gives the following output:

```
Before removing of time range from  900.00  to 1500.00
Date: 1120205.00   Time:   30.00   Impact:    1.00   Region: AUD   Event: MI Inflation Gauge m/m
Date: 1120205.00   Time:  130.00   Impact:    3.00   Region: AUD   Event: Retail Sales m/m
Date: 1120205.00   Time:  130.00   Impact:    2.00   Region: AUD   Event: ANZ Job Advertisements m/m
Date: 1120206.00   Time:  900.00   Impact:    3.00   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact:    2.00   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Factory Orders m/m
Date: 1120206.00   Time: 1600.00   Impact:    3.00   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact:    1.00   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact:    2.00   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: RBA Rate Statement
Date: 1120207.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Leading Indicators
Date: 1120207.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Trade Balance
Date: 1120207.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Industrial Production m/m
Date: 1120207.00   Time: 1405.00   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 1120207.00   Time: 1600.00   Impact:    3.00   Region: USD   Event: Fed Chairman Bernanke Testifies
```

```
Date: 1120207.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: IBD/TIPP Economic Optimism
Date: 1120207.00   Time: 2100.00   Impact:    1.00   Region: USD   Event: Consumer Credit m/m
Date: 1120207.00   Time:   30.00   Impact:    2.00   Region: AUD   Event: Westpac Consumer Sentiment
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Bank Lending y/y
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Current Account
Date: 1120207.00   Time:  101.00   Impact:    1.00   Region: GBP   Event: BRC Shop Price Index y/y
Date: 1120208.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Economy Watchers Sentiment
Date: 1120208.00   Time:  745.00   Impact:    1.00   Region: CHF   Event: Unemployment Rate
Date: 1120208.00   Time:  800.00   Impact:    1.00   Region: EUR   Event: German Trade Balance
Date: 1120208.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Gov Budget Balance
Date: 1120208.00   Time: 1415.00   Impact:    2.00   Region: CAD   Event: Housing Starts
Date: 1120208.00   Time: 1630.00   Impact:    2.00   Region: USD   Event: Crude Oil Inventories
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Employment Change q/q
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Unemployment Rate
Date: 1120208.00   Time:   50.00   Impact:    2.00   Region: JPY   Event: Core Machinery Orders m/m
Date: 1120208.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: M2 Money Stock y/y
Date: 1120208.00   Time:  230.00   Impact:    3.00   Region: CNY   Event: CPI y/y
Date: 1120208.00   Time:  230.00   Impact:    2.00   Region: CNY   Event: PPI y/y
Date: 1120209.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Household Confidence
Date: 1120209.00   Time:  700.00   Impact:    1.00   Region: JPY   Event: Prelim Machine Tool Orders y/y
Date: 1120209.00   Time:  745.00   Impact:    2.00   Region: CHF   Event: SECO Consumer Climate
Date: 1120209.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: Manufacturing Production m/m
Date: 1120209.00   Time: 1030.00   Impact:    2.00   Region: GBP   Event: Trade Balance
Date: 1120209.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: Industrial Production m/m
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Asset Purchase Facility
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Official Bank Rate
Date: 1120209.00   Time: 1345.00   Impact:    3.00   Region: EUR   Event: Minimum Bid Rate
Date: 1120209.00   Time: 1430.00   Impact:    2.00   Region: CAD   Event: NHPI m/m
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: EUR   Event: ECB Press Conference
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Unemployment Claims
Date: 1120209.00   Time: 1600.00   Impact:    2.00   Region: GBP   Event: NIESR GDP Estimate
Date: 1120209.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: Wholesale Inventories m/m
Date: 1120209.00   Time: 1630.00   Impact:    1.00   Region: USD   Event: Natural Gas Storage
Date: 1120209.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: CGPI y/y
Date: 1120209.00   Time:  130.00   Impact:    3.00   Region: AUD   Event: RBA Monetary Policy Statement
Date: 1120210.00   Time:  800.00   Impact:    1.00   Region: EUR   Event: German Final CPI m/m
Date: 1120210.00   Time:  845.00   Impact:    2.00   Region: EUR   Event: French Industrial Production m/m
Date: 1120210.00   Time:  915.00   Impact:    3.00   Region: CHF   Event: CPI m/m
Date: 1120210.00   Time: 1000.00   Impact:    1.00   Region: EUR   Event: Italian Industrial Production m/m
Date: 1120210.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: PPI Input m/m
Date: 1120210.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: PPI Output m/m
Date: 1120210.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Trade Balance
Date: 1120210.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Trade Balance
Date: 1120210.00   Time: 1555.00   Impact:    3.00   Region: USD   Event: Prelim UoM Consumer Sentiment
Date: 1120210.00   Time: 1555.00   Impact:    1.00   Region: USD   Event: Prelim UoM Inflation Expectations
Date: 1120210.00   Time: 1830.00   Impact:    3.00   Region: USD   Event: Fed Chairman Bernanke Speaks
Date: 1120210.00   Time: 1850.00   Impact:    2.00   Region: USD   Event: FOMC Member Pianalto Speaks
Date: 1120210.00   Time: 2000.00   Impact:    2.00   Region: USD   Event: Federal Budget Balance

After filtering out the economic events that fall outside of the range..
Date: 1120206.00   Time:  900.00   Impact:    3.00   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact:    2.00   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Factory Orders m/m
Date: 1120207.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Industrial Production m/m
Date: 1120207.00   Time: 1405.00   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 1120208.00   Time: 1415.00   Impact:    2.00   Region: CAD   Event: Housing Starts
Date: 1120209.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: Manufacturing Production m/m
Date: 1120209.00   Time: 1030.00   Impact:    2.00   Region: GBP   Event: Trade Balance
Date: 1120209.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: Industrial Production m/m
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Asset Purchase Facility
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Official Bank Rate
Date: 1120209.00   Time: 1345.00   Impact:    3.00   Region: EUR   Event: Minimum Bid Rate
Date: 1120209.00   Time: 1430.00   Impact:    2.00   Region: CAD   Event: NHPI m/m
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: EUR   Event: ECB Press Conference
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Unemployment Claims
Date: 1120210.00   Time:  915.00   Impact:    3.00   Region: CHF   Event: CPI m/m
Date: 1120210.00   Time: 1000.00   Impact:    1.00   Region: EUR   Event: Italian Industrial Production m/m
Date: 1120210.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: PPI Input m/m
Date: 1120210.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: PPI Output m/m
Date: 1120210.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Trade Balance
Date: 1120210.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Trade Balance
```

Removing multiple regions from an Economic Events list

With the use of EE.RemoveRegions function, multiple regions can be removed in one go. For example, to remove the CHF, EUR, JPY and USD regions from an economic events list:

```
Variables:
      x(0), myList("anExampleList");

// Get the economic events for the next 3 days
value1 = EE.GetNextDaysEvents(myList, 3);

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;

Print(newline, NewLine, "Removing regions EUR, USD, CHF, JPY....");

value1 = EE.RemoveRegions(myList, "EUR,USD,CHF,JPY");

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;
```

Which returns:

```
Date: 1120206.00   Time:  900.00   Impact:    3.00   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact:    2.00   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Factory Orders m/m
Date: 1120206.00   Time: 1600.00   Impact:    3.00   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact:    1.00   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact:    2.00   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: RBA Rate Statement
Date: 1120207.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Leading Indicators
Date: 1120207.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Trade Balance
Date: 1120207.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Industrial Production m/m
Date: 1120207.00   Time: 1405.00   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 1120207.00   Time: 1600.00   Impact:    3.00   Region: USD   Event: Fed Chairman Bernanke Testifies
Date: 1120207.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: IBD/TIPP Economic Optimism
Date: 1120207.00   Time: 2100.00   Impact:    1.00   Region: USD   Event: Consumer Credit m/m
Date: 1120207.00   Time:   30.00   Impact:    2.00   Region: AUD   Event: Westpac Consumer Sentiment
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Bank Lending y/y
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Current Account
Date: 1120207.00   Time:  101.00   Impact:    1.00   Region: GBP   Event: BRC Shop Price Index y/y
Date: 1120208.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Economy Watchers Sentiment
Date: 1120208.00   Time:  745.00   Impact:    1.00   Region: CHF   Event: Unemployment Rate
Date: 1120208.00   Time:  800.00   Impact:    1.00   Region: EUR   Event: German Trade Balance
Date: 1120208.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Gov Budget Balance
Date: 1120208.00   Time: 1415.00   Impact:    2.00   Region: CAD   Event: Housing Starts
Date: 1120208.00   Time: 1630.00   Impact:    2.00   Region: USD   Event: Crude Oil Inventories
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Employment Change q/q
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Unemployment Rate
Date: 1120208.00   Time:   50.00   Impact:    2.00   Region: JPY   Event: Core Machinery Orders m/m
Date: 1120208.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: M2 Money Stock y/y
Date: 1120208.00   Time:  230.00   Impact:    3.00   Region: CNY   Event: CPI y/y
Date: 1120208.00   Time:  230.00   Impact:    2.00   Region: CNY   Event: PPI y/y
Date: 1120209.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Household Confidence
Date: 1120209.00   Time:  700.00   Impact:    1.00   Region: JPY   Event: Prelim Machine Tool Orders y/y
Date: 1120209.00   Time:  745.00   Impact:    2.00   Region: CHF   Event: SECO Consumer Climate
Date: 1120209.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: Manufacturing Production m/m
Date: 1120209.00   Time: 1030.00   Impact:    2.00   Region: GBP   Event: Trade Balance
Date: 1120209.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: Industrial Production m/m
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Asset Purchase Facility
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Official Bank Rate
Date: 1120209.00   Time: 1345.00   Impact:    3.00   Region: EUR   Event: Minimum Bid Rate
Date: 1120209.00   Time: 1430.00   Impact:    2.00   Region: CAD   Event: NHPI m/m
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: EUR   Event: ECB Press Conference
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Unemployment Claims
Date: 1120209.00   Time: 1600.00   Impact:    2.00   Region: GBP   Event: NIESR GDP Estimate
Date: 1120209.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: Wholesale Inventories m/m
Date: 1120209.00   Time: 1630.00   Impact:    1.00   Region: USD   Event: Natural Gas Storage
Date: 1120209.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: CGPI y/y
Date: 1120209.00   Time:  130.00   Impact:    3.00   Region: AUD   Event: RBA Monetary Policy Statement


Removing regions EUR, USD, CHF, JPY....
Date: 1120206.00   Time: 1600.00   Impact:    3.00   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact:    1.00   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact:    2.00   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: RBA Rate Statement
Date: 1120207.00   Time: 1405.00   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 1120207.00   Time:   30.00   Impact:    2.00   Region: AUD   Event: Westpac Consumer Sentiment
Date: 1120207.00   Time:  101.00   Impact:    1.00   Region: GBP   Event: BRC Shop Price Index y/y
```

```
Date: 1120208.00   Time: 1415.00   Impact:    2.00   Region: CAD   Event: Housing Starts
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Employment Change q/q
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Unemployment Rate
Date: 1120208.00   Time:  230.00   Impact:    3.00   Region: CNY   Event: CPI y/y
Date: 1120208.00   Time:  230.00   Impact:    2.00   Region: CNY   Event: PPI y/y
Date: 1120209.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: Manufacturing Production m/m
Date: 1120209.00   Time: 1030.00   Impact:    2.00   Region: GBP   Event: Trade Balance
Date: 1120209.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: Industrial Production m/m
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Asset Purchase Facility
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Official Bank Rate
Date: 1120209.00   Time: 1430.00   Impact:    2.00   Region: CAD   Event: NHPI m/m
Date: 1120209.00   Time: 1600.00   Impact:    2.00   Region: GBP   Event: NIESR GDP Estimate
Date: 1120209.00   Time:  130.00   Impact:    3.00   Region: AUD   Event: RBA Monetary Policy Statement
```

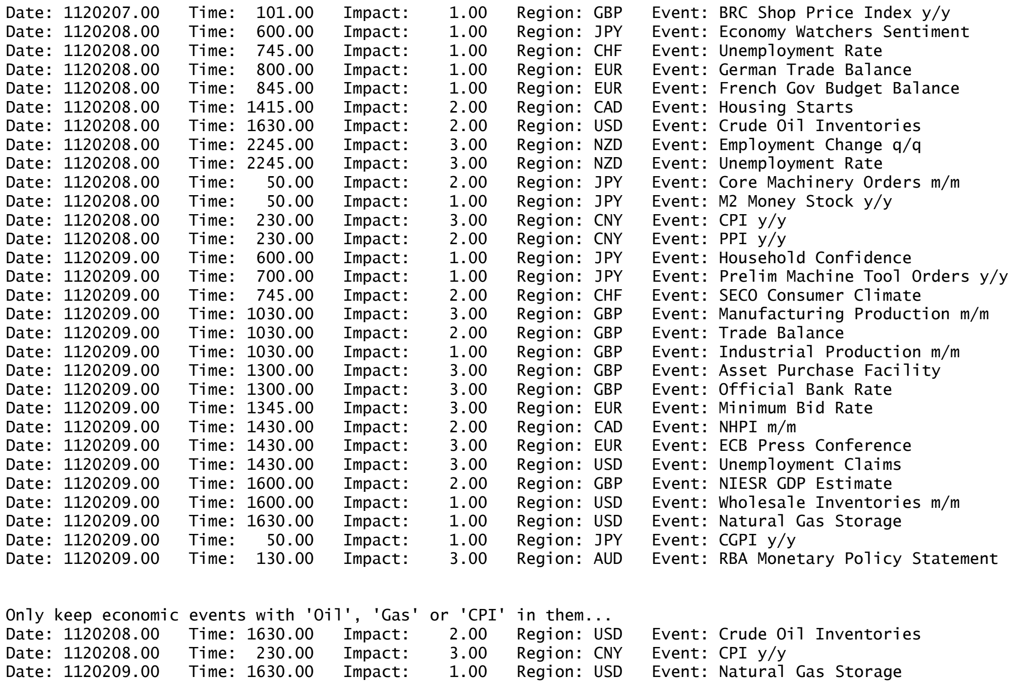
Only keep certain economic events on the Economic Events list

With the EE.KeepEvents function it's possible to only keep certain by the user specified events on the list. For example, an oil trader might want to use something like the following example.

```
Variables:
      x(0), myList("anExampleList");

// Get the economic events for the next 3 days
value1 = EE.GetNextDaysEvents(myList, 3);

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;

Print(newline, NewLine, "Only keep economic events with 'Oil', 'Gas' or 'CPI'
in them...");

value1 = EE.KeepEvents(myList, "Oil,Gas,CPI");

for x = 1 to EE.GetMaxIndex(myList) begin

      Print(
            "Date: ", EE.DateList(myList, x),
            "   Time: ", EE.TimeList(myList, x),
            "   Impact: ", EE.ImpactList(myList, x),
            "   Region: ", EE.RegionList(myList, x),
            "   Event: ", EE.EventList(myList, x)
            );
end;
```

Which gives the following output:

```
Date: 1120206.00   Time:  900.00   Impact:    3.00   Region: CHF   Event: Foreign Currency Reserves
Date: 1120206.00   Time: 1030.00   Impact:    2.00   Region: EUR   Event: Sentix Investor Confidence
Date: 1120206.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Factory Orders m/m
Date: 1120206.00   Time: 1600.00   Impact:    3.00   Region: CAD   Event: Ivey PMI
Date: 1120206.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Labor Cost Index q/q
Date: 1120206.00   Time: 2330.00   Impact:    1.00   Region: AUD   Event: AIG Construction Index
Date: 1120206.00   Time:  101.00   Impact:    2.00   Region: GBP   Event: BRC Retail Sales Monitor y/y
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: Cash Rate
Date: 1120206.00   Time:  430.00   Impact:    3.00   Region: AUD   Event: RBA Rate Statement
Date: 1120207.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Leading Indicators
Date: 1120207.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Trade Balance
Date: 1120207.00   Time: 1200.00   Impact:    2.00   Region: EUR   Event: German Industrial Production m/m
Date: 1120207.00   Time: 1405.00   Impact:    1.00   Region: CAD   Event: Gov Council Member Macklem Speaks
Date: 1120207.00   Time: 1430.00   Impact:    3.00   Region: CAD   Event: Building Permits m/m
Date: 1120207.00   Time: 1600.00   Impact:    3.00   Region: USD   Event: Fed Chairman Bernanke Testifies
Date: 1120207.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: IBD/TIPP Economic Optimism
Date: 1120207.00   Time: 2100.00   Impact:    1.00   Region: USD   Event: Consumer Credit m/m
Date: 1120207.00   Time:   30.00   Impact:    2.00   Region: AUD   Event: Westpac Consumer Sentiment
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Bank Lending y/y
Date: 1120207.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: Current Account
```



```
Date: 1120207.00   Time:  101.00   Impact:    1.00   Region: GBP   Event: BRC Shop Price Index y/y
Date: 1120208.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Economy Watchers Sentiment
Date: 1120208.00   Time:  745.00   Impact:    1.00   Region: CHF   Event: Unemployment Rate
Date: 1120208.00   Time:  800.00   Impact:    1.00   Region: EUR   Event: German Trade Balance
Date: 1120208.00   Time:  845.00   Impact:    1.00   Region: EUR   Event: French Gov Budget Balance
Date: 1120208.00   Time: 1415.00   Impact:    2.00   Region: CAD   Event: Housing Starts
Date: 1120208.00   Time: 1630.00   Impact:    2.00   Region: USD   Event: Crude Oil Inventories
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Employment Change q/q
Date: 1120208.00   Time: 2245.00   Impact:    3.00   Region: NZD   Event: Unemployment Rate
Date: 1120208.00   Time:   50.00   Impact:    2.00   Region: JPY   Event: Core Machinery Orders m/m
Date: 1120208.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: M2 Money Stock y/y
Date: 1120208.00   Time:  230.00   Impact:    3.00   Region: CNY   Event: CPI y/y
Date: 1120208.00   Time:  230.00   Impact:    2.00   Region: CNY   Event: PPI y/y
Date: 1120209.00   Time:  600.00   Impact:    1.00   Region: JPY   Event: Household Confidence
Date: 1120209.00   Time:  700.00   Impact:    1.00   Region: JPY   Event: Prelim Machine Tool Orders y/y
Date: 1120209.00   Time:  745.00   Impact:    2.00   Region: CHF   Event: SECO Consumer Climate
Date: 1120209.00   Time: 1030.00   Impact:    3.00   Region: GBP   Event: Manufacturing Production m/m
Date: 1120209.00   Time: 1030.00   Impact:    2.00   Region: GBP   Event: Trade Balance
Date: 1120209.00   Time: 1030.00   Impact:    1.00   Region: GBP   Event: Industrial Production m/m
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Asset Purchase Facility
Date: 1120209.00   Time: 1300.00   Impact:    3.00   Region: GBP   Event: Official Bank Rate
Date: 1120209.00   Time: 1345.00   Impact:    3.00   Region: EUR   Event: Minimum Bid Rate
Date: 1120209.00   Time: 1430.00   Impact:    2.00   Region: CAD   Event: NHPI m/m
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: EUR   Event: ECB Press Conference
Date: 1120209.00   Time: 1430.00   Impact:    3.00   Region: USD   Event: Unemployment Claims
Date: 1120209.00   Time: 1600.00   Impact:    2.00   Region: GBP   Event: NIESR GDP Estimate
Date: 1120209.00   Time: 1600.00   Impact:    1.00   Region: USD   Event: Wholesale Inventories m/m
Date: 1120209.00   Time: 1630.00   Impact:    1.00   Region: USD   Event: Natural Gas Storage
Date: 1120209.00   Time:   50.00   Impact:    1.00   Region: JPY   Event: CGPI y/y
Date: 1120209.00   Time:  130.00   Impact:    3.00   Region: AUD   Event: RBA Monetary Policy Statement


Only keep economic events with 'Oil', 'Gas' or 'CPI' in them...
Date: 1120208.00   Time: 1630.00   Impact:    2.00   Region: USD   Event: Crude Oil Inventories
Date: 1120208.00   Time:  230.00   Impact:    3.00   Region: CNY   Event: CPI y/y
Date: 1120209.00   Time: 1630.00   Impact:    1.00   Region: USD   Event: Natural Gas Storage
```

Store a custom Economic Events list in a text file and reload this custom list

Storing an reloading an Economic Events list is easily done with the EE.StoreListInFile and EE.LoadStoredList functions:

```
// Store the "myListName" in the "myListName.txt" file
value1 = EE.StoreListInFile("myListName", "myListName.txt");

// Load the stored list from the "myListName.txt" and save
//           this into the "mySecondList" list.
value1 = EE.LoadStoredList("mySecondList", "myListName.txt");

// Print the "mySecondList" to the log to verify
value1 = EE.PrintToOutputLog("mySecondList");
```

Which prints the following to the PowerLanguage Editor Output Log:

```
Date:       Time:       Impact:  Region: Economic Event:
09-02-12    14:30       3.00     EUR     ECB Press Conference
09-02-12    14:30       3.00     USD     Unemployment Claims
09-02-12    16:00       2.00     GBP     NIESR GDP Estimate
09-02-12    16:00       1.00     USD     Wholesale Inventories m/m
09-02-12    16:30       1.00     USD     Natural Gas Storage
10-02-12    08:00       1.00     EUR     German Final CPI m/m
10-02-12    08:45       2.00     EUR     French Industrial Production m/m
10-02-12    09:15       3.00     CHF     CPI m/m
10-02-12    10:00       1.00     EUR     Italian Industrial Production m/m
10-02-12    10:30       3.00     GBP     PPI Input m/m
10-02-12    10:30       1.00     GBP     PPI Output m/m
10-02-12    14:30       3.00     USD     Trade Balance
10-02-12    15:55       3.00     USD     Prelim UoM Consumer Sentiment
10-02-12    15:55       1.00     USD     Prelim UoM Inflation Expectations
10-02-12    18:30       3.00     USD     Fed Chairman Bernanke Speaks
10-02-12    18:50       2.00     USD     FOMC Member Pianalto Speaks
10-02-12    20:00       2.00     USD     Federal Budget Balance
```

Other examples can be found in the indicators that use the Economic Events Collection, like the ‘Economic Events – Alert Popup’ indicator.

Sources of error messages

ELCollection error: “List index is out of bounds.”

This error, from the ELCollection, can have one of the following causes:

- The code references an economic events list that wasn’t created yet (or wasn’t created successfully);
- The code removed an economic events list with EE.RemoveAll and then tried to recall it;
- The code referenced an IndexNumber beyond the length of the economic events list (for example, the list is 10 economic events long, but the code tried to reference up to 12 items). Use the EE.GetMaxIndex function to get the maximum length of the list.

ELCollection error: “Attempted to add a collection to more than one collection.”

This error from the ELCollection is caused by trying to create another economic events list with the same name as an already existing one.

Support

If you come across any errors or have any suggestions for improvements, we would be thankful if you took the time and effort to share these, so that the Economic Events Collection for MultiCharts can be further improved.

Also, if you run into specific difficulties or have a request for a specific function, go this BMT thread or this MC thread.

Disclaimer

The MultiCharts functions and Java program where created to the best of our knowledge and experience. However, that does not imply it works flawlessly now and/or will continue in the future.

It was our responsibility to test and document the features extensively - now it’s your responsibility to test your use of them extensively in a demo environment.

The MultiCharts functions and Java program are provided “as is”. The authors and/or contributors of the Economic Events Collection do not provide any warranty whatsoever, whether express, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the Economic Events Collection will be error-free.

In no respect shall the authors and/or contributors to the Economic Events Collection incur any liability for any damages, including, but limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or any way connected to the use of the item, whether or not based upon warranty, contract, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the Economic

Events Collection, or any comments and/or suggestions provided by the authors and/or contributors of the Economic Events Collection.

Changelog

v. 1.14 14-2-2012

Changes made:

- Simplified comments in the EE.UserSettings function,
- Updated manual and 'installation' instructions.
- And the *Economic Events Collection for MultiCharts* went out of beta and was released on BMT and MC forum.

v. 1.13b 11-2-2012

Changes made:

- Made a small change to the print out of the EE.PrintToOutputLog function by removing the redundant "Header" word in the output,
- Fixed an important bug in the EE.RemoveRegions function that caused a "list index is out of bounds"-error message on certain economic events list.

v. 1.12b 10-2-2012

Changes made:

- Fixed errors in the example Alert indicator, renamed it to "Economic Events – Alert Popup", simplified & cleaned up the code, and added three sound files.
- Updated manual, rewrote some descriptions, made others easier to understand, and added a standard disclaimer to prevent trouble.

v. 1.1b 9-2-2012

The following functions were added to the Economic Events Collection:

- EE.PrintToOutputLog; to make printing and checking an Economic Events list easier.
- EE.GetHistoricalEvents; to load a list with historical economic events.
- EE.StrToDate; to make the input in the EE.GetHistoricalEvents function easier.
- EE.StoreListInFile; to store a manually created list in a text file for use in the future.
- EE.LoadStoredList; to load a saved manually created list from a text file.

Changes to existing functions:

- Added a FileDirectory StringRef variable to EE.UserSettings; after that the existing functions EE.GetFutureEvents and EE.GetNextDaysEvents had to be changed too.

Miscellaneous changes:

- Updated manual; added description of new functions and changed some descriptions of existing.

v. 1.00b 6-2-2012

Beta testing for the Economic Events functions started.

v. 0.1a 3-2-2012

An idea was born. :)