

MiscMMAProcessor

Inherits From: NSObject
Declared In: MiscMMAProcessor.h

Class Description

An instance of MiscMMAProcessor manages a Mathlink connection to Mathematica.
(Only Mathematica 2.2 or higher is supported).

This class does not attempt to encapsulate and hide the complex functionality Mathlink provides. Hiding the Mathlink connection from the user of this class would have meant implementing a cover for the whole Mathlink library calls (making Mathlink all over again :-)) or losing some of the great capabilities Mathlink provides. This class gives "building blocks" of code but the user has to know what he's doing. A good starting point is the Mathlink documentation available at <http://www.wolfram.com>.

But don't despair. You can use this class in two modes:

- a highlevel, simple mode in which you lose some of the Mathlink functionality but don't have to call Mathlink functions yourself
- a lowlevel, expert mode with the whole range of Mathlink functions available to you.

In the highlevel mode all you have to do is create a MiscMMAProcessor instance and send it pairs of **enterExpression:** and **readReturn** messages. You must be careful to always send these messages in pairs (and not interwoven) otherwise you can hang the connection by reading something which does not exist. The

next version of this class will avoid this problem with multithreading and timers. The downside is that you only get results from Mathematica in text form as NSStrings (besides of the Mathematica graphics returned as Postscript data in NSData objects suitable for MiscMMAView views). If you can live with this restriction then the highlevel mode is for you.

In the lowlevel mode you create a MiscMMAProcessor and use its "code blocks" to manage the Mathlink connection. The MiscMMAProcessor class has methods for opening and closing a Mathlink connection, for reading packets, discarding remaining packets and error handling. In this mode the MiscMMAProcessor class is just a convenience. You could do without it but it's nice and handy to have it. Again a word of caution here: if you don't use the "code blocks" properly you can hang the connection by reading something which does not exist. The lowlevel use is best demonstrated with a simple example which lets Mathematica calculate the inverse of a matrix. Suppose we have a class MatrixInverter which implements the method invertMatrix:size:in:.

MMADoc.rtf –Example MatrixInverter

An MiscMMAProcessor instance does not know how to read an arbitrary RETURNPKT packet. When an object is passed in **readReturnWithReader:** this object gets the chance to read the packet, when **nil** is passed the RETURNPKT packet is discarded. Objects passed to this method must implement the **MiscMMAReturnPacketReading** protocol consisting of the method **readReturnPacketWithProcessor:**.

Instance Variables

No public or protected instance variables declared in this class.

Method Types

Opening, accessing and closing Mathlink connection
- openLink

	<ul style="list-style-type: none"> - closeLink - link
Packet management	<ul style="list-style-type: none"> - nextPacketWithTag: - discardRemainingPackets
Error and state handling	<ul style="list-style-type: none"> - error - errorMessage - stateMessage
Executing Mathematica expressions	<ul style="list-style-type: none"> - enterExpression:
Reading Mathematica results	<ul style="list-style-type: none"> - readReturnWithReader: - readReturn

Instance Methods

closeLink

- (void)**closeLink**

Closes an existing link to Mathematica. If no link exists does nothing.

discardRemainingPackets

- (BOOL)**discardRemainingPackets**

Discards all remaining packets from the link.

enterExpression:

- (BOOL)**enterExpression:(NSString *)***expr*

Wraps the expression *expr* in the Mathematica function Enter[] and sends it over the link. If succesful returns YES otherwise NO.

error

- (BOOL)**error**

Returns whether the receiver has encountered an error in the link.

errorMessage

- (NSString *)**errorMessage**

Returns the error message or empty string if no error occurred.

link

- (MLINK)**link**

Returns the link managed by the receiver.

nextPacketWithTag:

- (BOOL)**nextPacketWithTag:(int)***tag*

Reads and discards packets from the link until a packet with *tag tag* is encountered.

openLink

- (BOOL)**openLink**

Receiver opens a link to Mathematica. If a link already exists does nothing. Returns YES upon success and NO upon error. Find out the error with **errorMessage**.

readReturn

- (NSDictionary *)**readReturn**

Reads result packets from link and returns a NSDictionary with all results. The dictionary contains an object for the key "Plot" if a Mathematica graphic was part of the result. The dictionary contains an object for the key "Text" if a Mathematica text result was part of the result. The dictionary contains an object for the key "Syntax" if a Mathematica syntax error was part of the result. All other packet types are discarded.

readReturnWithReader:

- (NSDictionary *)**readReturnWithReader:**(id <MiscMMAReturnPacketReading>) *reader*

Reads result packets from link and returns a NSDictionary with all results. The dictionary contains an object for the key "Plot" if a Mathematica graphic was part of the result. The dictionary contains an object for the key "Text" if a Mathematica text result was part of the result. The dictionary contains an object for the key "Syntax" if a Mathematica syntax error was part of the result. RETURNPKT are handled by *reader* and all other packet types are discarded.

stateMessage

- (NSString *)**stateMessage**

Returns the state message for the state in which the receiver is.