

# INSTALLATION INSTRUCTIONS

## For Installing Jurik Tools for Use with AmiBroker

1. Before you begin the installation process, shut down Amibroker, if it is already running.
2. You must have a licensed copy of Jurik tools in DL format installed on your computer for AmiBroker. If you had already installed Jurik Tools for some other platform, re-run the installer file JRS\_DLL.EXE and select "AmiBroker" as the target platform. For information about which Jurik tools run in AmiBroker, go to <http://www.jurikres.com/catalog/compatib.htm#ami>.
3. Note that you must be using AmiBroker, version 4.5 or later. You can get the latest version and update of AmiBroker from <http://www.amibroker.com> . The AmiBroker installation file will have the name "AmiBroker\_\_\_\_.exe" where "\_\_\_\_" is replaced by a version number, such as 450. The latest version of AmiBroker will have the most recent file date.
4. Download the plug-in file JRPLUG2.EXE from <http://www.amibroker.com/bin/JRPlug2.exe>. Run the file. This will add to AmiBroker additional software needed to run Jurik tools.
5. After installing both the DL version of Jurik Tools and JRPLUG2.EXE, restart Amibroker. Choose the Tools/Plugins menu. You should see "Jurik Research Tools" in the list of plugins. If you do not see "Jurik Research Tools" contact Jurik Research Software technical support at [nfs@nfsmith.net](mailto:nfs@nfsmith.net). Select the plug-in item and press the LOAD button, then press CLOSE.

Please note that availability of JMA, CFB, VEL, RSX, and DMX depends on the license you purchased from Jurik Research. If you attempt to use the Jurik tool not licensed for your system, an error message will appear.

---

JRPLUG2.EXE creates wrapper file JRPlug.dll, which was written by Tomasz Janeczko of AmiBroker.com. It is copyrighted freeware. © 2002 AmiBroker.com. All rights reserved. You may distribute and use it freely provided all original documentation is supplied.

CFB, JMA, RSX, VEL, DMX are trademarks of Jurik Research. © 2000 Jurik Research ; [www.jurikres.com](http://www.jurikres.com)

# CREATING INDICATORS with JURIK TOOLS

To plot Jurik Research indicators, you will need to write some code using AFL (AmiBroker Formula Language). For those of you who are not familiar with writing code, don't worry. We will describe exactly what you need to do, step by step in order to plot our tools as indicators.

There are 8 Jurik Tools functions that can be used in Amibroker's Formula Editor, Automatic Analysis (for backtesting, optimization, exploration and scans) and Chart Commentary windows. Some of them may not be available to you if you did not acquire a license to all of the available Jurik Tools.

## Coding Example

Here are detailed instructions for plotting the JMA. One example will be enough for you to get the idea how to do the others. After this example there is a Reference List of all available Jurik functions.

1. After starting Amibroker, click **Analysis / Formula Editor**. Then select **File / New**.
2. Create a name for your indicator by typing it in the name edit box (where it says "Unnamed"). Type **MyJMA**, for instance.
3. Copy and paste the following text into the Formula Editor:

```
setbarsrequired( 200, 0 );  
JMAarray = JurikJMA(Close, 7, 50);  
Plot( JMAarray, "JMA", colorRed );
```

4. Select **File / Save**, and close the Formula Editor.
5. In the Workspace window on the left, click the **Charts** tab.
6. Then click the **Custom** folder, and your indicator will be there. Drag and drop it onto a price chart.

Note that other types of indicators are not scaled the same as price, so new window panes should be created for them. For example, after creating and saving your own indicator based on Jurik's RSX, select the menu command **Window / New Linked** to create a new pane. Then drag your indicator from the **Custom** folder into the new pane.

For in-depth coverage on features and performance of Jurik Research Tools, consult original documentation created by the Jurik Tools installer.

# FUNCTION REFERENCE GUIDE

**All Jurik functions return a time series array.**

## ***JurikCFB***

SYNTAX	JurikCFB( ARRAY, smooth, spansize )
FUNCTION	Composite Fractal Behavior Index. (Jurik's version of ADX) Smooth may be any positive <u>integer</u> . Typical value = 8. Valid spansize values { 24, 48, 96, 192 }
EXAMPLE	<pre>setbarsrequired( 200, 0 ); CFBarray = JurikCFB( Close, 8, 48 ); Plot( CFBarray, "CFB", colorRed );</pre>

---

## ***JurikDMX***

SYNTAX	JurikDMX( periods )
FUNCTION	Bipolar directional movement index. (Jurik's version of DMI) Periods may be any positive decimal value. Typical value = 14.
EXAMPLE	<pre>setbarsrequired( 200, 0 ); DMXarray = JurikDMX( 14 ); Plot( DMXarray, "DMX", colorRed );</pre>

---

## ***JurikDMXPlus***

SYNTAX	JurikDMXPlus( periods )
FUNCTION	Calculates DM+ component of Jurik's DMX. Periods may be any positive decimal value. Typical value = 14.
EXAMPLE	<pre>setbarsrequired( 200, 0 ); DMXplusArray = JurikDMXPlus( 14 ); Plot( DMXplusArray, "DM+", colorRed );</pre>

---

### ***JurikDMXMinus***

SYNTAX            JurikDMXMinus( periods )

FUNCTION          Calculates DM- component of Jurik's DMX.  
Periods may be any positive decimal value. Typical value = 14.

EXAMPLE          `setbarsrequired( 200, 0 );`  
`DMXminusArray = JurikDMXMinus( 14 );`  
`Plot( DMXminusArray, "DM-", colorRed );`

---

### ***JurikRSX***

SYNTAX            JurikRSX( ARRAY, length )

FUNCTION          Relative Trend Strength Index  
Length may be any number greater than 2. Typical value = 12.

EXAMPLE          The following formula compares the standard RSI(14) indicator and Jurik's  
smoother RSX(14):

```
setbarsrequired( 200, 0 );
RSIarray = RSI( 14 );
RSXarray = JurikRSX( Close, 14 );
Plot( RSIarray, "RSI", colorRed );
plot( RSXarray, "RSX", colorGreen );
```

---

### ***JurikVEL***

SYNTAX            JurikVEL( ARRAY, depth )

FUNCTION          Zero-Lag Velocity  
Depth may be any integer greater than 1. Typical value = 12.

EXAMPLE          `setbarsrequired( 200, 0 );`  
`VELarray = JurikVEL( Close, 14 );`  
`Plot( VELarray, "VEL", colorRed );`

## ***JurikJMAV***

**SYNTAX**            JurikJMAV( ARRAY, speedArray, phaseArray)

**FUNCTION**         Jurik's Moving Average. this version permits both speed and phase to be uniquely specified for each bar in the time series. Speed and phase values are located in the speedArray and phaseArray respectively or as formulas.

Valid range for each phase value is from -100 to +100  
Valid range for each speed value is any number greater than 1.

**EXAMPLE**            The following formula plots JMA, whose length is controlled by ADX:

```
setbarsrequired( 200, 0);  
JMAarray = JurikJMAV( Close, 30+ADX()/5, 0);  
Plot( JMAarray, "vJMA", colorRed );
```

## ***JurikJMA***

**SYNTAX**            JurikJMA( ARRAY, speed, phase )

**FUNCTION**         Calculates the Jurik's Moving Average  
Valid range for phase is from -100 to +100  
Valid range for speed is any decimal number greater than 1.

**EXAMPLE**

```
setbarsrequired( 200, 0);  
JMAarray = JurikJMA( Close, 7, 50);  
Plot( JMAarray, "JMA", colorRed );
```

The following formula plots a MACD based on JMA:

```
setbarsrequired( 200, 0);  
JMA_MACD = JurikJMA(Close, 7, 50) - JurikJMA(Close, 35, 0);  
Plot( JMA_MACD, "JMA MACD", colorRed );
```