MAMA C API Reference Manual

2.2.2.1

Generated by Doxygen 1.4.7

Thu Feb 7 17:04:46 2013

Contents

Chapter 1

Middleware Agnostic Messaging API (MAMA) C API

Middleware Agnostic Messaging API. The Middleware Agnostic Messaging (MAMA) API provides an abstraction layer over various messaging middleware platforms. In particular, MAMA adds market data semantics to messaging platforms that would otherwise be too limited for use as a market data distribution middleware. Features that MAMA adds to any messaging middleware are:

- Subscription management (initial values, throttling).
- Entitlements/permissioning
- Data quality.

The goal of MAMA is to maximize application portability. Once an application has been ported to MAMA, it should never have to be ported to another market data messaging API again. Many firms have invested time in building and maintaining their own abstraction APIs - and they should be commended for that. We hope that even those firms will see the benefit in migrating to MAMA and thereby reducing costs further and, as more third party firms migrate applications to MAMA (and MAMDA, see below), benefit even more from this compatibility.

MAMA currently supports the following middleware platforms:

- Wombat TCP Middleware (see http://wombatfs.com)
- 29West Latency Busters Messaging (see http://29west.com)
- Tibco TIB/RV versions 6 and 7 (see http://tibco.com)

A higher level market data API is also available: the Middleware Agnostic Market Data API (MAMDA). While MAMA provides a field-based abstraction to market

data, MAMDA provides smart, specialized data types to deal with specific market data events, such as trades, quotes, order books, etc. MAMDA is particularly suitable for applications such as program trading and tick capture systems, where context is important. MAMA is more suited to applications that don't care about the meaning of fields, such as a simple, field-based market data display application.

Chapter 2

MAMA C API Data Structure Index

2.1 MAMA C API Data Structures

Here are the data structures with brief descriptions:

Chapter 3

MAMA C API File Index

3.1 MAMA C API File List

Here is a list of all files with brief descriptions:

Chapter 4

MAMA C API Data Structure Documentation

4.1 mamaCommand Struct Reference

```
#include <clientmanage.h>
```

Data Fields

- void * mHandle
- cmCommandDtor mDtor

4.1.1 Field Documentation

4.1.1.1 void* mamaCommand::mHandle

4.1.1.2 cmCommandDtor mamaCommand::mDtor

The documentation for this struct was generated from the following file:

- clientmanage.h

4.2 mamaDictionaryCallbackSet Struct Reference

A structure containing the callbacks for dictionary creation.

```
#include <dictionary.h>
```

Data Fields

- mamaDictionary_completeCallback onComplete
- mamaDictionary_timeoutCallback onTimeout
- mamaDictionary_errorCallback onError

4.2.1 Detailed Description

A structure containing the callbacks for dictionary creation.

4.2.2 Field Documentation

4.2.2.1 mamaDictionary_completeCallback mamaDictionaryCallbackSet::onComplete

4.2.2.2 mamaDictionary_timeoutCallback mamaDictionaryCallbackSet::onTimeout

4.2.2.3 mamaDictionary_errorCallback mamaDictionaryCallbackSet::onError

The documentation for this struct was generated from the following file:

- dictionary.h

4.3 mamaDQPublisherManagerCallbacks_ Struct Reference

```
#include <dqpublishermanager.h>
```

Data Fields

- mamaDQPublisherCreateCB onCreate
- mamaDQPublisherNewRequestCB onNewRequest
- mamaDQPublisherRequestCB onRequest
- mamaDQPublisherRefreshCB onRefresh
- mamaDQPublisherErrorCB onError
- mamaDQPublisherMsgCB onMsg

4.3.1 Field Documentation

4.3.1.1 mamaDQPublisherCreateCB mamaDQPublisherManagerCallbacks_::onCreate

4.3.1.2 mamaDQPublisherNewRequestCB mamaDQPublisherManagerCallbacks_::onNewRequest

4.3.1.3 mamaDQPublisherRequestCB mamaDQPublisherManagerCallbacks_::onRequest

4.3.1.4 mamaDQPublisherRefreshCB mamaDQPublisherManagerCallbacks_::onRefresh

4.3.1.5 mamaDQPublisherErrorCB mamaDQPublisherManagerCallbacks_::onError

4.3.1.6 mamaDQPublisherMsgCB mamaDQPublisherManagerCallbacks_::onMsg

The documentation for this struct was generated from the following file:

- dqpublishermanager.h

4.4 mamaMsgCallbacks_ Struct Reference

A convenience structure for passing the callbacks to the subscription factory methods.

`#include <subscription.h>`

Data Fields

- wombat_subscriptionCreateCB onCreate
- wombat_subscriptionErrorCB onError
- wombat_subscriptionOnMsgCB onMsg
- wombat_subscriptionQualityCB onQuality
- wombat_subscriptionGapCB onGap
- wombat_subscriptionRecapCB onRecapRequest
- wombat_subscriptionDestroyCB onDestroy

4.4.1 Detailed Description

A convenience structure for passing the callbacks to the subscription factory methods.

4.4.2 Field Documentation

4.4.2.1 wombat_subscriptionCreateCB mamaMsgCallbacks_::onCreate

4.4.2.2 wombat_subscriptionErrorCB mamaMsgCallbacks_::onError

4.4.2.3 wombat_subscriptionOnMsgCB mamaMsgCallbacks_::onMsg

4.4.2.4 wombat_subscriptionQualityCB mamaMsgCallbacks_::onQuality

4.4.2.5 wombat_subscriptionGapCB mamaMsgCallbacks_::onGap

4.4.2.6 wombat_subscriptionRecapCB mamaMsgCallbacks_::onRecapRequest

4.4.2.7 wombat_subscriptionDestroyCB mamaMsgCallbacks_::onDestroy

The documentation for this struct was generated from the following file:

- subscription.h

4.5 mamaPublishTopic_ Struct Reference

```
#include <dqpublishermanager.h>
```

Data Fields

- const char * symbol
- mamaDQPublisher pub
- void * cache

4.5.1 Field Documentation

4.5.1.1 const char* mamaPublishTopic_::symbol

4.5.1.2 mamaDQPublisher mamaPublishTopic_::pub

4.5.1.3 void* mamaPublishTopic_::cache

The documentation for this struct was generated from the following file:

- dqpublishermanager.h

4.6 mamaQueueMonitorCallbacks_ Struct Reference

callbacks which may be invoked in response to certain conditions on the specified queue being met.

```
#include <queue.h>
```

Data Fields

- mamaQueueHighWatermarkExceededCb onQueueHighWatermarkExceeded
- mamaQueueLowWatermarkCb onQueueLowWatermark

4.6.1 Detailed Description

callbacks which may be invoked in response to certain conditions on the specified queue being met.

4.6.2 Field Documentation

4.6.2.1 mamaQueueHighWatermarkExceededCb mamaQueueMonitorCallbacks_::onQueueHighWatermarkExceeded

4.6.2.2 mamaQueueLowWatermarkCb mamaQueueMonitorCallbacks_::onQueueLowWatermark

The documentation for this struct was generated from the following file:

- queue.h

4.7 MamaReservedField_ Struct Reference

```
#include <reservedfields.h>
```

Data Fields

- char * mName
- unsigned short mFid

4.7.1 Field Documentation

4.7.1.1 char* MamaReservedField_::mName

4.7.1.2 unsigned short MamaReservedField_::mFid

The documentation for this struct was generated from the following file:

- reservedfields.h

4.8 mamaWildCardMsgCallbacks_ Struct Reference

A convenience structure for passing the callbacks to wild card subscription factory methods.

```
#include <subscription.h>
```

Data Fields

- wombat_subscriptionCreateCB onCreate
- wombat_subscriptionErrorCB onError
- wombat_subscriptionWildCardOnMsgCB onMsg
- wombat_subscriptionDestroyCB onDestroy

4.8.1 Detailed Description

A convenience structure for passing the callbacks to wild card subscription factory methods.

4.8.2 Field Documentation

4.8.2.1 wombat_subscriptionCreateCB mamaWildCardMsgCallbacks_::onCreate

4.8.2.2 wombat_subscriptionErrorCB mamaWildCardMsgCallbacks_::onError

4.8.2.3 wombat_subscriptionWildCardOnMsgCB mamaWildCardMsgCallbacks_::onMsg

4.8.2.4 wombat_subscriptionDestroyCB mamaWildCardMsgCallbacks_::onDestroy

The documentation for this struct was generated from the following file:

- subscription.h

Chapter 5

MAMA C API File Documentation

5.1 clientmanage.h File Reference

Data Structures

- struct mamaCommand

Defines

- #define MAMA_CM_TOPIC "__MAMA_CM"

  Client Management Request Get Sent HERE.

- #define MAMA_CM_PUB_TOPIC "__MAMA_PUB_CM"
- #define MAMA_SYNC_TOPICS_ID 101

  The SyncResponder puts the topics in a string array with this ID.

- #define MAMA_CM_COMMAND_ID 102

  Future: We may add other (possibly wadmin style) commands U16.

- #define MAMA_SYNC_SOURCE_ID 103

  Client only sends topics for this SOURCE.

- #define MAMA_SYNC_TOPICS_PER_MSG_ID 104

  Client responds with max TOPICS_PER_MSG U16.

- #define MAMA_SYNC_RESPONSE_DELAY_ID 105

  Clients waits random interval > 0 < RESPONSE_DELAY before initiating response.

- #define MAMA_SYNC_RESPONSE_DURATION_ID 106

  Clients send response over this duration.

- #define MAMA_SYNC_TYPES_ID 107

  The SyncResponder puts the subscription types in a string array with this ID.

Typedefs

- typedef void(∗) cmCommandDtor (void ∗handle)

  This structure allows for generic commands to be destroyed by the clientmanagere-sponder.

- typedef void(∗) mamaCommandEndCB (mamaCommand ∗command, void ∗closure)

  Commands invoke this callback when they complete so the responder can destroy them.

Enumerations

- enum mamaCmCommand { MAMA_COMMAND_SYNC = 1 }

  Commands.

5.1.1 Define Documentation

5.1.1.1 #define MAMA_CM_TOPIC "__MAMA_CM"

Client Management Request Get Sent HERE.

5.1.1.2 #define MAMA_CM_PUB_TOPIC "__MAMA_PUB_CM"

5.1.1.3 #define MAMA_SYNC_TOPICS_ID 101

The SyncResponder puts the topics in a string array with this ID.

5.1.1.4 #define MAMA_CM_COMMAND_ID 102

Future: We may add other (possibly wadmin style) commands U16.

5.1.1.5 #define MAMA_SYNC_SOURCE_ID 103

Client only sends topics for this SOURCE.

5.1.1.6 #define MAMA_SYNC_TOPICS_PER_MSG_ID 104

Client responds with max TOPICS_PER_MSG U16.

5.1.1.7 #define MAMA_SYNC_RESPONSE_DELAY_ID 105

Clients waits random interval $> 0 <$ RESPONSE_DELAY before initiating response.

5.1.1.8 #define MAMA_SYNC_RESPONSE_DURATION_ID 106

Clients send response over this duration.

5.1.1.9 #define MAMA_SYNC_TYPES_ID 107

The SyncResponder puts the subscription types in a string array with this ID.

5.1.2 Typedef Documentation

5.1.2.1 typedef void(∗) cmCommandDtor(void ∗handle)

This structure allows for generic commands to be destroyed by the clientmanagere-sponder.

It contains a handle and a pointer to a destructor function.

It is not correct for the commands to destroy themselves on completion since the responder creates them. Furthermore, there would be problems at shutdown.

Every command needs to populate this structure accordingly.

5.1.2.2 typedef void(∗) mamaCommandEndCB(mamaCommand ∗command, void ∗closure)

Commands invoke this callback when they complete so the responder can destroy them.

5.1.3 Enumeration Type Documentation

5.1.3.1 enum mamaCmCommand

Commands.

Enumerator:

MAMA_COMMAND_SYNC Sync Request.

```
{
    MAMA_COMMAND_SYNC = 1
} mamaCmCommand;
```

5.2 config.h File Reference

```
#include <wombat/wConfig.h>
```

5.3 datetime.h File Reference

```
#include <mama/config.h>
#include <mama/status.h>
#include <mama/types.h>
#include <mama/timezone.h>
#include "wombat/port.h"
#include <time.h>
```

Defines

- #define MAMA_DATE_TIME_HAS_DATE ((mamaDateTimeHints) 0x01)
- #define MAMA_DATE_TIME_HAS_TIME ((mamaDateTimeHints) 0x02)
- #define MAMA_DATE_TIME_NO_TIMEZONE ((mamaDateTimeHints) 0x04)

Typedefs

- typedef enum mamaDateTimePrecision_ mamaDateTimePrecision
- typedef enum mamaDayOfWeek_ mamaDayOfWeek
- typedef mama_u8_t mamaDateTimeHints

Enumerations

- enum mamaDateTimePrecision_ {

  MAMA_DATE_TIME_PREC_SECONDS = 0, MAMA_DATE_TIME_-PREC_DECISECONDS = 1, MAMA_DATE_TIME_PREC_CENTISECONDS = 2, MAMA_DATE_TIME_PREC_MILLISECONDS = 3,

  MAMA_DATE_TIME_PREC_MICROSECONDS = 6, MAMA_DATE_-TIME_PREC_DAYS = 10, MAMA_DATE_TIME_PREC_MINUTES = 12, MAMA_DATE_TIME_PREC_UNKNOWN = 15 }
- enum mamaDayOfWeek_ {

  Sunday = 0, Monday, Tuesday, Wednesday,

  Thursday, Friday, Saturday }

Functions

- MAMAExpDLL mama_status mamaDateTime_create (mamaDateTime *date-Time)

Create a date/time object.

- MAMAExpDLL mama_status mamaDateTime_destroy (mamaDateTime dateTime)

  Destroy a mamaDateTime object.

- MAMAExpDLL mama_status mamaDateTime_clear (mamaDateTime dateTime)

  Clear a mamaDateTime object.

- MAMAExpDLL mama_status mamaDateTime_clearDate (mamaDateTime dateTime)

  Clear the date part of a mamaDateTime object.

- MAMAExpDLL mama_status mamaDateTime_clearTime (mamaDateTime dateTime)

  Clear the time of day part of a mamaDateTime object (preserving the date).

- MAMAExpDLL mama_status mamaDateTime_copy (mamaDateTime dest, const mamaDateTime src)
- MAMAExpDLL int mamaDateTime_empty (const mamaDateTime dateTime)
- MAMAExpDLL int mamaDateTime_equal (const mamaDateTime lhs, const mamaDateTime rhs)
- MAMAExpDLL int mamaDateTime_compare (const mamaDateTime lhs, const mamaDateTime rhs)
- MAMAExpDLL mama_status mamaDateTime_setEpochTime (mamaDateTime dateTime, mama_u32_t seconds, mama_u32_t microseconds, mamaDateTimePrecision precision)

  Set the date and time as seconds and microseconds since the Epoch (UTC time zone) with an option to designate the accuracy of the time.

- MAMAExpDLL mama_status mamaDateTime_setEpochTimeF64 (mamaDateTime dateTime, mama_f64_t seconds)

  Set the date and time as seconds (plus, possibly fractional seconds) since the Epoch (UTC time zone).

- MAMAExpDLL mama_status mamaDateTime_setEpochTimeMilliseconds (mamaDateTime dateTime, mama_u64_t milliseconds)

  Set the date and time as milliseconds.

- MAMAExpDLL mama_status mamaDateTime_setEpochTimeMicroseconds (mamaDateTime dateTime, mama_u64_t milliseconds)

  Set the date and time as microseconds.

- MAMAExpDLL mama_status mamaDateTime_setWithHints (mamaDateTime dateTime, mama_u32_t seconds, mama_u32_t microseconds, mamaDateTimePrecision precision, mamaDateTimeHints hints)

  Set the date and/or time with special, optional hints to indicate whether the date/time includes date information, time information and/or whether a user of the date/time should consider it in the context of a time zone.

- MAMAExpDLL mama_status mamaDateTime_setPrecision (mamaDateTime dateTime, mamaDateTimePrecision precision)

  Set the precision hint.

- MAMAExpDLL mama_status mamaDateTime_setToNow (mamaDateTime dateTime)

  Set the date and time to the current UTC time.

- MAMAExpDLL mama_status mamaDateTime_setToMidnightToday (mamaDateTime dateTime, const mamaTimeZone tz)

  Set the dateTime object to 12am of the current date in the timezone provided (or UTC if NULL).

- MAMAExpDLL mama_status mamaDateTime_setWithPrecisionAndTz (mamaDateTime dateTime, mama_u32_t year, mama_u32_t month, mama_u32_t day, mama_u32_t hour, mama_u32_t minute, mama_u32_t second, mama_u32_t microsecond, mamaDateTimePrecision precision, const mamaTimeZone tz)

  Set the entire date and time for the MamaDateTime.

- MAMAExpDLL mama_status mamaDateTime_setTime (mamaDateTime dateTime, mama_u32_t hour, mama_u32_t minute, mama_u32_t second, mama_u32_t microsecond)

  Set the time-of-day portion of the MamaDateTime.

- MAMAExpDLL mama_status mamaDateTime_setTimeWithPrecisionAndTz (mamaDateTime dateTime, mama_u32_t hour, mama_u32_t minute, mama_u32_t second, mama_u32_t microsecond, mamaDateTimePrecision precision, const mamaTimeZone tz)

  Set the time-of-day portion of the MamaDateTime with explicit precision.

- MAMAExpDLL mama_status mamaDateTime_setDate (mamaDateTime dateTime, mama_u32_t year, mama_u32_t month, mama_u32_t day)

  Set the date portion of the MamaDateTime.

- MAMAExpDLL mama_status mamaDateTime_copyTime (mamaDateTime dest, const mamaDateTime src)

Copy the time-of-day portion of the mamaDateTime.

- MAMAExpDLL mama_status mamaDateTime_copyDate (mamaDateTime dest, const mamaDateTime src)

  Copy the date portion of the mamaDateTime.

- MAMAExpDLL mama_status mamaDateTime_hasTime (const mamaDateTime dateTime, mama_bool_t *result)

  Determine whether the time-of-day portion of the MamaDateTime is set.

- MAMAExpDLL mama_status mamaDateTime_hasDate (const mamaDateTime dateTime, mama_bool_t *result)

  Determine whether the date portion of the MamaDateTime is set.

- MAMAExpDLL mama_status mamaDateTime_addSeconds (mamaDateTime dateTime, mama_f64_t seconds)

  Add a number of seconds, including fractional seconds (may be negative).

- MAMAExpDLL mama_status mamaDateTime_addWholeSeconds (mamaDateTime dateTime, mama_i32_t seconds)

  Add a number of whole seconds (may be negative).

- MAMAExpDLL mama_status mamaDateTime_addMicroseconds (mamaDateTime dateTime, mama_i64_t microseconds)

  Add a number of microseconds (may be negative) Add a number of microseconds (may be negative and greater than 1000000 or less than -1000000).

- MAMAExpDLL mama_status mamaDateTime_setFromString (mamaDateTime dateTime, const char *str)

  Set the date and time from a string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

- MAMAExpDLL mama_status mamaDateTime_setFromStringWithTz (mamaDateTime dateTime, const char *str, const mamaTimeZone tz)

  Set the date and time from a string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

- MAMAExpDLL mama_status mamaDateTime_setFromStringBuffer (mamaDateTime dateTime, const char *str, mama_size_t strLen)

  Set the date and time from an unterminated string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

- MAMAExpDLL mama_status mamaDateTime_setFromStringBufferWithTz (mamaDateTime dateTime, const char *str, mama_size_t strLen, const mamaTimeZone tz)

Set the date and time from an unterminated string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

- MAMAExpDLL mama_status mamaDateTime_getEpochTime (const mamaDateTime dateTime, mama_u32_t *seconds, mama_u32_t *microseconds, mamaDateTimePrecision *precision)

  Get the date and time as seconds and microseconds since the Epoch (UTC time zone).

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeWithTz (const mamaDateTime dateTime, mama_u32_t *seconds, mama_u32_t *microseconds, mamaDateTimePrecision *precision, const mamaTimeZone tz)

  Get the date and time as seconds and microseconds since the Epoch in the timezone supplied.

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeMicroseconds (const mamaDateTime dateTime, mama_u64_t *microseconds)

  Get the date and time as microseconds since the Epoch, (using the UTC timezone).

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeMicrosecondsWithTz (const mamaDateTime dateTime, mama_u64_t *microseconds, const mamaTimeZone tz)

  Get the date and time as microseconds since the Epoch in the supplied time zone.

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeMilliseconds (const mamaDateTime dateTime, mama_u64_t *milliseconds)

  Get the date and time as milliseconds since the Epoch (UTC time zone).

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeMillisecondsWithTz (const mamaDateTime dateTime, mama_u64_t *milliseconds, const mamaTimeZone tz)

  Get the date and time as milliseconds since the Epoch in the timezone supplied.

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeSeconds (const mamaDateTime dateTime, mama_f64_t *seconds)

  Get the date and time as seconds since the Epoch (UTC time zone).

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeSecondsWithCheck (const mamaDateTime dateTime, mama_f64_t *seconds)

  Get the date and time as seconds since the Epoch, (using the UTC timezone).

- MAMAExpDLL mama_status mamaDateTime_getEpochTimeSecondsWithTz (const mamaDateTime dateTime, mama_f64_t *seconds, const mamaTimeZone tz)

  Get the date and time as seconds since the Epoch in the timezone supplied.

- MAMAExpDLL mama_status mamaDateTime_getWithHints (const mamaDateTime dateTime, mama_u32_t *seconds, mama_u32_t *microseconds, mamaDateTimePrecision *precision, mama_u8_t *hints)

  Get the date and/or time with special, optional hints to indicate whether the date/time includes date information, time information and/or whether a user of the date/time should consider it in the context of a time zone.

- MAMAExpDLL mama_status mamaDateTime_getStructTimeVal (const mamaDateTime dateTime, struct timeval *result)

  Get the date/time as a "struct timeval".

- MAMAExpDLL mama_status mamaDateTime_getStructTimeValWithTz (const mamaDateTime dateTime, struct timeval *result, const mamaTimeZone tz)

  Get the date/time as a "struct timeval" in the timezone supplied.

- MAMAExpDLL mama_status mamaDateTime_getStructTm (const mamaDateTime dateTime, struct tm *result)

  Get the date/time as a "struct tm".

- MAMAExpDLL mama_status mamaDateTime_getStructTmWithTz (const mamaDateTime dateTime, struct tm *result, const mamaTimeZone tz)

  Get the date/time as a "struct tm" in the timezone supplied.

- MAMAExpDLL mama_status mamaDateTime_getAsString (const mamaDateTime dateTime, char *str, mama_size_t maxLen)

  Get the date and/or time as a string.

- MAMAExpDLL mama_status mamaDateTime_getTimeAsString (const mamaDateTime dateTime, char *str, mama_size_t maxLen)

  Get the time (no date) as a string.

- MAMAExpDLL mama_status mamaDateTime_getDateAsString (const mamaDateTime dateTime, char *str, mama_size_t maxLen)

  Get the date (no time) as a string.

- MAMAExpDLL mama_status mamaDateTime_getAsFormattedString (const mamaDateTime dateTime, char *str, mama_size_t maxLen, const char *format)

  Get the date and/or time as a string using the format provided.

- MAMAExpDLL mama_status mamaDateTime_getAsFormattedStringWithTz (const mamaDateTime dateTime, char *str, mama_size_t maxLen, const char *format, const mamaTimeZone tz)

Get the date and/or time as a string using the format provided.

- MAMAExpDLL mama_status mamaDateTime_getYear (const mamaDateTime dateTime, mama_u32_t *result)

  Get the year (1970 onwards).

- MAMAExpDLL mama_status mamaDateTime_getMonth (const mamaDateTime dateTime, mama_u32_t *result)

  Get the month (1-12).

- MAMAExpDLL mama_status mamaDateTime_getDay (const mamaDateTime dateTime, mama_u32_t *result)

  Get the day of month (1-31).

- MAMAExpDLL mama_status mamaDateTime_getHour (const mamaDateTime dateTime, mama_u32_t *result)

  Get the hour (0-23).

- MAMAExpDLL mama_status mamaDateTime_getMinute (const mamaDateTime dateTime, mama_u32_t *result)

  Get the minute (0-59).

- MAMAExpDLL mama_status mamaDateTime_getSecond (const mamaDateTime dateTime, mama_u32_t *result)

  Get the second (0-59).

- MAMAExpDLL mama_status mamaDateTime_getMicrosecond (const mamaDateTime dateTime, mama_u32_t *result)

  Get the microsecond (0-999999).

- MAMAExpDLL mama_status mamaDateTime_getDayOfWeek (const mamaDateTime dateTime, mamaDayOfWeek *result)

  Get the day of week.

- MAMAExpDLL mama_status mamaDateTime_diffSeconds (const mamaDateTime t1, const mamaDateTime t0, mama_f64_t *result)

  Return the difference, in seconds (including fractions of seconds), between the two times.

- MAMAExpDLL mama_status mamaDateTime_diffSecondsSameDay (const mamaDateTime t1, const mamaDateTime t0, mama_f64_t *result)

  Return the difference, in seconds (including fractions of seconds), between the two times, ignoring any date information.

- MAMAExpDLL mama_status mamaDateTime_diffMicroseconds (const mamaDateTime t1, const mamaDateTime t0, mama_i64_t *result)

  Return the difference, in microseconds, between the two times.

5.3.1 Define Documentation

5.3.1.1 #define MAMA_DATE_TIME_HAS_DATE ((mamaDateTimeHints) 0x01)

5.3.1.2 #define MAMA_DATE_TIME_HAS_TIME ((mamaDateTimeHints) 0x02)

5.3.1.3 #define MAMA_DATE_TIME_NO_TIMEZONE ((mamaDateTimeHints) 0x04)

5.3.2 Typedef Documentation

5.3.2.1 typedef enum mamaDateTimePrecision_ mamaDateTimePrecision

5.3.2.2 typedef enum mamaDayOfWeek_ mamaDayOfWeek

5.3.2.3 typedef mama_u8_t mamaDateTimeHints

5.3.3 Enumeration Type Documentation

5.3.3.1 enum mamaDateTimePrecision_

Enumerator:

MAMA_DATE_TIME_PREC_SECONDS

MAMA_DATE_TIME_PREC_DECISECONDS

MAMA_DATE_TIME_PREC_CENTISECONDS

MAMA_DATE_TIME_PREC_MILLISECONDS

MAMA_DATE_TIME_PREC_MICROSECONDS

MAMA_DATE_TIME_PREC_DAYS

MAMA_DATE_TIME_PREC_MINUTES

MAMA_DATE_TIME_PREC_UNKNOWN

```
{
    MAMA_DATE_TIME_PREC_SECONDS      = 0,
    MAMA_DATE_TIME_PREC_DECISECONDS  = 1,
    MAMA_DATE_TIME_PREC_CENTISECONDS = 2,
```

```
    MAMA_DATE_TIME_PREC_MILLISECONDS = 3,
    MAMA_DATE_TIME_PREC_MICROSECONDS = 6,
    MAMA_DATE_TIME_PREC_DAYS         = 10,
    MAMA_DATE_TIME_PREC_MINUTES      = 12,
    MAMA_DATE_TIME_PREC_UNKNOWN      = 15
} mamaDateTimePrecision;
```

5.3.3.2 enum mamaDayOfWeek_

Enumerator:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

```
{
    Sunday = 0, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
} mamaDayOfWeek;
```

5.3.4 Function Documentation

5.3.4.1 MAMAExpDLL mama_status mamaDateTime_create (mamaDateTime * *dateTime*)

Create a date/time object.

Parameters:

dateTime The location of a mamaDateTime to store the result

5.3.4.2 MAMAExpDLL mama_status mamaDateTime_destroy (mamaDateTime *dateTime*)

Destroy a mamaDateTime object.

Parameters:

dateTime The date/time object to destroy.

5.3.4.3 MAMAExpDLL mama_status mamaDateTime_clear (mamaDateTime *dateTime*)

Clear a mamaDateTime object.

Parameters:

dateTime The date/time object to clear.

5.3.4.4 MAMAExpDLL mama_status mamaDateTime_clearDate (mamaDateTime *dateTime*)

Clear the date part of a mamaDateTime object.

Parameters:

dateTime The date/time object to clear (preserving the time of day).

5.3.4.5 MAMAExpDLL mama_status mamaDateTime_clearTime (mamaDateTime *dateTime*)

Clear the time of day part of a mamaDateTime object (preserving the date).

Parameters:

dateTime The date/time object to clear.

5.3.4.6 MAMAExpDLL mama_status mamaDateTime_copy (mamaDateTime *dest*, const mamaDateTime *src*)

5.3.4.7 MAMAExpDLL int mamaDateTime_empty (const mamaDateTime *dateTime*)

5.3.4.8 MAMAExpDLL int mamaDateTime_equal (const mamaDateTime *lhs*, const mamaDateTime *rhs*)

5.3.4.9 MAMAExpDLL int mamaDateTime_compare (const mamaDateTime *lhs*, const mamaDateTime *rhs*)

5.3.4.10 MAMAExpDLL mama_status mamaDateTime_setEpochTime (mamaDateTime *dateTime*, mama_u32_t *seconds*, mama_u32_t *microseconds*, mamaDateTimePrecision *precision*)

Set the date and time as seconds and microseconds since the Epoch (UTC time zone) with an option to designate the accuracy of the time.

Parameters:

dateTime The dateTime to set.

seconds The number of seconds since the Epoch.

microseconds The number of microseconds.

precision The precision of the time stamp.

5.3.4.11 MAMAExpDLL mama_status mamaDateTime_setEpochTimeF64 (mamaDateTime *dateTime*, mama_f64_t *seconds*)

Set the date and time as seconds (plus, possibly fractional seconds) since the Epoch (UTC time zone).

Fractional seconds will be rounded to microseconds.

Parameters:

dateTime The dateTime to set.

seconds The number of seconds since the Epoch.

5.3.4.12 MAMAExpDLL mama_status mamaDateTime_setEpochTimeMilliseconds (mamaDateTime *dateTime*, mama_u64_t *milliseconds*)

Set the date and time as milliseconds.

Parameters:

dateTime The dateTime to set.

milliseconds The number of milliseconds since the Epoch.

5.3.4.13 MAMAExpDLL mama_status mamaDateTime_setEpochTimeMicroseconds (mamaDateTime *dateTime*, mama_u64_t *milliseconds*)

Set the date and time as microseconds.

Parameters:

dateTime The dateTime to set.

milliseconds The number of microseconds since the Epoch.

5.3.4.14 MAMAExpDLL mama_status mamaDateTime_setWithHints (mamaDateTime *dateTime*, mama_u32_t *seconds*, mama_u32_t *microseconds*, mamaDateTimePrecision *precision*, mamaDateTimeHints *hints*)

Set the date and/or time with special, optional hints to indicate whether the date/time includes date information, time information and/or whether a user of the date/time should consider it in the context of a time zone.

Parameters:

dateTime The dateTime to set.

seconds The number of seconds (since the Epoch or start-of-day if no date).

microseconds The number of microseconds.

precision The precision of the date/time stamp.

hints Additional hints

5.3.4.15 MAMAExpDLL mama_status mamaDateTime_setPrecision (mamaDateTime *dateTime*, mamaDateTimePrecision *precision*)

Set the precision hint.

Parameters:

dateTime The dateTime to set.

precision The precision of the date/time stamp.

5.3.4.16 MAMAExpDLL mama_status mamaDateTime_setToNow (mamaDateTime *dateTime*)

Set the date and time to the current UTC time.

Precision and hints will be set appropriately.

Parameters:

dateTime The dateTime to set.

5.3.4.17 MAMAExpDLL mama_status mamaDateTime_setToMidnightToday (mamaDateTime *dateTime*, const mamaTimeZone *tz*)

Set the dateTime object to 12am of the current date in the timezone provided (or UTC if NULL).

Parameters:

dateTime The dateTime to set.

tz The timezone in which the date will be set.

5.3.4.18 MAMAExpDLL mama_status mamaDateTime_setWithPrecisionAndTz (mamaDateTime *dateTime*, mama_u32_t *year*, mama_u32_t *month*, mama_u32_t *day*, mama_u32_t *hour*, mama_u32_t *minute*, mama_u32_t *second*, mama_u32_t *microsecond*, mamaDateTimePrecision *precision*, const mamaTimeZone *tz*)

Set the entire date and time for the MamaDateTime.

The year, month and day parameters must all be integers greater than zero.

Parameters:

dateTime The dateTime to set.

year The year (must be 1970 or later).

month The month (1 - 12).

day The day (1 - 31).

hour The hour (0 - 23).

minute The minute (0 - 59).

second The second (0 - 59).

microsecond The second (0 - 999999).

precision An explicit precision, if known.

tz A timezone to shift from.

5.3.4.19 MAMAExpDLL mama_status mamaDateTime_setTime (mamaDateTime *dateTime*, mama_u32_t *hour*, mama_u32_t *minute*, mama_u32_t *second*, mama_u32_t *microsecond*)

Set the time-of-day portion of the MamaDateTime.

The date portion is not affected.

Parameters:

dateTime The dateTime to set.

hour The hour (0 - 23).

minute The minute (0 - 59).

second The second (0 - 59).

microsecond The microsecond (0 - 999999).

5.3.4.20 MAMAExpDLL mama_status mamaDateTime_setTimeWithPrecisionAndTz (mamaDateTime *dateTime*, mama_u32_t *hour*, mama_u32_t *minute*, mama_u32_t *second*, mama_u32_t *microsecond*, mamaDateTimePrecision *precision*, const mamaTimeZone *tz*)

Set the time-of-day portion of the MamaDateTime with explicit precision.

The date portion is not affected.

Parameters:

dateTime The dateTime to set.

hour The hour (0 - 23).

minute The minute (0 - 59).

second The second (0 - 59).

microsecond The microsecond (0 - 999999).

precision An explicit precision, if known.

tz A timezone to shift from.

5.3.4.21 MAMAExpDLL mama_status mamaDateTime_setDate (mamaDateTime *dateTime*, mama_u32_t *year*, mama_u32_t *month*, mama_u32_t *day*)

Set the date portion of the MamaDateTime.

The time-of-day portion is not affected.

Parameters:

dateTime The dateTime to set.

year The year (must be 1970 or later).

month The month (1 - 12).

day The day (1 - 31).

5.3.4.22 MAMAExpDLL mama_status mamaDateTime_copyTime (mamaDateTime *dest*, const mamaDateTime *src*)

Copy the time-of-day portion of the mamaDateTime.

The date portion is not affected.

Parameters:

dest The dateTime to copy to.

src The dateTime to copy from.

5.3.4.23 MAMAExpDLL mama_status mamaDateTime_copyDate (mamaDateTime *dest*, const mamaDateTime *src*)

Copy the date portion of the mamaDateTime.

The time-of-day portion is not affected.

Parameters:

dest The dateTime to copy to.

src The dateTime to copy from.

5.3.4.24 MAMAExpDLL mama_status mamaDateTime_hasTime (const mamaDateTime *dateTime*, mama_bool_t * *result*)

Determine whether the time-of-day portion of the MamaDateTime is set.

Parameters:

dateTime The dateTime to check.

result Address of a bool to store the result. 0 false.

5.3.4.25 MAMAExpDLL mama_status mamaDateTime_hasDate (const mamaDateTime *dateTime*, mama_bool_t * *result*)

Determine whether the date portion of the MamaDateTime is set.

Parameters:

dateTime The dateTime to check.

result Address of mama_bool_t to store the result. 0 false.

5.3.4.26 MAMAExpDLL mama_status mamaDateTime_addSeconds (mamaDateTime *dateTime*, mama_f64_t *seconds*)

Add a number of seconds, including fractional seconds (may be negative).

Parameters:

dateTime The dateTime to set.

seconds The number of seconds to add (or subtract).

5.3.4.27 MAMAExpDLL mama_status mamaDateTime_addWholeSeconds (mamaDateTime *dateTime*, mama_i32_t *seconds*)

Add a number of whole seconds (may be negative).

Parameters:

dateTime The dateTime to set.

seconds The number of seconds to add (or subtract).

5.3.4.28 MAMAExpDLL mama_status mamaDateTime_addMicroseconds (mamaDateTime *dateTime*, mama_i64_t *microseconds*)

Add a number of microseconds (may be negative) Add a number of microseconds (may be negative and greater than 1000000 or less than -1000000).

Parameters:

dateTime The dateTime to set.

microseconds The number of microseconds to add. Can be positive or negative. Note that there are no restrictions on the magnitude of this value.

5.3.4.29 MAMAExpDLL mama_status mamaDateTime_setFromString (mamaDateTime *dateTime*, const char * *str*)

Set the date and time from a string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

(YYYY/mm/dd also works.) The precision of subseconds is determined by the number of digits after the decimal point.

Parameters:

dateTime The dateTime to set.

str The string representation of some date and/or time.

5.3.4.30 MAMAExpDLL mama_status mamaDateTime_setFromStringWithTz (mamaDateTime *dateTime*, const char * *str*, const mamaTimeZone *tz*)

Set the date and time from a string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

(YYYY/mm/dd also works.) The precision of subseconds is determined by the number of digits after the decimal point. If tz is not NULL, the time string is assumed to be set in a different time zone and will be adjusted to UTC accordingly.

Parameters:

dateTime The dateTime to set.

str The string representation of some date and/or time.

tz A timezone to shift from.

5.3.4.31 MAMAExpDLL mama_status mamaDateTime_setFromStringBuffer (mamaDateTime *dateTime*, const char * *str*, mama_size_t *strLen*)

Set the date and time from an unterminated string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

(YYYY/mm/dd also works.) The precision of subseconds is determined by the number of digits after the decimal point.

Parameters:

dateTime The dateTime to set.

str The string representation of some date and/or time.

strLen The length of the unterminated string date/time.

5.3.4.32 MAMAExpDLL mama_status mamaDateTime_setFromStringBuffer-WithTz (mamaDateTime *dateTime*, const char * *str*, mama_size_t *strLen*, const mamaTimeZone *tz*)

Set the date and time from an unterminated string representation of format YYYY-mm-dd HH:MM:SS.mmmmmm.

(YYYY/mm/dd also works.) The precision of subseconds is determined by the number of digits after the decimal point. If tz is not NULL, the time string is assumed to be set in a different time zone and will be adjusted to UTC accordingly.

Parameters:

dateTime The dateTime to set.

str The string representation of some date and/or time.

strLen The length of the unterminated string date/time.

tz A timezone to shift from.

5.3.4.33 MAMAExpDLL mama_status mamaDateTime_getEpochTime (const mamaDateTime *dateTime*, mama_u32_t * *seconds*, mama_u32_t * *microseconds*, mamaDateTimePrecision * *precision*)

Get the date and time as seconds and microseconds since the Epoch (UTC time zone).

Parameters:

dateTime The dateTime to set.

seconds The number of seconds since the Epoch.

microseconds The number of additional microseconds.

precision The precision of the returned time value.

5.3.4.34 MAMAExpDLL mama_status mamaDateTime_getEpochTimeWithTz (const mamaDateTime *dateTime*, mama_u32_t * *seconds*, mama_u32_t * *microseconds*, mamaDateTimePrecision * *precision*, const mamaTimeZone *tz*)

Get the date and time as seconds and microseconds since the Epoch in the timezone supplied.

Parameters:

dateTime The dateTime to set.

seconds The number of seconds since the Epoch.

microseconds The number of additional microseconds.

precision The precision of the returned time value.

tz The timezone for the returned values.

5.3.4.35 MAMAExpDLL mama_status mamaDateTime_getEpochTime-Microseconds (const mamaDateTime *dateTime*, mama_u64_t * *microseconds*)

Get the date and time as microseconds since the Epoch, (using the UTC timezone).

Parameters:

← ***dateTime*** The dateTime to obtain the number of microseconds from.

→ ***microseconds*** The number of microseconds since the Epoch.

Returns:

Indicates whether the function succeeded or failed and could be one of:

- MAMA_STATUS_INVALID_ARG
- MAMA_STATUS_OK

5.3.4.36 MAMAExpDLL mama_status mamaDateTime_getEpochTime-MicrosecondsWithTz (const mamaDateTime *dateTime*, mama_u64_t * *microseconds*, const mamaTimeZone *tz*)

Get the date and time as microseconds since the Epoch in the supplied time zone.

Parameters:

← ***dateTime*** The dateTime to obtain the number of microseconds from.

→ ***microseconds*** The number of microseconds since the Epoch.

int] tz The timezone.

Returns:

Indicates whether the function succeeded or failed and could be one of:

- MAMA_STATUS_INVALID_ARG
- MAMA_STATUS_OK

5.3.4.37 MAMAExpDLL mama_status mamaDateTime_getEpochTime-Milliseconds (const mamaDateTime *dateTime*, mama_u64_t * *milliseconds*)

Get the date and time as milliseconds since the Epoch (UTC time zone).

Parameters:

dateTime The dateTime to set.

milliseconds The number of milliseconds since the Epoch.

5.3.4.38 MAMAExpDLL mama_status mamaDateTime_getEpochTime-MillisecondsWithTz (const mamaDateTime *dateTime*, mama_u64_t * *milliseconds*, const mamaTimeZone *tz*)

Get the date and time as milliseconds since the Epoch in the timezone supplied.

Parameters:

dateTime The dateTime to set.

milliseconds The number of milliseconds since the Epoch.

tz The timezone for the returned values.

5.3.4.39 MAMAExpDLL mama_status mamaDateTime_getEpochTimeSeconds (const mamaDateTime *dateTime*, mama_f64_t * *seconds*)

Get the date and time as seconds since the Epoch (UTC time zone).

Parameters:

dateTime The dateTime to set

seconds The number of seconds (including partial seconds) since the Epoch.

5.3.4.40 MAMAExpDLL mama_status mamaDateTime_getEpochTime-SecondsWithCheck (const mamaDateTime *dateTime*, mama_f64_t * *seconds*)

Get the date and time as seconds since the Epoch, (using the UTC timezone).

If no date value is contained in the dateTime then it will be set to today's date and the calculation made.

Parameters:

← ***dateTime*** The dateTime to obtain the number of microseconds from.

→ ***seconds*** The number of seconds, (including partial seconds), since the Epoch.

Returns:

Indicates whether the function succeeded or failed and could be one of:

- MAMA_STATUS_INVALID_ARG
- MAMA_STATUS_SYSTEM_ERROR
- MAMA_STATUS_NULL_ARG
- MAMA_STATUS_OK

5.3.4.41 MAMAExpDLL mama_status mamaDateTime_getEpochTime-SecondsWithTz (const mamaDateTime *dateTime*, mama_f64_t * *seconds*, const mamaTimeZone *tz*)

Get the date and time as seconds since the Epoch in the timezone supplied.

Parameters:

dateTime The dateTime to set

seconds The number of seconds (including partial seconds) since the Epoch.

tz The timezone for the returned values.

5.3.4.42 MAMAExpDLL mama_status mamaDateTime_getWithHints (const mamaDateTime *dateTime*, mama_u32_t * *seconds*, mama_u32_t * *microseconds*, mamaDateTimePrecision * *precision*, mama_u8_t * *hints*)

Get the date and/or time with special, optional hints to indicate whether the date/time includes date information, time information and/or whether a user of the date/time should consider it in the context of a time zone.

Parameters:

dateTime The dateTime to set.

seconds The number of seconds (since the Epoch or start-of-day if no date).

microseconds The number of microseconds.

precision The precision of the date/time stamp.

hints Additional hints

5.3.4.43 MAMAExpDLL mama_status mamaDateTime_getStructTimeVal (const mamaDateTime *dateTime*, struct timeval * *result*)

Get the date/time as a "struct timeval".

Parameters:

dateTime The dateTime to set.

result The struct timeval to fill in.

5.3.4.44 MAMAExpDLL mama_status mamaDateTime_getStructTimeVal-WithTz (const mamaDateTime *dateTime*, struct timeval * *result*, const mamaTimeZone *tz*)

Get the date/time as a "struct timeval" in the timezone supplied.

Parameters:

dateTime The dateTime to set.

result The struct timeval to fill in.

tz The timezone for the returned values.

5.3.4.45 MAMAExpDLL mama_status mamaDateTime_getStructTm (const mamaDateTime *dateTime*, struct tm * *result*)

Get the date/time as a "struct tm".

Parameters:

dateTime The dateTime to set.

result The struct timeval to fill in.

5.3.4.46 MAMAExpDLL mama_status mamaDateTime_getStructTmWithTz (const mamaDateTime *dateTime*, struct tm * *result*, const mamaTimeZone *tz*)

Get the date/time as a "struct tm" in the timezone supplied.

Parameters:

dateTime The dateTime to set.

result The struct timeval to fill in.

tz The timezone for the returned values.

5.3.4.47 MAMAExpDLL mama_status mamaDateTime_getAsString (const mamaDateTime *dateTime*, char * *str*, mama_size_t *maxLen*)

Get the date and/or time as a string.

If no date information is available, no date is printed in the resulting string. The format for dates is YYYY-mm-dd (the ISO 8601 date format) and the format for times is HH:MM:SS.mmmmmmm (where the precision of the subseconds may vary depending upon any precision hints available).

Parameters:

dateTime The dateTime to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.3.4.48 MAMAExpDLL mama_status mamaDateTime_getTimeAsString (const mamaDateTime *dateTime*, char * *str*, mama_size_t *maxLen*)

Get the time (no date) as a string.

The format for times is HH:MM:SS.mmmmmmm (where the precision of the subseconds may vary depending upon any precision hints available).

Parameters:

dateTime The dateTime to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.3.4.49 MAMAExpDLL mama_status mamaDateTime_getDateAsString (const mamaDateTime *dateTime*, char * *str*, mama_size_t *maxLen*)

Get the date (no time) as a string.

The format for dates is YYYY-mm-dd (the ISO 8601 date format).

Parameters:

dateTime The dateTime to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.3.4.50 MAMAExpDLL mama_status mamaDateTime_getAsFormattedString (const mamaDateTime *dateTime*, char * *str*, mama_size_t *maxLen*, const char * *format*)

Get the date and/or time as a string using the format provided.

The format string can be the common ones supported by strftime() (Y, m, d, F, T, Y, H, M, S, B, b, h,), with the additional format strings, "%:" and "%;", which represents the number of subseconds in millis. "%;" includes the dot and only prints the subseconds if they are non-zero. "%:" does not include the dot, and prints "000" for 0 milliseconds. "%." and "%," are identical to "%:" and "%;" except they use the internal precision field to determine how many decimal places to print.

Parameters:

dateTime The dateTime to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

format The output format.

5.3.4.51 MAMAExpDLL mama_status mamaDateTime_getAsFormatted-StringWithTz (const mamaDateTime *dateTime*, char * *str*, mama_size_t *maxLen*, const char * *format*, const mamaTimeZone *tz*)

Get the date and/or time as a string using the format provided.

The format string can be the common ones supported by strftime() (Y, m, d, F, T, Y, H, M, S, B, b, h,), with the additional format strings, "%:" and "%;", which represents the number of subseconds in millis. "%;" includes the dot and only prints the subseconds if they are non-zero. "%:" does not include the dot, and prints "000" for 0 milliseconds. "%." and "%," are identical to "%:" and "%;" except they use the internal precision field to determine how many decimal places to print.

Parameters:

dateTime The dateTime to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

format The output format.

tz The timezone for the returned date/time.

5.3.4.52 MAMAExpDLL mama_status mamaDateTime_getYear (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the year (1970 onwards).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.53 MAMAExpDLL mama_status mamaDateTime_getMonth (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the month (1-12).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.54 MAMAExpDLL mama_status mamaDateTime_getDay (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the day of month (1-31).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.55 MAMAExpDLL mama_status mamaDateTime_getHour (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the hour (0-23).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.56 MAMAExpDLL mama_status mamaDateTime_getMinute (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the minute (0-59).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.57 MAMAExpDLL mama_status mamaDateTime_getSecond (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the second (0-59).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.58 MAMAExpDLL mama_status mamaDateTime_getMicrosecond (const mamaDateTime *dateTime*, mama_u32_t * *result*)

Get the microsecond (0-999999).

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.59 MAMAExpDLL mama_status mamaDateTime_getDayOfWeek (const mamaDateTime *dateTime*, mamaDayOfWeek * *result*)

Get the day of week.

Parameters:

dateTime The dateTime from which to get the result.

result The result of the get method.

5.3.4.60 MAMAExpDLL mama_status mamaDateTime_diffSeconds (const mamaDateTime *t1*, const mamaDateTime *t0*, mama_f64_t * *result*)

Return the difference, in seconds (including fractions of seconds), between the two times.

5.3.4.61 MAMAExpDLL mama_status mamaDateTime_diffSecondsSameDay (const mamaDateTime *t1*, const mamaDateTime *t0*, mama_f64_t * *result*)

Return the difference, in seconds (including fractions of seconds), between the two times, ignoring any date information.

5.3.4.62 MAMAExpDLL mama_status mamaDateTime_diffMicroseconds (const mamaDateTime *t1*, const mamaDateTime *t0*, mama_i64_t * *result*)

Return the difference, in microseconds, between the two times.

5.4 dictionary.h File Reference

```
#include "mama/types.h"
#include "mama/fielddesc.h"
```

Data Structures

- struct mamaDictionaryCallbackSet

  A structure containing the callbacks for dictionary creation.

Typedefs

- typedef void(MAMACALLTYPE ∗) mamaDictionary_completeCallback (mamaDictionary, void ∗)

  Typedefs for the dictionary callbacks.

- typedef void(MAMACALLTYPE ∗) mamaDictionary_timeoutCallback (mamaDictionary, void ∗)
- typedef void(MAMACALLTYPE ∗) mamaDictionary_errorCallback (mamaDictionary, const char ∗, void ∗)

Functions

- MAMAExpDLL mama_status mama_createDictionary (mamaDictionary ∗dictionary, mamaQueue queue, mamaDictionaryCallbackSet dictionaryCallbacks, mamaSource source, double timeout, int retries, void ∗closure)

  Create a data dictionary from a subscription.

- MAMAExpDLL mama_status mamaDictionary_create (mamaDictionary ∗dictionary)

  Create an empty mamaDictionary so that can be populated at a later stage via a call to buildDictionaryFromMessage () or populated manually via calls to addFieldDescriptor ().

- MAMAExpDLL mama_status mamaDictionary_destroy (mamaDictionary dictionary)

  Destroy this mamaDictionary object and free all its resources.

- MAMAExpDLL mama_status mamaDictionary_getFeedName (mamaDictionary dictionary, const char ∗∗result)

Return the dictionary source feed name.

- MAMAExpDLL mama_status mamaDictionary_getFeedHost (mamaDictionary dictionary, const char **result)

  Return the dictionary source feed host.

- MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByFid (mamaDictionary dictionary, mamaFieldDescriptor *result, mama_fid_t fid)

  Return the mamaFieldDescriptor with the specified field FID.

- MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByIndex (mamaDictionary dictionary, mamaFieldDescriptor *result, unsigned short index)

  Return the field with the corresponding zero based index.

- MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorBy-Name (mamaDictionary dictionary, mamaFieldDescriptor *result, const char *fname)

  Return the descriptor of the field with the specified name.

- MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByNameAll (mamaDictionary dictionary, const char *fname, mamaFieldDescriptor *desc-List, size_t *size)

  Return an array of mamaFieldDescriptor which includes every field in the dictionary with the specified name.

- MAMAExpDLL mama_status mamaDictionary_getMaxFid (mamaDictionary dictionary, mama_fid_t *value)

  Return the highest field identifier.

- MAMAExpDLL mama_status mamaDictionary_getSize (mamaDictionary dictionary, size_t *value)

  Return the number of fields in the dictionary.

- MAMAExpDLL mama_status mamaDictionary_hasDuplicates (mama-Dictionary dictionary, int *value)

  Return true if there are multiple fields with the same name.

- MAMAExpDLL mama_status mamaDictionary_buildDictionaryFromMessage (mamaDictionary dictionary, const mamaMsg msg)

  Build a data dictionary from the specified message.

- MAMAExpDLL mama_status mamaDictionary_getDictionaryMessage (mama-Dictionary dictionary, mamaMsg *msg)

Get the underlying message for the data dictionary.

- MAMAExpDLL mama_status mamaDictionary_createFieldDescriptor (mamaDictionary dictionary, mama_fid_t fid, const char *name, mamaFieldType type, mamaFieldDescriptor *descriptor)

  Create a new field descriptor and add it to the dictionary.

- MAMAExpDLL mama_status mamaDictionary_setMaxFid (mamaDictionary dictionary, mama_size_t maxFid)

  Tell the dictionary what the probable maximum fid in the data dictionary may be.

- MAMAExpDLL mama_status mamaDictionary_writeToFile (mamaDictionary dictionary, const char *fileName)

  Write the data dictionary to a file.

- MAMAExpDLL mama_status mamaDictionary_populateFromFile (mamaDictionary dictionary, const char *fileName)

  Populate a dictionary from the contents of a file.

5.4.1 Typedef Documentation

5.4.1.1 typedef void(MAMACALLTYPE *) mamaDictionary_completeCallback(mamaDictionary, void *)

Typedefs for the dictionary callbacks.

- mamaDictionary_setErrorCallback ()
- mamaDictionary_setCompleteCallback ()
- mamaDictionary_setTimeoutCallback ()

5.4.1.2 typedef void(MAMACALLTYPE ∗) mamaDictionary_timeoutCallback(mamaDictionary, void ∗)

5.4.1.3 typedef void(MAMACALLTYPE ∗) mamaDictionary_errorCallback(mamaDictionary, const char ∗, void ∗)

5.4.2 Function Documentation

5.4.2.1 MAMAExpDLL mama_status mama_createDictionary (mamaDictionary ∗ *dictionary*, mamaQueue *queue*, mamaDictionaryCallbackSet *dictionaryCallbacks*, mamaSource *source*, double *timeout*, int *retries*, void ∗ *closure*)

Create a data dictionary from a subscription.

Parameters:

dictionary A pointer for the dictionary being created.

queue The mama queue.

dictionaryCallbacks A mamaDictionaryCallbackSet with the callbacks for completion, errors and timeouts.

source The mamaSource identifying the source of the dictionary.

timeout the timeout

retries number of retries

closure A user supplied value passed to the callbacks.

5.4.2.2 MAMAExpDLL mama_status mamaDictionary_create (mamaDictionary ∗ *dictionary*)

Create an empty mamaDictionary so that can be populated at a later stage via a call to buildDictionaryFromMessage () or populated manually via calls to addFieldDescriptor ().

Parameters:

dictionary The address to where the dictionary will be written

5.4.2.3 MAMAExpDLL mama_status mamaDictionary_destroy (mamaDictionary *dictionary*)

Destroy this mamaDictionary object and free all its resources.

Parameters:

dictionary The dictionary.

5.4.2.4 MAMAExpDLL mama_status mamaDictionary_getFeedName (mamaDictionary *dictionary*, const char ** *result*)

Return the dictionary source feed name.

Parameters:

dictionary The dictionary.

result (out) points to the feed name

5.4.2.5 MAMAExpDLL mama_status mamaDictionary_getFeedHost (mamaDictionary *dictionary*, const char ** *result*)

Return the dictionary source feed host.

Parameters:

dictionary The dictionary.

result (out) points to the feed host

5.4.2.6 MAMAExpDLL mama_status mamaDictionary_getFieldDescriptor-ByFid (mamaDictionary *dictionary*, mamaFieldDescriptor * *result*, mama_fid_t *fid*)

Return the mamaFieldDescriptor with the specified field FID.

This method is very efficient (constant time).

Parameters:

dictionary The dictionary.

fid The field id.

result (out) points to the mamaFieldDescriptor (not a copy)

5.4.2.7 MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByIndex (mamaDictionary *dictionary*, mamaFieldDescriptor * *result*, unsigned short *index*)

Return the field with the corresponding zero based index.

This method is O (N) with respect to the size of the dictionary.

Parameters:

dictionary The dictionary.

index The zero-based index.

result The result.

5.4.2.8 MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByName (mamaDictionary *dictionary*, mamaFieldDescriptor * *result*, const char * *fname*)

Return the descriptor of the field with the specified name.

If there is more than one field with the same name, the one with the lowest field id is returned.

Parameters:

dictionary The dictionary.

result the result * or NULL if no such field

fname The name of the field to search for.

5.4.2.9 MAMAExpDLL mama_status mamaDictionary_getFieldDescriptorByNameAll (mamaDictionary *dictionary*, const char * *fname*, mamaFieldDescriptor * *descList*, size_t * *size*)

Return an array of mamaFieldDescriptor which includes every field in the dictionary with the specified name.

The caller is responsible for allocating descList with room enough for all possible duplicate fields (use mamaDictionary_getSize () to be safe).

Parameters:

dictionary The dictionary.

fname (in) The name to search dictionary for.

descList (out) An array of mamaFieldDescriptor objects, which are not copies and should not be destroyed by the caller.

size (out) The final number of entries in descList. The value should be intialised to the size allocated to decList. This will be modified to the actual number found on return, or will return once this value has been found.

5.4.2.10 MAMAExpDLL mama_status mamaDictionary_getMaxFid (mamaDictionary *dictionary*, mama_fid_t * *value*)

Return the highest field identifier.

Parameters:

dictionary The dictionary.

value A pointer that will contain highest FID.

5.4.2.11 MAMAExpDLL mama_status mamaDictionary_getSize (mamaDictionary *dictionary*, size_t * *value*)

Return the number of fields in the dictionary.

Parameters:

dictionary The dictionary.

value The number of entries in the dictionary.

5.4.2.12 MAMAExpDLL mama_status mamaDictionary_hasDuplicates (mamaDictionary *dictionary*, int * *value*)

Return true if there are multiple fields with the same name.

Parameters:

dictionary The dictionary.

value 1 if there are duplicates, 0 otherwise.

5.4.2.13 MAMAExpDLL mama_status mamaDictionary_buildDictionaryFromMessage (mamaDictionary *dictionary*, const mamaMsg *msg*)

Build a data dictionary from the specified message.

Parameters:

dictionary The dictionary

msg A mamaMsg representing the contents of a data dictionary.

5.4.2.14 MAMAExpDLL mama_status mamaDictionary_getDictionaryMessage (mamaDictionary *dictionary*, mamaMsg * *msg*)

Get the underlying message for the data dictionary.

A new message instance is created each time this function is called. It is the responsibility for the caller to destroy the message when no longer required.

Parameters:

dictionary The Dictionary

msg The address of the mamaMsg where the result is to be written

5.4.2.15 MAMAExpDLL mama_status mamaDictionary_createFieldDescriptor (mamaDictionary *dictionary*, mama_fid_t *fid*, const char * *name*, mamaFieldType *type*, mamaFieldDescriptor * *descriptor*)

Create a new field descriptor and add it to the dictionary.

New fields can be added to an existing dictionary obtained from the MAMA infrastructure. This function can also be used to manually populate a new data dictionary.

Parameters:

dictionary The dictionary to which the field is to be added.

fid The fid for the new field descriptor.

name The name for the new field descriptor.

type The type for the new field descriptor.

descriptor The newly created fieldDescriptor. NULL can be specified.

5.4.2.16 MAMAExpDLL mama_status mamaDictionary_setMaxFid (mamaDictionary *dictionary*, mama_size_t *maxFid*)

Tell the dictionary what the probable maximum fid in the data dictionary may be.

This is not necessary but will aid performance for manually creating a new dictionary or adding new fields to an existing dictionary.

Calling this function ensures that there is capacity in the dictionary for field descriptors with fids up to the max specified.

Fields with fids greater than specified can be added to the dictionary but this will incur the overhead of allocating more memory and copying dictionary elements.

Parameters:

dictionary The mamaDictionary.

maxFid The probable maximum fid being added to the dictionary.

5.4.2.17 MAMAExpDLL mama_status mamaDictionary_writeToFile (mamaDictionary *dictionary*, const char * *fileName*)

Write the data dictionary to a file.

The dictionary will be written in the form: fid|fieldName|fieldType

Parameters:

dictionary The dictionary to serialize.

fileName The name of the file to serialize the dictionary to. This can be a fully qualified name, relative or a file on the $WOMBAT_PATH

5.4.2.18 MAMAExpDLL mama_status mamaDictionary_populateFromFile (mamaDictionary *dictionary*, const char * *fileName*)

Populate a dictionary from the contents of a file.

Can be used to add additional fields to an existing dictionary or to populate a new dictionary.

Parameters:

dictionary The dictionary to populate.

fileName The file from which to populate the dictionary. This can be a fully qualified name, relative or a file on the $WOMBAT_PATH

5.5 dqpublisher.h File Reference

```
#include "mama/types.h"
#include "wombat/wConfig.h"
```

Functions

- MAMAExpDLL mama_status mamaDQPublisher_allocate (mamaDQPublisher *result)

  Allocate a mamaDQPublisher.

- MAMAExpDLL mama_status mamaDQPublisher_create (mamaDQPublisher pub, mamaTransport transport, const char *topic)

  Create a mamaDQPublisher for the corresponding transport.

- MAMAExpDLL mama_status mamaDQPublisher_send (mamaDQPublisher pub, mamaMsg msg)

  Send a message.

- MAMAExpDLL mama_status mamaDQPublisher_sendReply (mamaDQPublisher pub, mamaMsg request, mamaMsg reply)
- MAMAExpDLL mama_status mamaDQPublisher_sendReplyWithHandle (mamaDQPublisher pub, mamaMsgReply replyAddress, mamaMsg reply)
- MAMAExpDLL void mamaDQPublisher_destroy (mamaDQPublisher pub)
- MAMAExpDLL void mamaDQPublisher_setStatus (mamaDQPublisher pub, mamaMsgStatus status)
- MAMAExpDLL void mamaDQPublisher_setSenderId (mamaDQPublisher pub, uint64_t senderid)
- MAMAExpDLL void mamaDQPublisher_setSeqNum (mamaDQPublisher pub, mama_seqnum_t num)
- MAMAExpDLL void mamaDQPublisher_setClosure (mamaDQPublisher pub, void *closure)
- MAMAExpDLL void mamaDQPublisher_enableSendTime (mamaDQPublisher pub, mama_bool_t enable)
- MAMAExpDLL void * mamaDQPublisher_getClosure (mamaDQPublisher pub)
- MAMAExpDLL void mamaDQPublisher_setCache (mamaDQPublisher pub, void *cache)
- MAMAExpDLL void * mamaDQPublisher_getCache (mamaDQPublisher pub)

5.5.1 Function Documentation

5.5.1.1 MAMAExpDLL mama_status mamaDQPublisher_allocate (mamaDQPublisher * *result*)

Allocate a mamaDQPublisher.

Parameters:

result A pointer to hold the mamaDQPublisher.

5.5.1.2 MAMAExpDLL mama_status mamaDQPublisher_create (mamaDQPublisher *pub*, mamaTransport *transport*, const char * *topic*)

Create a mamaDQPublisher for the corresponding transport.

Parameters:

pub A pointer to the mamaDQPublisher.

transport The transport.

topic Symbol on which to publish.

5.5.1.3 MAMAExpDLL mama_status mamaDQPublisher_send (mamaDQPublisher *pub*, mamaMsg *msg*)

Send a message.

Parameters:

publisher The publish from which to send the message.

msg The mamaMsg to send.

5.5.1.4 MAMAExpDLL mama_status mamaDQPublisher_sendReply (mamaDQPublisher *pub*, mamaMsg *request*, mamaMsg *reply*)

5.5.1.5 MAMAExpDLL mama_status mamaDQPublisher_sendReplyWithHandle (mamaDQPublisher *pub*, mamaMsgReply *replyAddress*, mamaMsg *reply*)

5.5.1.6 MAMAExpDLL void mamaDQPublisher_destroy (mamaDQPublisher *pub*)

5.5.1.7 MAMAExpDLL void mamaDQPublisher_setStatus (mamaDQPublisher *pub*, mamaMsgStatus *status*)

5.5.1.8 MAMAExpDLL void mamaDQPublisher_setSenderId (mamaDQPublisher *pub*, uint64_t *senderid*)

5.5.1.9 MAMAExpDLL void mamaDQPublisher_setSeqNum (mamaDQPublisher *pub*, mama_seqnum_t *num*)

5.5.1.10 MAMAExpDLL void mamaDQPublisher_setClosure (mamaDQPublisher *pub*, void * *closure*)

5.5.1.11 MAMAExpDLL void mamaDQPublisher_enableSendTime (mamaDQPublisher *pub*, mama_bool_t *enable*)

5.5.1.12 MAMAExpDLL void* mamaDQPublisher_getClosure (mamaDQPublisher *pub*)

5.5.1.13 MAMAExpDLL void mamaDQPublisher_setCache (mamaDQPublisher *pub*, void * *cache*)

5.5.1.14 MAMAExpDLL void* mamaDQPublisher_getCache (mamaDQPublisher *pub*)

5.6 dqpublishermanager.h File Reference

```
#include "mama/types.h"
#include "wombat/wConfig.h"
```

Data Structures

- struct mamaPublishTopic_
- struct mamaDQPublisherManagerCallbacks_

Typedefs

- typedef mamaPublishTopic_ mamaPublishTopic
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherCreateCB (mamaDQPublisherManager manager)
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherNewRequestCB (mamaDQPublisherManager manager, const char ∗symbol, short subType, short msgType, mamaMsg msg)
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherRequestCB (mamaDQPublisherManager manager, mamaPublishTopic ∗info, short subType, short msgType, mamaMsg msg)
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherRefreshCB (mamaDQPublisherManager manager, mamaPublishTopic ∗info, short subType, short msgType, mamaMsg msg)
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherErrorCB (mamaDQPublisherManager manager, mama_status status, const char ∗errortxt, mamaMsg msg)
- typedef void(MAMACALLTYPE ∗) mamaDQPublisherMsgCB (mamaDQPublisherManager manager, mamaMsg msg)
- typedef mamaDQPublisherManagerCallbacks_ mamaDQPublisherManagerCallbacks

Functions

- MAMAExpDLL mama_status mamaDQPublisherManager_allocate (mamaDQPublisherManager ∗result)

  Create a mama publisher for the corresponding transport.

- MAMAExpDLL mama_status mamaDQPublisherManager_create (mamaDQPublisherManager dqPublisher, mamaTransport transport, mamaQueue queue, const mamaDQPublisherManagerCallbacks ∗callback, const char ∗sourcename, const char ∗root, void ∗closure)

Send a message from the specified publisher.

- MAMAExpDLL void mamaDQPublisherManager_destroy (mamaDQPublisherManager manager)
- MAMAExpDLL mama_status mamaDQPublisherManager_addPublisher (mamaDQPublisherManager manager, const char ∗symbol, mamaDQPublisher pub, void ∗cache)
- MAMAExpDLL mama_status mamaDQPublisherManager_removePublisher (mamaDQPublisherManager manager, const char ∗symbol, mamaDQPublisher ∗pub)
- MAMAExpDLL mama_status mamaDQPublisherManager_createPublisher (mamaDQPublisherManager manager, const char ∗symbol, void ∗cache, mamaDQPublisher ∗newPublisher)
- MAMAExpDLL mama_status mamaDQPublisherManager_destroyPublisher (mamaDQPublisherManager manager, const char ∗symbol)
- MAMAExpDLL void mamaDQPublisherManager_setStatus (mamaDQPublisherManager manager, mamaMsgStatus status)
- MAMAExpDLL void ∗ mamaDQPublisherManager_getClosure (mamaDQPublisherManager manager)
- MAMAExpDLL void mamaDQPublisherManager_setSenderId (mamaDQPublisherManager manager, uint64_t senderid)
- MAMAExpDLL void mamaDQPublisherManager_setSeqNum (mamaDQPublisherManager manager, mama_seqnum_t num)
- MAMAExpDLL mama_status mamaDQPublisherManager_sendSyncRequest (mamaDQPublisherManager manager, mama_u16_t nummsg, mama_f64_t delay, mama_f64_t duration)
- MAMAExpDLL mama_status mamaDQPublisherManager_sendNoSubscribers (mamaDQPublisherManager manager, const char ∗symbol)
- MAMAExpDLL void mamaDQPublisherManager_enableSendTime (mamaDQPublisherManager manager, mama_bool_t enable)

5.6.1 Typedef Documentation

5.6.1.1 typedef struct mamaPublishTopic_ mamaPublishTopic

5.6.1.2 typedef void(MAMACALLTYPE ∗) mamaDQPublisherCreateCB(mamaDQPublisherManager manager)

5.6.1.3 typedef void(MAMACALLTYPE ∗) mamaDQPublisherNewRequestCB(mamaDQPublisherManager manager, const char ∗symbol, short subType, short msgType, mamaMsg msg)

5.6.1.4 typedef void(MAMACALLTYPE ∗) mamaDQPublisherRequestCB(mamaDQPublisherManager manager, mamaPublishTopic ∗info, short subType, short msgType, mamaMsg msg)

5.6.1.5 typedef void(MAMACALLTYPE ∗) mamaDQPublisherRefreshCB(mamaDQPublisherManager manager, mamaPublishTopic ∗info, short subType, short msgType, mamaMsg msg)

5.6.1.6 typedef void(MAMACALLTYPE ∗) mamaDQPublisherErrorCB(mamaDQPublisherManager manager, mama_status status, const char ∗errortxt, mamaMsg msg)

5.6.1.7 typedef void(MAMACALLTYPE ∗) mamaDQPublisherMsgCB(mamaDQPublisherManager manager, mamaMsg msg)

5.6.1.8 typedef struct mamaDQPublisherManagerCallbacks_ mamaDQPublisherManagerCallbacks

5.6.2 Function Documentation

5.6.2.1 MAMAExpDLL mama_status mamaDQPublisherManager_allocate (mamaDQPublisherManager ∗ *result*)

Create a mama publisher for the corresponding transport.

If the transport is a marketdata transport, as opposed to a "basic" transport, the topic corresponds to the symbol. For a basic transport, the source and root get ignored.

Parameters:

result A pointer to hold the resulting mamaPublisher.

tport The transport.

symbol Symbol on which to publish.

source The source for market data publishers. (e.g. source.symbol)

root The root for market data publishers. Used internally.

5.6.2.2 MAMAExpDLL mama_status mamaDQPublisherManager_create (mamaDQPublisherManager *dqPublisher*, mamaTransport *transport*, mamaQueue *queue*, const mamaDQPublisherManagerCallbacks * *callback*, const char * *sourcename*, const char * *root*, void * *closure*)

Send a message from the specified publisher.

Parameters:

publisher The publisher from which to send a message.

msg The mamaMsg to send.

5.6.2.3 MAMAExpDLL void mamaDQPublisherManager_destroy (mamaDQPublisherManager *manager*)

5.6.2.4 MAMAExpDLL mama_status mamaDQPublisherManager_addPublisher (mamaDQPublisherManager *manager*, const char * *symbol*, mamaDQPublisher *pub*, void * *cache*)

5.6.2.5 MAMAExpDLL mama_status mamaDQPublisherManager_removePublisher (mamaDQPublisherManager *manager*, const char * *symbol*, mamaDQPublisher * *pub*)

5.6.2.6 MAMAExpDLL mama_status mamaDQPublisherManager_createPublisher (mamaDQPublisherManager *manager*, const char * *symbol*, void * *cache*, mamaDQPublisher * *newPublisher*)

5.6.2.7 MAMAExpDLL mama_status mamaDQPublisherManager_destroyPublisher (mamaDQPublisherManager *manager*, const char * *symbol*)

5.6.2.8 MAMAExpDLL void mamaDQPublisherManager_setStatus (mamaDQPublisherManager *manager*, mamaMsgStatus *status*)

5.6.2.9 MAMAExpDLL void* mamaDQPublisherManager_getClosure (mamaDQPublisherManager *manager*)

5.6.2.10 MAMAExpDLL void mamaDQPublisherManager_setSenderId (mamaDQPublisherManager *manager*, uint64_t *senderid*)

5.6.2.11 MAMAExpDLL void mamaDQPublisherManager_setSeqNum (mamaDQPublisherManager *manager*, mama_seqnum_t *num*)

5.6.2.12 MAMAExpDLL mama_status mamaDQPublisherManager_sendSyncRequest (mamaDQPublisherManager *manager*, mama_u16_t *nummsg*, mama_f64_t *delay*, mama_f64_t *duration*)

5.6.2.13 MAMAExpDLL mama_status mamaDQPublisherManager_sendNoSubscribers (mamaDQPublisherManager *manager*, const char * *symbol*)

5.6.2.14 MAMAExpDLL void mamaDQPublisherManager_enableSendTime (mamaDQPublisherManager *manager*, mama_bool_t *enable*)

5.7 error.h File Reference

```
#include <mama/config.h>
```

Typedefs

- typedef enum mamaError_ mamaError

Enumerations

- enum mamaError_ {

  MAMA_ERROR_DEFAULT = 0, MAMA_ERROR_INITIAL_TIMEOUT = 1, MAMA_ERROR_RECAP_TIMEOUT = 2, MAMA_ERROR_NOT_-ENTITLED = 3,

  MAMA_ERROR_NO_RESOLVER = 4, MAMA_ERROR_UNKNOWN = 999 }

Functions

- MAMAExpDLL mamaError mamaError_convertFromString (const char *str)

  Convert a string to a mamaError value.

- MAMAExpDLL const char * mamaError_convertToString (mamaError error)

  Convert a mamaError value to a string.

5.7.1 Typedef Documentation

5.7.1.1 typedef enum mamaError_ mamaError

5.7.2 Enumeration Type Documentation

5.7.2.1 enum mamaError_

Enumerator:

MAMA_ERROR_DEFAULT

MAMA_ERROR_INITIAL_TIMEOUT

MAMA_ERROR_RECAP_TIMEOUT

MAMA_ERROR_NOT_ENTITLED

MAMA_ERROR_NO_RESOLVER

MAMA_ERROR_UNKNOWN

```
{
    MAMA_ERROR_DEFAULT         = 0,
    MAMA_ERROR_INITIAL_TIMEOUT = 1,
    MAMA_ERROR_RECAP_TIMEOUT   = 2,
    MAMA_ERROR_NOT_ENTITLED    = 3,
    MAMA_ERROR_NO_RESOLVER     = 4,
    MAMA_ERROR_UNKNOWN         = 999
} mamaError;
```

5.7.3 Function Documentation

5.7.3.1 MAMAExpDLL mamaError mamaError_convertFromString (const char ∗ *str*)

Convert a string to a mamaError value.

Parameters:

str The str to convert.

5.7.3.2 MAMAExpDLL const char∗ mamaError_convertToString (mamaError *error*)

Convert a mamaError value to a string.

Do no attempt to free the string result.

Parameters:

error The mamaError to convert.

5.8 fielddesc.h File Reference

```
#include <mama/status.h>
#include <mama/types.h>
#include <stdlib.h>
#include "wombat/port.h"
```

Typedefs

- typedef enum mamaFieldType_ mamaFieldType

  The mamaFieldDescriptor class represents a field in a mamaMsg.

Enumerations

- enum mamaFieldType_ {

  MAMA_FIELD_TYPE_MSG = 1, MAMA_FIELD_TYPE_OPAQUE = 7, MAMA_FIELD_TYPE_STRING = 8, MAMA_FIELD_TYPE_BOOL = 9,

  MAMA_FIELD_TYPE_CHAR = 10, MAMA_FIELD_TYPE_I8 = 14, MAMA_FIELD_TYPE_U8 = 15, MAMA_FIELD_TYPE_I16 = 16,

  MAMA_FIELD_TYPE_U16 = 17, MAMA_FIELD_TYPE_I32 = 18, MAMA_FIELD_TYPE_U32 = 19, MAMA_FIELD_TYPE_I64 = 20,

  MAMA_FIELD_TYPE_U64 = 21, MAMA_FIELD_TYPE_F32 = 24, MAMA_FIELD_TYPE_F64 = 25, MAMA_FIELD_TYPE_TIME = 26,

  MAMA_FIELD_TYPE_PRICE = 27, MAMA_FIELD_TYPE_VECTOR_I8 = 34, MAMA_FIELD_TYPE_VECTOR_U8 = 35, MAMA_FIELD_TYPE_-VECTOR_I16 = 36,

  MAMA_FIELD_TYPE_VECTOR_U16 = 37, MAMA_FIELD_TYPE_-VECTOR_I32 = 38, MAMA_FIELD_TYPE_VECTOR_U32 = 39, MAMA_-FIELD_TYPE_VECTOR_I64 = 40,

  MAMA_FIELD_TYPE_VECTOR_U64 = 41, MAMA_FIELD_TYPE_-VECTOR_F32 = 44, MAMA_FIELD_TYPE_VECTOR_F64 = 45, MAMA_-FIELD_TYPE_VECTOR_STRING = 46,

  MAMA_FIELD_TYPE_VECTOR_MSG = 47, MAMA_FIELD_TYPE_-VECTOR_TIME = 48, MAMA_FIELD_TYPE_VECTOR_PRICE = 49, MAMA_FIELD_TYPE_QUANTITY = 50,

  MAMA_FIELD_TYPE_COLLECTION = 99, MAMA_FIELD_TYPE_-UNKNOWN = 100 }

  The mamaFieldDescriptor class represents a field in a mamaMsg.

Functions

- MAMAExpDLL const char * mamaFieldTypeToString (mamaFieldType type)

  Return the field type as a string.

- MAMAExpDLL mamaFieldType stringToMamaFieldType (const char *str)

  Return the string as a field type.

- MAMAExpDLL mama_status mamaFieldDescriptor_create (mamaFieldDescriptor *descriptor, mama_fid_t fid, mamaFieldType type, const char *name)

  Create a mamaFieldDescriptor.

- MAMAExpDLL mama_status mamaFieldDescriptor_destroy (mamaFieldDescriptor descriptor)

  Destroy a mamaFieldDescriptor.

- MAMAExpDLL mama_fid_t mamaFieldDescriptor_getFid (const mamaFieldDescriptor descriptor)

  Return the field identifier.

- MAMAExpDLL mamaFieldType mamaFieldDescriptor_getType (const mamaFieldDescriptor descriptor)

  Return the data type.

- MAMAExpDLL const char * mamaFieldDescriptor_getName (const mamaFieldDescriptor descriptor)

  Return the human readable name of the field.

- MAMAExpDLL const char * mamaFieldDescriptor_getTypeName (const mamaFieldDescriptor descriptor)

  return a human readable representation of the type name.

- MAMAExpDLL mama_status mamaFieldDescriptor_setClosure (mamaFieldDescriptor descriptor, void *closure)

  Associate some user supplied data with this field descriptor.

- MAMAExpDLL mama_status mamaFieldDescriptor_getClosure (mamaFieldDescriptor descriptor, void **closure)

  Return the user supplied data with this field descriptor.

5.8.1 Typedef Documentation

5.8.1.1 typedef enum mamaFieldType_ mamaFieldType

The mamaFieldDescriptor class represents a field in a mamaMsg.

5.8.2 Enumeration Type Documentation

5.8.2.1 enum mamaFieldType_

The mamaFieldDescriptor class represents a field in a mamaMsg.

Enumerator:

MAMA_FIELD_TYPE_MSG Sub message.

MAMA_FIELD_TYPE_OPAQUE Opaque binary.

MAMA_FIELD_TYPE_STRING String.

MAMA_FIELD_TYPE_BOOL Boolean.

MAMA_FIELD_TYPE_CHAR Character.

MAMA_FIELD_TYPE_I8 Signed 8 bit integer.

MAMA_FIELD_TYPE_U8 Unsigned byte.

MAMA_FIELD_TYPE_I16 Signed 16 bit integer.

MAMA_FIELD_TYPE_U16 Unsigned 16 bit integer.

MAMA_FIELD_TYPE_I32 Signed 32 bit integer.

MAMA_FIELD_TYPE_U32 Unsigned 32 bit integer.

MAMA_FIELD_TYPE_I64 Signed 64 bit integer.

MAMA_FIELD_TYPE_U64 Unsigned 64 bit integer.

MAMA_FIELD_TYPE_F32 32 bit float

MAMA_FIELD_TYPE_F64 64 bit float

MAMA_FIELD_TYPE_TIME 64 bit MAMA time

MAMA_FIELD_TYPE_PRICE MAMA price.

MAMA_FIELD_TYPE_VECTOR_I8 Array type support.

MAMA_FIELD_TYPE_VECTOR_U8

MAMA_FIELD_TYPE_VECTOR_I16

MAMA_FIELD_TYPE_VECTOR_U16

MAMA_FIELD_TYPE_VECTOR_I32

MAMA_FIELD_TYPE_VECTOR_U32

MAMA_FIELD_TYPE_VECTOR_I64

MAMA_FIELD_TYPE_VECTOR_U64

MAMA_FIELD_TYPE_VECTOR_F32

MAMA_FIELD_TYPE_VECTOR_F64

MAMA_FIELD_TYPE_VECTOR_STRING

MAMA_FIELD_TYPE_VECTOR_MSG

MAMA_FIELD_TYPE_VECTOR_TIME

MAMA_FIELD_TYPE_VECTOR_PRICE

MAMA_FIELD_TYPE_QUANTITY

MAMA_FIELD_TYPE_COLLECTION Collection.

MAMA_FIELD_TYPE_UNKNOWN Unknown.

```
{
    MAMA_FIELD_TYPE_MSG           =    1,

    MAMA_FIELD_TYPE_OPAQUE        =    7,

    MAMA_FIELD_TYPE_STRING        =    8,

    MAMA_FIELD_TYPE_BOOL          =    9,

    MAMA_FIELD_TYPE_CHAR          =   10,

    MAMA_FIELD_TYPE_I8            =   14,

    MAMA_FIELD_TYPE_U8            =   15,

    MAMA_FIELD_TYPE_I16           =   16,

    MAMA_FIELD_TYPE_U16           =   17,

    MAMA_FIELD_TYPE_I32           =   18,

    MAMA_FIELD_TYPE_U32           =   19,

    MAMA_FIELD_TYPE_I64           =   20,

    MAMA_FIELD_TYPE_U64           =   21,

    MAMA_FIELD_TYPE_F32           =   24,

    MAMA_FIELD_TYPE_F64           =   25,

    MAMA_FIELD_TYPE_TIME          =   26,

    MAMA_FIELD_TYPE_PRICE         =   27,

    MAMA_FIELD_TYPE_VECTOR_I8     =   34,
    MAMA_FIELD_TYPE_VECTOR_U8     =   35,
    MAMA_FIELD_TYPE_VECTOR_I16    =   36,
    MAMA_FIELD_TYPE_VECTOR_U16    =   37,
    MAMA_FIELD_TYPE_VECTOR_I32    =   38,
```

```
    MAMA_FIELD_TYPE_VECTOR_U32    =   39,
    MAMA_FIELD_TYPE_VECTOR_I64    =   40,
     MAMA_FIELD_TYPE_VECTOR_U64    =   41,
     MAMA_FIELD_TYPE_VECTOR_F32    =   44,
     MAMA_FIELD_TYPE_VECTOR_F64    =   45,
     MAMA_FIELD_TYPE_VECTOR_STRING =   46,
     MAMA_FIELD_TYPE_VECTOR_MSG    =   47,
     MAMA_FIELD_TYPE_VECTOR_TIME   =   48,
     MAMA_FIELD_TYPE_VECTOR_PRICE  =   49,
     MAMA_FIELD_TYPE_QUANTITY      =   50,

     MAMA_FIELD_TYPE_COLLECTION    =   99,

     MAMA_FIELD_TYPE_UNKNOWN       =  100

} mamaFieldType;
```

5.8.3 Function Documentation

5.8.3.1 MAMAExpDLL const char* mamaFieldTypeToString (mamaFieldType *type*)

Return the field type as a string.

Parameters:

type The type.

Returns:

The type as a string.

5.8.3.2 MAMAExpDLL mamaFieldType stringToMamaFieldType (const char * *str*)

Return the string as a field type.

Parameters:

str The string to transform.

Returns:

The field type.

5.8.3.3 MAMAExpDLL mama_status mamaFieldDescriptor_create (mamaFieldDescriptor * *descriptor*, mama_fid_t *fid*, mamaFieldType *type*, const char * *name*)

Create a mamaFieldDescriptor.

Parameters:

descriptor The descriptor.

fid The field identifier.

type The type.

name The field name.

Returns:

mama_status code

5.8.3.4 MAMAExpDLL mama_status mamaFieldDescriptor_destroy (mamaFieldDescriptor *descriptor*)

Destroy a mamaFieldDescriptor.

Parameters:

descriptor The descriptor.

Returns:

mama_status code

5.8.3.5 MAMAExpDLL mama_fid_t mamaFieldDescriptor_getFid (const mamaFieldDescriptor *descriptor*)

Return the field identifier.

0 indicates no fid.

Parameters:

descriptor The descriptor.

Returns:

The fid.

5.8.3.6 MAMAExpDLL mamaFieldType mamaFieldDescriptor_getType (const mamaFieldDescriptor *descriptor*)

Return the data type.

Returns:

The type.

5.8.3.7 MAMAExpDLL const char* mamaFieldDescriptor_getName (const mamaFieldDescriptor *descriptor*)

Return the human readable name of the field.

Parameters:

descriptor The descriptor.

Returns:

The name.

5.8.3.8 MAMAExpDLL const char* mamaFieldDescriptor_getTypeName (const mamaFieldDescriptor *descriptor*)

return a human readable representation of the type name.

Parameters:

descriptor The descriptor.

5.8.3.9 MAMAExpDLL mama_status mamaFieldDescriptor_setClosure (mamaFieldDescriptor *descriptor*, void * *closure*)

Associate some user supplied data with this field descriptor.

Parameters:

descriptor The field descriptor on which the closure is being set.

closure The arbitrary user supplied data.

Returns:

mama_status MAMA_STATUS_OK if the call was successful.

5.8.3.10 MAMAExpDLL mama_status mamaFieldDescriptor_getClosure (mamaFieldDescriptor *descriptor*, void ∗∗ *closure*)

Return the user supplied data with this field descriptor.

Returns NULL if no data was associated with this field descriptor.

Parameters:

descriptor The field descriptor from which the closure is being obtained.

closure The location to where the user supplied data will be written.

Returns:

mama_status MAMA_STATUS_OK if the call was successful.

5.9 ft.h File Reference

```
#include <mama/mama.h>
```

Typedefs

- typedef enum mamaFtState_ mamaFtState
- typedef enum mamaFtType_ mamaFtType
- typedef void ∗ mamaFtMember
- typedef const char ∗ groupName
- typedef const char mamaFtState state
- typedef const char mamaFtState void ∗ closure

Enumerations

- enum mamaFtState_ { MAMA_FT_STATE_STANDBY = 0, MAMA_FT_-STATE_ACTIVE = 1, MAMA_FT_STATE_UNKNOWN = 99 }
- enum mamaFtType_ { MAMA_FT_TYPE_MULTICAST = 1, MAMA_FT_-TYPE_BRIDGE = 2, MAMA_FT_TYPE_MAX = 3 }

Functions

- typedef void (MAMACALLTYPE ∗mamaFtMemberOnFtStateChange-Cb)(mamaFtMember ftMember

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL mama_status mamaFtMember_create (mamaFtMember ∗member)

  Allocate a MAMA fault tolerance group member.

- MAMAExpDLL mama_status mamaFtMember_setup (mamaFtMember member, mamaFtType fttype, mamaQueue eventQueue, mamaFtMemberOnFtState-ChangeCb callback, mamaTransport transport, const char ∗groupName, mama_-u32_t weight, mama_f64_t heartbeatInterval, mama_f64_t timeoutInterval, void ∗closure)

  Set up a MAMA fault tolerance group member.

- MAMAExpDLL mama_status mamaFtMember_destroy (mamaFtMember member)

  Create a MAMA fault tolerance group member.

- MAMAExpDLL mama_status mamaFtMember_activate (mamaFtMember member)

  Activate the MAMA fault tolerance group member.

- MAMAExpDLL mama_status mamaFtMember_deactivate (mamaFtMember member)

  Deactivate the MAMA fault tolerance group member.

- MAMAExpDLL mama_status mamaFtMember_isActive (const mamaFtMember member, int *result)

  Get whether the MAMA fault tolerance member is actively running (not related to its state).

- MAMAExpDLL mama_status mamaFtMember_getGroupName (const mamaFtMember member, const char **result)

  Get the group name to which this MAMA FT member belongs.

- MAMAExpDLL mama_status mamaFtMember_getWeight (const mamaFtMember member, mama_u32_t *result)

  Get the fault tolerance weight of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_getHeartbeatInterval (const mamaFtMember member, mama_f64_t *result)

  Get the fault tolerance heartbeat interval of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_getTimeoutInterval (const mamaFtMember member, mama_f64_t *result)

  Get the fault tolerance timeout interval of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_getHeartbeatTick (const mamaFtMember member, mama_u32_t *result)

  Get the current heartbeat tick of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_getClosure (const mamaFtMember member, void **result)

  Get the closure argument (provided in the mamaFtMember_create() function) of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_setWeight (mamaFtMember member, mama_u32_t value)

  Set the fault tolerance weight of the MAMA FT member.

- MAMAExpDLL mama_status mamaFtMember_setInstanceId (mamaFtMember member, const char *id)

  Set the instance ID of the MAMA FT member.

- MAMAExpDLL const char * mamaFtStateToString (mamaFtState state)

  Convert a fault tolerant state to a string.

5.9.1 Typedef Documentation

5.9.1.1 typedef enum mamaFtState_ mamaFtState

5.9.1.2 typedef enum mamaFtType_ mamaFtType

5.9.1.3 typedef void* mamaFtMember

5.9.1.4 typedef const char* groupName

5.9.1.5 typedef const char mamaFtState state

5.9.1.6 typedef const char mamaFtState void* closure

5.9.2 Enumeration Type Documentation

5.9.2.1 enum mamaFtState_

Enumerator:

MAMA_FT_STATE_STANDBY

MAMA_FT_STATE_ACTIVE

MAMA_FT_STATE_UNKNOWN

```
{
    MAMA_FT_STATE_STANDBY  = 0,
    MAMA_FT_STATE_ACTIVE   = 1,
    MAMA_FT_STATE_UNKNOWN  = 99
} mamaFtState;
```

5.9.2.2 enum mamaFtType_

Enumerator:

MAMA_FT_TYPE_MULTICAST

MAMA_FT_TYPE_BRIDGE

MAMA_FT_TYPE_MAX

```
{
    MAMA_FT_TYPE_MULTICAST = 1,
    MAMA_FT_TYPE_BRIDGE    = 2,
    MAMA_FT_TYPE_MAX       = 3
} mamaFtType;
```

5.9.3 Function Documentation

5.9.3.1 typedef void (MAMACALLTYPE ∗ *mamaQueueLowWatermarkCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Not currently used. Invoked when an error is encountered during p2p messaging.

Parameters:

status The mama_status describing the error condition.

closure The user supplied data passed to mamaInbox_create()

5.9.3.2 MAMAExpDLL mama_status mamaFtMember_create (mamaFtMember ∗ *member*)

Allocate a MAMA fault tolerance group member.

This function is typically followed by mamaFtMember_setup() and mamaFtMember_-activate().

5.9.3.3 MAMAExpDLL mama_status mamaFtMember_setup (mamaFtMember *member*, mamaFtType *fttype*, mamaQueue *eventQueue*, mamaFtMemberOnFtStateChangeCb *callback*, mamaTransport *transport*, const char ∗ *groupName*, mama_u32_t *weight*, mama_f64_t *heartbeatInterval*, mama_f64_t *timeoutInterval*, void ∗ *closure*)

Set up a MAMA fault tolerance group member.

This is only an initialization function. In order to actually start the fault tolerance monitoring, use mamaFtMember_activate().

5.9.3.4 MAMAExpDLL mama_status mamaFtMember_destroy (mamaFtMember *member*)

Create a MAMA fault tolerance group member.

5.9.3.5 MAMAExpDLL mama_status mamaFtMember_activate (mamaFtMember *member*)

Activate the MAMA fault tolerance group member.

5.9.3.6 MAMAExpDLL mama_status mamaFtMember_deactivate (mamaFtMember *member*)

Deactivate the MAMA fault tolerance group member.

5.9.3.7 MAMAExpDLL mama_status mamaFtMember_isActive (const mamaFtMember *member*, int * *result*)

Get whether the MAMA fault tolerance member is actively running (not related to its state).

Use mamaFtMember_getState() to determine the actual state of the member.

5.9.3.8 MAMAExpDLL mama_status mamaFtMember_getGroupName (const mamaFtMember *member*, const char ** *result*)

Get the group name to which this MAMA FT member belongs.

5.9.3.9 MAMAExpDLL mama_status mamaFtMember_getWeight (const mamaFtMember *member*, mama_u32_t * *result*)

Get the fault tolerance weight of the MAMA FT member.

5.9.3.10 MAMAExpDLL mama_status mamaFtMember_getHeartbeatInterval (const mamaFtMember *member*, mama_f64_t * *result*)

Get the fault tolerance heartbeat interval of the MAMA FT member.

5.9.3.11 MAMAExpDLL mama_status mamaFtMember_getTimeoutInterval (const mamaFtMember *member*, mama_f64_t ∗ *result*)

Get the fault tolerance timeout interval of the MAMA FT member.

5.9.3.12 MAMAExpDLL mama_status mamaFtMember_getHeartbeatTick (const mamaFtMember *member*, mama_u32_t ∗ *result*)

Get the current heartbeat tick of the MAMA FT member.

5.9.3.13 MAMAExpDLL mama_status mamaFtMember_getClosure (const mamaFtMember *member*, void ∗∗ *result*)

Get the closure argument (provided in the mamaFtMember_create() function) of the MAMA FT member.

5.9.3.14 MAMAExpDLL mama_status mamaFtMember_setWeight (mamaFtMember *member*, mama_u32_t *value*)

Set the fault tolerance weight of the MAMA FT member.

The FT weight can be changed dynamically, if desired. The member with the highest weight will become the active member.

5.9.3.15 MAMAExpDLL mama_status mamaFtMember_setInstanceId (mamaFtMember *member*, const char ∗ *id*)

Set the instance ID of the MAMA FT member.

The instance ID is used to uniquely identify members of a fault tolerant group. Most applications should allow the MAMA API to automatically set the instance ID and this function would not be called. If not set explicitly before activation, then the instance ID is automatically set according to the following format: {group-name}.{hex-ip-addr}.{hex-pid}. For example, if the group name is "FOO", the IP address is 192.168.187.9, and the PId is 10777, the default instance ID would be: FOO.c0a8bb9.2a19

5.9.3.16 MAMAExpDLL const char∗ mamaFtStateToString (mamaFtState *state*)

Convert a fault tolerant state to a string.

Parameters:

state The state to convert.

5.10 inbox.h File Reference

Typedefs

- typedef void ∗ closure
- typedef void ∗ closure
- typedef void ∗ closure

Functions

- typedef void (MAMACALLTYPE ∗mamaInboxMsgCallback)(mamaMsg msg

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL mama_status mamaInbox_create (mamaInbox ∗inbox, mama-Transport transport, mamaQueue queue, mamaInboxMsgCallback msgCB, mamaInboxErrorCallback errorCB, void ∗closure)

  Creates an inbox and stores at the address specified by the calling client.

- MAMAExpDLL mama_status mamaInbox_create2 (mamaInbox ∗inbox, mamaTransport transport, mamaQueue queue, mamaInboxMsgCallback msgCB, mamaInboxErrorCallback errorCB, mamaInboxDestroyCallback onInboxDestroyed, void ∗closure)

  Creates an inbox and stores at the address specified by the calling client.

- MAMAExpDLL mama_status mamaInbox_destroy (mamaInbox inbox)

  Destroy the supplied inbox structure.

5.10.1 Typedef Documentation

5.10.1.1 typedef void∗ closure

5.10.1.2 typedef void∗ closure

5.10.1.3 typedef void∗ closure

5.10.2 Function Documentation

5.10.2.1 typedef void (MAMACALLTYPE ∗ *mamaInboxDestroyCallback*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Parameters:

msg The mamaMsg received in the p2p response.

closure The user supplied data passed to mamaInbox_create()

5.10.2.2 MAMAExpDLL mama_status mamaInbox_create (mamaInbox ∗ *inbox*, mamaTransport *transport*, mamaQueue *queue*, mamaInboxMsgCallback *msgCB*, mamaInboxErrorCallback *errorCB*, void ∗ *closure*)

Creates an inbox and stores at the address specified by the calling client.

Parameters:

inbox Pointer to the inbox which will be created

transport The mamaTransport being used.

queue The mamaQueue to use.

msgCB Invoked for any point to point responses.

errorCB For future use. Not currently used.

closure User supplied data to be passed back in callbacks.

Returns:

mama_status MAMA_STATUS_OK if the function is successful.

5.10.2.3 MAMAExpDLL mama_status mamaInbox_create2 (mamaInbox * *inbox*, mamaTransport *transport*, mamaQueue *queue*, mamaInboxMsgCallback *msgCB*, mamaInboxErrorCallback *errorCB*, mamaInboxDestroyCallback *onInboxDestroyed*, void * *closure*)

Creates an inbox and stores at the address specified by the calling client.

Parameters:

inbox Pointer to the inbox which will be created

transport The mamaTransport being used.

queue The mamaQueue to use.

msgCB Invoked for any point to point responses.

errorCB For future use. Not currently used.

closure User supplied data to be passed back in callbacks.

Returns:

mama_status MAMA_STATUS_OK if the function is successful.

5.10.2.4 MAMAExpDLL mama_status mamaInbox_destroy (mamaInbox *inbox*)

Destroy the supplied inbox structure.

Note that this function is asynchronous and is only guaranteed to have finished whenever the onInboxDestroyed function passed to the mamaInbox_create2 has been called.

Parameters:

inbox The mamaInbox to be destroyed.

Returns:

mama_Status MAMA_STATUS_OK if function call successful.

5.11 io.h File Reference

```
#include <mama/types.h>
#include <mama/status.h>
```

Typedefs

- typedef mamaIoType ioType
- typedef mamaIoType void ∗ closure

Enumerations

- enum mamaIoType {

  MAMA_IO_READ, MAMA_IO_WRITE, MAMA_IO_CONNECT, MAMA_-IO_ACCEPT,

  MAMA_IO_CLOSE, MAMA_IO_ERROR, MAMA_IO_EXCEPT }

  IO Types.

Functions

- typedef void (MAMACALLTYPE ∗mamaIoCb)(mamaIo io

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL mama_status mamaIo_create (mamaIo ∗result, mamaQueue queue, uint32_t descriptor, mamaIoCb action, mamaIoType ioType, void ∗closure)

  Create a IO handler.

- MAMAExpDLL mama_status mamaIo_getDescriptor (mamaIo io, uint32_-t ∗d)

  Get the descriptor.

- MAMAExpDLL mama_status mamaIo_destroy (mamaIo io)

  Destroy the IO.

5.11.1 Typedef Documentation

5.11.1.1 typedef mamaIoType ioType

5.11.1.2 typedef mamaIoType void* closure

5.11.2 Enumeration Type Documentation

5.11.2.1 enum mamaIoType

IO Types.

Not all implementation support all mamaIoTypes.

MAMA_IO_READ: the socket is readable. MAMA_IO_WRITE: the socket is writable. MAMA_IO_CONNECT: the socket is connected MAMA_IO_ACCEPT: the socket accepted a connection MAMA_IO_CLOSE: the socket was closed MAMA_-IO_ERROR: an error occurred MAMA_IO_EXCEPT: An exceptional event like out of band data occurred.

Enumerator:

MAMA_IO_READ

MAMA_IO_WRITE

MAMA_IO_CONNECT

MAMA_IO_ACCEPT

MAMA_IO_CLOSE

MAMA_IO_ERROR

MAMA_IO_EXCEPT

```
{
    MAMA_IO_READ,
    MAMA_IO_WRITE,
    MAMA_IO_CONNECT,
    MAMA_IO_ACCEPT,
    MAMA_IO_CLOSE,
    MAMA_IO_ERROR,
    MAMA_IO_EXCEPT
} mamaIoType;
```

5.11.3 Function Documentation

5.11.3.1 typedef void (MAMACALLTYPE * *mamaIoCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Parameters:

io The mamaIo handle.

ioType The mamaIoType for the event.

closure Caller supplied closure.

5.11.3.2 MAMAExpDLL mama_status mamaIo_create (mamaIo ∗ *result*, mamaQueue *queue*, uint32_t *descriptor*, mamaIoCb *action*, mamaIoType *ioType*, void ∗ *closure*)

Create a IO handler.

If the underlying infrastructure does not support the requested mamaIoType, mamaIo_-create returns MAMA_STATUS_UNSUPPORTED_IO_TYPE. For example RV only supports READ, WRITE, and EXCEPT. LBM supports all types except ERROR.

Parameters:

result A pointer to the io handle.

queue The event queue for the io events. NULL specifies the Mama default queue.

action The callback to be invoked when an event occurs.

descriptor Wait for IO on this descriptor.

ioType Wait for occurrences of this type.

closure The closure that is passed to the callback.

5.11.3.3 MAMAExpDLL mama_status mamaIo_getDescriptor (mamaIo *io*, uint32_t ∗ *d*)

Get the descriptor.

5.11.3.4 MAMAExpDLL mama_status mamaIo_destroy (mamaIo *io*)

Destroy the IO.

5.12 log.h File Reference

```
#include <stdio.h>
#include <mama/config.h>
#include <stdarg.h>
#include "mama/status.h"
```

Typedefs

- typedef const char ∗ format
- typedef const char va_list ap
- typedef const char ∗ message
- typedef void(∗) logSizeCbType (void)

Enumerations

- enum MamaLogLevel {

  MAMA_LOG_LEVEL_OFF = 0, MAMA_LOG_LEVEL_SEVERE = 1, MAMA_LOG_LEVEL_ERROR = 2, MAMA_LOG_LEVEL_WARN = 3,

  MAMA_LOG_LEVEL_NORMAL = 4, MAMA_LOG_LEVEL_FINE = 5, MAMA_LOG_LEVEL_FINER = 6, MAMA_LOG_LEVEL_FINEST = 7 }

  The level of detail when logging is enabled within the API.

- enum mamaLogFilePolicy { LOGFILE_UNBOUNDED = 1, LOGFILE_ROLL = 2, LOGFILE_OVERWRITE = 3, LOGFILE_USER = 4 }

  The policy to control log file size.

Functions

- typedef void (MAMACALLTYPE ∗mamaLogCb)(MamaLogLevel level

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL void MAMACALLTYPE mama_logDefault (MamaLogLevel level, const char ∗format, va_list ap)

  The default logging within the API unless otherwise specified.

- MAMAExpDLL void MAMACALLTYPE mama_logDefault2 (MamaLogLevel level, const char *message)

  This second logging function takes only a message and not a format string with a variable argument list.

- MAMAExpDLL void mama_logStdout (MamaLogLevel level, const char *format,...)
- MAMAExpDLL void MAMACALLTYPE mama_forceLogDefault (MamaLogLevel level, const char *format, va_list ap)

  The default function used within the API for the mama_forceLog function pointer.

- MAMAExpDLL mama_status mama_enableLogging (FILE *file, MamaLogLevel level)

  Enable logging.

- MAMAExpDLL mama_status mama_logToFile (const char *file, MamaLogLevel level)

  Behaves as mama_enableLogging() but accepts a string representing the file location.

- MAMAExpDLL mama_status mama_disableLogging (void)

  Disable logging.

- MAMAExpDLL void mama_log (MamaLogLevel level, const char *format,...)

  Used for the majority of logging within the API.

- MAMAExpDLL void MAMACALLTYPE mama_log2 (MamaLogLevel level, const char *message)
- MAMAExpDLL void mama_logVa (MamaLogLevel level, const char *format, va_list args)
- MAMAExpDLL void mama_forceLogVa (const char *format, va_list args)

  Used for the force logging using variable argument parameters.

- MAMAExpDLL void mama_forceLog (MamaLogLevel level, const char *format,...)
- MAMAExpDLL mama_status mama_setLogCallback (mamaLogCb callback)

  Set the callback to be used for mama_log calls.

- MAMAExpDLL mama_status mama_setLogCallback2 (mamaLogCb2 callback)

  Set the callback to be used for mama_log calls.

- MAMAExpDLL mama_status mama_setForceLogCallback (mamaLogCb callback)

Set the callback to be used for mama logging.

- MAMAExpDLL mama_status mama_setLogLevel (MamaLogLevel level)

  Sets the log level for Mama.

- MAMAExpDLL MamaLogLevel mama_getLogLevel (void)

  Returns the current log level for Mama.

- MAMAExpDLL mama_status mama_setLogSize (unsigned long size)

  Set the maxmum size of the log file (bytes).

- MAMAExpDLL mama_status mama_setNumLogFiles (int numFiles)

  Set the number of rolled logfiles to keep before overwriting.

- MAMAExpDLL mama_status mama_setLogFilePolicy (mamaLogFilePolicy policy)

  Set the policy regarding how to handle files when Max file size is reached.

- MAMAExpDLL mama_status mama_setAppendToLogFile (int append)

  Set append to existing log file.

- MAMAExpDLL int mama_loggingToFile (void)

  Return status of loggingToFile.

- MAMAExpDLL mama_status mama_setLogSizeCb (logSizeCbType logCallbacks)

  Set a callback for when the max log size is reached.

- MAMAExpDLL const char * mama_logLevelToString (MamaLogLevel level)

  Return string version of log level.

- MAMAExpDLL int mama_tryStringToLogLevel (const char *s, MamaLogLevel *level)

  Try to convert string to log level Return non-zero for success, zero for failure The string comparison is case insensitive.

- MAMAExpDLL const char * mama_logPolicyToString (mamaLogFilePolicy level)

  Return string version of log policy.

- MAMAExpDLL int mama_tryStringToLogPolicy (const char *s, mamaLogFilePolicy *policy)

  Try to convert string to log policy.

- MAMAExpDLL int mama_logIncrementVerbosity (MamaLogLevel *level)
  Increase by one log level the verbosity of a MamaLogLevel variable.
- MAMAExpDLL int mama_logDecrementVerbosity (MamaLogLevel *level)
  Decrease by one log level the verbosity of a MamaLogLevel variable.
- MAMAExpDLL mama_status mama_logForceRollLogFiles (void)
  Force rolling the log file.
- void mama_logDestroy (void)
  Destroy memory held by the logging.
- void mama_loginit (void)

Variables

- MAMAExpDLL MamaLogLevel gMamaLogLevel
  The current log level within the API.
- MAMAExpDLL FILE * gMamaLogFile
  The file to which all logging will be written by default.

5.12.1 Typedef Documentation

5.12.1.1 typedef const char* format

5.12.1.2 typedef const char va_list ap

5.12.1.3 typedef const char* message

5.12.1.4 typedef void(*) logSizeCbType(void)

5.12.2 Enumeration Type Documentation

5.12.2.1 enum MamaLogLevel

The level of detail when logging is enabled within the API.

Enumerator:

MAMA_LOG_LEVEL_OFF

MAMA_LOG_LEVEL_SEVERE

MAMA_LOG_LEVEL_ERROR

MAMA_LOG_LEVEL_WARN

MAMA_LOG_LEVEL_NORMAL

MAMA_LOG_LEVEL_FINE

MAMA_LOG_LEVEL_FINER

MAMA_LOG_LEVEL_FINEST

```
{
    MAMA_LOG_LEVEL_OFF    = 0,
    MAMA_LOG_LEVEL_SEVERE = 1,
    MAMA_LOG_LEVEL_ERROR  = 2,
    MAMA_LOG_LEVEL_WARN   = 3,
    MAMA_LOG_LEVEL_NORMAL = 4,
    MAMA_LOG_LEVEL_FINE   = 5,
    MAMA_LOG_LEVEL_FINER  = 6,
    MAMA_LOG_LEVEL_FINEST = 7
} MamaLogLevel;
```

5.12.2.2 enum mamaLogFilePolicy

The policy to control log file size.

Enumerator:

LOGFILE_UNBOUNDED

LOGFILE_ROLL

LOGFILE_OVERWRITE

LOGFILE_USER

```
{
    LOGFILE_UNBOUNDED   = 1,
    LOGFILE_ROLL        = 2,
    LOGFILE_OVERWRITE   = 3,
    LOGFILE_USER        = 4
} mamaLogFilePolicy;
```

5.12.3 Function Documentation

5.12.3.1 typedef void (MAMACALLTYPE ∗ *mamaLogCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Parameters:

io The mamaIo handle.

ioType The mamaIoType for the event.

closure Caller supplied closure.

5.12.3.2 MAMAExpDLL void MAMACALLTYPE mama_logDefault (MamaLogLevel *level*, const char * *format*, va_list *ap*)

The default logging within the API unless otherwise specified.

5.12.3.3 MAMAExpDLL void MAMACALLTYPE mama_logDefault2 (MamaLogLevel *level*, const char * *message*)

This second logging function takes only a message and not a format string with a variable argument list.

It is required for interoperability with all platforms that do not support C variable argument list, (e.g. .Net). Other than that it performs in exactly the same way as the first.

Parameters:

level (in) The log level.

message (in) The message to log.

5.12.3.4 MAMAExpDLL void mama_logStdout (MamaLogLevel *level*, const char * *format*, ...)

5.12.3.5 MAMAExpDLL void MAMACALLTYPE mama_forceLogDefault (MamaLogLevel *level*, const char * *format*, va_list *ap*)

The default function used within the API for the mama_forceLog function pointer.

5.12.3.6 MAMAExpDLL mama_status mama_enableLogging (FILE * *file*, MamaLogLevel *level*)

Enable logging.

No per-message or per-tick messages appear at `WOMBAT_LOG_LEVEL_FINE`. `WOMBAT_LOG_LEVEL_FINER` and `WOMBAT_LOG_LEVEL_FINEST` provide successively more detailed logging.

Parameters:

file File to write to.

level Output level.

See also:

Level

5.12.3.7 MAMAExpDLL mama_status mama_logToFile (const char * *file*, MamaLogLevel *level*)

Behaves as mama_enableLogging() but accepts a string representing the file location.

Parameters:

file The path to the file. Can be relative, absolute or on $WOMBAT_PATH.

level The level at which the API should log messages.

5.12.3.8 MAMAExpDLL mama_status mama_disableLogging (void)

Disable logging.

5.12.3.9 MAMAExpDLL void mama_log (MamaLogLevel *level*, const char * *format*, ...)

Used for the majority of logging within the API.

5.12.3.10 MAMAExpDLL void MAMACALLTYPE mama_log2 (MamaLogLevel *level*, const char * *message*)

5.12.3.11 MAMAExpDLL void mama_logVa (MamaLogLevel *level*, const char * *format*, va_list *args*)

5.12.3.12 MAMAExpDLL void mama_forceLogVa (const char * *format*, va_list *args*)

Used for the force logging using variable argument parameters.

5.12.3.13 MAMAExpDLL void mama_forceLog (MamaLogLevel *level*, const char * *format*, ...)

5.12.3.14 MAMAExpDLL mama_status mama_setLogCallback (mamaLogCb *callback*)

Set the callback to be used for mama_log calls.

If not set then mama_logDefault will be used

5.12.3.15 MAMAExpDLL mama_status mama_setLogCallback2 (mamaLogCb2 *callback*)

Set the callback to be used for mama_log calls.

This function will set a log callback that receives a formatted string and not a variable argument list. This function is used mainly to support managed clients.

Parameters:

callback (in) The callback to be used. Pass NULL to restore the mama_logDefault function.

Returns:

MAMA_STATUS_OK

5.12.3.16 MAMAExpDLL mama_status mama_setForceLogCallback (mamaLogCb *callback*)

Set the callback to be used for mama logging.

If not set then mama_ForceLogDefault will be used

5.12.3.17 MAMAExpDLL mama_status mama_setLogLevel (MamaLogLevel *level*)

Sets the log level for Mama.

5.12.3.18 MAMAExpDLL MamaLogLevel mama_getLogLevel (void)

Returns the current log level for Mama.

5.12.3.19 MAMAExpDLL mama_status mama_setLogSize (unsigned long *size*)

Set the maxmum size of the log file (bytes).

When this size is reached the logsize callback is called, or if no callback is set then the default action is to overwrite file from the start. Default max size is 500 Mb

5.12.3.20 MAMAExpDLL mama_status mama_setNumLogFiles (int *numFiles*)

Set the number of rolled logfiles to keep before overwriting.

Default is 10

5.12.3.21 MAMAExpDLL mama_status mama_setLogFilePolicy (mamaLogFilePolicy *policy*)

Set the policy regarding how to handle files when Max file size is reached.

Default is LOGFILE_UNBOUNDED

5.12.3.22 MAMAExpDLL mama_status mama_setAppendToLogFile (int *append*)

Set append to existing log file.

5.12.3.23 MAMAExpDLL int mama_loggingToFile (void)

Return status of loggingToFile.

5.12.3.24 MAMAExpDLL mama_status mama_setLogSizeCb (logSizeCbType *logCallbacks*)

Set a callback for when the max log size is reached.

This can be used to override the default action which is to wrap the file and continue logging at the beginning

5.12.3.25 MAMAExpDLL const char* mama_logLevelToString (MamaLogLevel *level*)

Return string version of log level.

5.12.3.26 MAMAExpDLL int mama_tryStringToLogLevel (const char * *s*, MamaLogLevel * *level*)

Try to convert string to log level Return non-zero for success, zero for failure The string comparison is case insensitive.

5.12.3.27 MAMAExpDLL const char* mama_logPolicyToString (mamaLogFilePolicy *level*)

Return string version of log policy.

5.12.3.28 MAMAExpDLL int mama_tryStringToLogPolicy (const char * *s*, mamaLogFilePolicy * *policy*)

Try to convert string to log policy.

Return non-zero for success, zero for failure. The string comparison is case insensitive.

5.12.3.29 MAMAExpDLL int mama_logIncrementVerbosity (MamaLogLevel * *level*)

Increase by one log level the verbosity of a MamaLogLevel variable.

If the level is already at the maximum verbosity it will be unchanged after calling the function, otherwise the level will be incremented. Returns zero if level is not changed, or non-zero if it is changed If an unrecognized level is passed, the function will return non-zero and the variable will be set to the minimum verbosity

5.12.3.30 MAMAExpDLL int mama_logDecrementVerbosity (MamaLogLevel * *level*)

Decrease by one log level the verbosity of a MamaLogLevel variable.

If the level is already at the minimum verbosity it will be unchanged after calling the function, otherwise the level will be decremented. Returns zero if level is not changed, or non-zero if it is changed If an unrecognized level is passed, the function will return non-zero and the variable will be set to the maximum verbosity

5.12.3.31 MAMAExpDLL mama_status mama_logForceRollLogFiles (void)

Force rolling the log file.

Returns:

The status of the operation.

5.12.3.32 void mama_logDestroy (void)

Destroy memory held by the logging.

5.12.3.33 void mama_loginit (void)

5.12.4 Variable Documentation

5.12.4.1 MAMAExpDLL MamaLogLevel gMamaLogLevel

The current log level within the API.

5.12.4.2 MAMAExpDLL FILE* gMamaLogFile

The file to which all logging will be written by default.

5.13 mama.h File Reference

```
#include "mama/config.h"
#include <mama/log.h>
#include <mama/error.h>
#include <mama/types.h>
#include <mama/datetime.h>
#include <mama/servicelevel.h>
#include <mama/status.h>
#include <mama/msgstatus.h>
#include <mama/msgtype.h>
#include <mama/msgfield.h>
#include <mama/dictionary.h>
#include <mama/transport.h>
#include <mama/msg.h>
#include <mama/middleware.h>
#include <mama/price.h>
#include <mama/publisher.h>
#include <mama/reservedfields.h>
#include <mama/inbox.h>
#include <mama/timer.h>
#include <mama/queue.h>
#include <mama/senderId.h>
#include <mama/symbolmap.h>
#include <mama/symbolmapfile.h>
#include <mama/timezone.h>
#include <mama/source.h>
#include <mama/subscriptiontype.h>
#include <mama/quality.h>
#include <mama/ft.h>
```

Defines

- #define MAMA_OPEN_MD ((uint32_t)(0x00000001))
- #define MAMA_OPEN_PUB_SUB ((uint32_t)(0x00000002))
- #define MAMA_DEFAULT_THROTTLE_RATE 500
- #define MAMA_DEFAULT_RECAP_THROTTLE_RATE 250
- #define MAMA_DEFAULT_RETRIES 2
- #define MAMA_DEFAULT_TIMEOUT 30.0f
- #define MAMA_MAX_SYMBOL_LEN 128
- #define MAMA_MAX_SOURCE_LEN 64
- #define MAMA_MAX_TRANSPORT_LEN 64
- #define MAMA_LINK_BRIDGE(implIdentifier, impl)
- #define MAMA_CREATE_BRIDGE(implIdentifier, impl)

Typedefs

- typedef void ∗MAMAExpDLL mama_status mama_startBackgroundEx (mamaBridge bridgeImpl, mamaStopCBEx callback, void ∗closure)

  Start Mama in the background, with extended parameters.

- typedef const char ∗ message

Functions

- mama_status mama_setDefaultPayload (char id)

  The `Mama` *data type provides methods global initialization and manipulating global options.*

- MAMAExpDLL mama_status mama_loadBridge (mamaBridge ∗impl, const char ∗middleware)

  Load the bridge specified by middleware string.

- MAMAExpDLL mama_status mama_loadPayloadBridge (mamaPayloadBridge ∗bridge, const char ∗payloadName)
- MAMAExpDLL mama_status mama_loadBridgeWithPath (mamaBridge ∗impl, const char ∗middleware, const char ∗path)

  Load the bridge specified by middleware string using the path specified by the user.

- MAMAExpDLL mama_status mama_open (void)

  mama_status mama_open (void)

- MAMAExpDLL mama_status mama_openWithProperties (const char ∗path, const char ∗filename)

Initialize MAMA.

- MAMAExpDLL mama_status mama_setProperty (const char *name, const char *value)

  Set a specific property for the API.

- MAMAExpDLL mama_status mama_setPropertiesFromFile (const char *path, const char *filename)

  Load a set of properties through the API.

- MAMAExpDLL const char * mama_getProperty (const char *name)

  Retrieve a specific property from the API.

- MAMAExpDLL mama_status mama_close (void)

  Close MAMA and free all associated resource.

- MAMAExpDLL const char * mama_getVersion (mamaBridge bridgeImpl)

  Return the version information for the library.

- MAMAExpDLL mama_status mama_start (mamaBridge bridgeImpl)

  Start processing messages on the internal queue.

- typedef void (MAMACALLTYPE *mamaStartCB)(mama_status status)

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

- MAMAExpDLL mama_status mama_startBackground (mamaBridge bridgeImpl, mamaStartCB callback)

  Start Mama in the background.

- MAMAExpDLL mama_status mama_stop (mamaBridge bridgeImpl)

  Stop dispatching on the default event queue for the specified bridge.

- MAMAExpDLL mama_status mama_stopAll (void)

  Stop dispatching on the default event queue for all bridges.

- MAMAExpDLL mama_status mama_setApplicationName (const char *applicationName)

  mama_setApplicationName - sets the mama application name This should be called before mama_open

- MAMAExpDLL mama_status mama_setApplicationClassName (const char *className)

mama_setApplicationClass - sets the mama class name This should be called before mama_open

- MAMAExpDLL mama_status mama_getApplicationName (const char **applicationName)

  mama_getApplicationName - gets the mama application name

- MAMAExpDLL mama_status mama_getApplicationClassName (const char **className)

  mama_getApplicationClass - sets the mama class name

- MAMAExpDLL mama_status mama_getUserName (const char **userName)

  mama_getUserName - gets the user name

- MAMAExpDLL mama_status mama_getHostName (const char **hostName)

  mama_getHostName - gets the host name

- MAMAExpDLL mama_status mama_getIpAddress (const char **ipAddress)

  mama_getIpAddressName - gets the IP Address

- MAMAExpDLL mama_status mama_getDefaultEventQueue (mamaBridge bridgeImpl, mamaQueue *defaultQueue)

  Get a reference to the internal default event queue in use for the specified middleware.

- MAMAExpDLL void mama_setLastError (mamaError error)

  Set the last error to occur in Mama.

- MAMAExpDLL mamaError mama_getLastErrorCode (void)

  Get the code of the last error to have occurred in Mama.

- MAMAExpDLL const char * mama_getLastErrorText (void)

  Get the text of the last error to have occurred in Mama.

- MAMAExpDLL mama_status mama_setBridgeInfoCallback (mamaBridge bridgeImpl, bridgeInfoCallback callback)

  Sets a callback to be invoked whenever an information message is logged at the bridge level.

- MAMAExpDLL mama_status mama_addStatsCollector (mamaStatsCollector statsCollector)

  Add a user stats collector.

- MAMAExpDLL mama_status mama_removeStatsCollector (mamaStatsCollector statsCollector)

Remove a user stats collector.

Variables

- typedef mamaBridge

5.13.1 Define Documentation

5.13.1.1 #define MAMA_OPEN_MD ((uint32_t)(0x00000001))

5.13.1.2 #define MAMA_OPEN_PUB_SUB ((uint32_t)(0x00000002))

5.13.1.3 #define MAMA_DEFAULT_THROTTLE_RATE 500

5.13.1.4 #define MAMA_DEFAULT_RECAP_THROTTLE_RATE 250

5.13.1.5 #define MAMA_DEFAULT_RETRIES 2

5.13.1.6 #define MAMA_DEFAULT_TIMEOUT 30.0f

5.13.1.7 #define MAMA_MAX_SYMBOL_LEN 128

5.13.1.8 #define MAMA_MAX_SOURCE_LEN 64

5.13.1.9 #define MAMA_MAX_TRANSPORT_LEN 64

5.13.1.10 #define MAMA_LINK_BRIDGE(implIdentifier, impl)

Value:

```
do                                                                  \
    {                                                                   \
        implIdentifier ## Bridge_createImpl (impl);                     \
    }                                                                   \
    while(0)                                                            \
```

5.13.1.11 #define MAMA_CREATE_BRIDGE(implIdentifier, impl)

Value:

```
do                                                                  \
    {                                                                   \
```

```
        if (MAMA_LINK_##implIdentifier)                                 \
        {                                                               \
            implIdentifier##Bridge_createImpl (impl);                   \
        }                                                               \
        else                                                            \
        {                                                               \
            mama_loadBridge (impl, "##implIdentifier##");               \
        }                                                               \
    }                                                                   \
    while(0)                                                            \
```

5.13.2 Typedef Documentation

5.13.2.1 typedef void* MAMAExpDLL mama_status mama_startBackgroundEx(mamaBridge bridgeImpl, mamaStopCBEx callback, void *closure)

Start Mama in the background, with extended parameters.

This method performs the same functionality as mama_startBackground accept it provides the facility to pass in a closure. The C++ wrapper layer uses this version of the function, and stores the MamaStartCallback object in the closure.

Parameters:

← ***bridgeImpl*** The bridge specific structure.

← ***callback*** The extended callback for asynchronous status.

Returns:

MAMA_STATUS_OK if successful.

5.13.2.2 typedef const char* message

5.13.3 Function Documentation

5.13.3.1 mama_status mama_setDefaultPayload (char *id*)

The `Mama` data type provides methods global initialization and manipulating global options.

Related `Mama` functions also provide means for creating transports, dictionaries, and subscriptions.

5.13.3.2 MAMAExpDLL mama_status mama_loadBridge (mamaBridge * *impl*, const char * *middleware*)

Load the bridge specified by middleware string.

If the bridge has already been loaded then the existing bridge instance will be returned.

Parameters:

impl The bridge object

middleware The middleware string. Can be "wmw", "lbm" or "tibrv".

Returns:

mama_status Whether the call was successful or not.

5.13.3.3 MAMAExpDLL mama_status mama_loadPayloadBridge (mamaPayloadBridge * *bridge*, const char * *payloadName*)

5.13.3.4 MAMAExpDLL mama_status mama_loadBridgeWithPath (mamaBridge * *impl*, const char * *middleware*, const char * *path*)

Load the bridge specified by middleware string using the path specified by the user.

If the bridge has already been loaded then the existing bridge instance will be returned.

Parameters:

impl The bridge object

middleware The middleware string. Can be "wmw", "lbm" or "tibrv".

path The path to the bridge library

Returns:

mama_status Whether the call was successful or not.

5.13.3.5 MAMAExpDLL mama_status mama_open (void)

mama_status mama_open (void)

Initialize MAMA.

MAMA employs a reference count to track multiple calls to mama_open() and mama_close(). The count is incremented every time mama_open() is called and decremented when mama_close() is called. The resources are not actually released until the count reaches zero.

5.13.3.6 MAMAExpDLL mama_status mama_openWithProperties (const char * *path*, const char * *filename*)

Initialize MAMA.

Allows users of the API to override the default behavior of mama_open() where a file mama.properties is required to be located in the directory specified by $WOMBAT_-PATH.

The properties file must have the same structure as a standard mama.properties file.

If null is passed as the path the API will look for the properties file on the $WOMBAT_-PATH.

If null is passed as the filename the API will look for the default filename of mama.properties.

Parameters:

path Fully qualified path to the directory containing the properties file

filename The name of the file containing MAMA properties.

Returns:

mama_status Whether the call was successful or not.

5.13.3.7 MAMAExpDLL mama_status mama_setProperty (const char * *name*, const char * *value*)

Set a specific property for the API.

If the property being set has already been given a value from a properties file that value will be replaced.

See the example mama.properties provided with the distribution for examples of property formatting. The properties set via this function should be formatted in the same manner as those specified in mama.properties.

The strings passed to the function are copied.

Parameters:

name The name of the property being set.

value The value of the property being set.

Returns:

MAMA_STATUS_NULL_ARG is either name or value is null. MAMA_-STATUS_OK if the function completed successfully.

5.13.3.8 MAMAExpDLL mama_status mama_setPropertiesFromFile (const char * *path*, const char * *filename*)

Load a set of properties through the API.

If the property being set has already been given a value from a properties file that value will be replaced.

The properties file must have the same structure as a standard mama.properties file.

If null is passed as the path the API will look for the properties file on the $WOMBAT_-PATH.

Parameters:

path Fully qualified path to the directory containing the properties file

filename The name of the file containing properties.

Returns:

MAMA_STATUS_NULL_ARG is either name or value is null. MAMA_-STATUS_OK if the function completed successfully.

5.13.3.9 MAMAExpDLL const char* mama_getProperty (const char * *name*)

Retrieve a specific property from the API.

If the property has not been set, a NULL value will be returned.

Parameters:

name The name of the property to retrieve.

Returns:

the value of the property or NULL if unset.

5.13.3.10 MAMAExpDLL mama_status mama_close (void)

Close MAMA and free all associated resource.

5.13.3.11 MAMAExpDLL const char* mama_getVersion (mamaBridge *bridgeImpl*)

Return the version information for the library.

The version of the underlying transport follows in parens.

Parameters:

bridgeImpl The bridge specific structure.

5.13.3.12 MAMAExpDLL mama_status mama_start (mamaBridge *bridgeImpl*)

Start processing messages on the internal queue.

This starts Mama's internal throttle, refresh logic, and other internal Mama processes as well as dispatching messages from the internal queue.

mama_start( ) blocks until an invocation of mama_stop() occurs.

Parameters:

bridgeImpl The bridge specific structure.

5.13.3.13 typedef void (MAMACALLTYPE * *mamaQueueHighWatermark-ExceededCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Parameters:

queue The mamaQueue for which the size limit has been exceeded. NULL if the queue is the default internal MAMA queue.

size The current number of events outstanding on the queue (if supported on the underlying middleware)

closure User supplied data set when the callback was registered. NULL in the case of the default MAMA queue as no closure can be specified when registering the data quality callbacks.

5.13.3.14 MAMAExpDLL mama_status mama_startBackground (mamaBridge *bridgeImpl*, mamaStartCB *callback*)

Start Mama in the background.

This method invokes mama_start() in a separate thread.

This API uses the deprecated mamaStartCB callback type. mama_startBackgroundEx uses the replacement type mamaStopCBEx. To retain backward compatability mama_-startBackground casts callback to mamaStopCB for further processing.

Parameters:

bridgeImpl The bridge specific structure.

callback The callback for asynchronous status.

Returns:

MAMA_STATUS_OK if successful.

5.13.3.15 MAMAExpDLL mama_status mama_stop (mamaBridge *bridgeImpl*)

Stop dispatching on the default event queue for the specified bridge.

Parameters:

bridgeImpl The bridge specific structure.

5.13.3.16 MAMAExpDLL mama_status mama_stopAll (void)

Stop dispatching on the default event queue for all bridges.

5.13.3.17 MAMAExpDLL mama_status mama_setApplicationName (const char * *applicationName*)

mama_setApplicationName - sets the mama application name This should be called before mama_open

Parameters:

applicationName

5.13.3.18 MAMAExpDLL mama_status mama_setApplicationClassName (const char * *className*)

mama_setApplicationClass - sets the mama class name This should be called before mama_open

Parameters:

className

5.13.3.19 MAMAExpDLL mama_status mama_getApplicationName (const char ** *applicationName*)

mama_getApplicationName - gets the mama application name

Parameters:

applicationName address of where to put applicationName

5.13.3.20 MAMAExpDLL mama_status mama_getApplicationClassName (const char ** *className*)

mama_getApplicationClass - sets the mama class name

Parameters:

className address of where to put className

5.13.3.21 MAMAExpDLL mama_status mama_getUserName (const char ** *userName*)

mama_getUserName - gets the user name

Parameters:

userName address of where to put user name

5.13.3.22 MAMAExpDLL mama_status mama_getHostName (const char ** *hostName*)

mama_getHostName - gets the host name

Parameters:

hostName address of where to put host name

5.13.3.23 MAMAExpDLL mama_status mama_getIpAddress (const char ** *ipAddress*)

mama_getIpAddressName - gets the IP Address

Parameters:

ipAddress address of where to put IP address

5.13.3.24 MAMAExpDLL mama_status mama_getDefaultEventQueue (mamaBridge *bridgeImpl*, mamaQueue * *defaultQueue*)

Get a reference to the internal default event queue in use for the specified middleware.

Parameters:

bridgeImpl The middleware for which the default event queue is being obtained.

defaultQueue Address to which the defaultQueue is to be written.

Returns:

MAMA_STATUS_OK if the function returns successfully.

5.13.3.25 MAMAExpDLL void mama_setLastError (mamaError *error*)

Set the last error to occur in Mama.

Each thread will have its own last error.

This function is for internal usage only.

Parameters:

error The code of the last error to have occured in this thread.

5.13.3.26 MAMAExpDLL mamaError mama_getLastErrorCode (void)

Get the code of the last error to have occurred in Mama.

Each thread will have its own last error.

Parameters:

error The code of the last error to have occured in this thread.

5.13.3.27 MAMAExpDLL const char* mama_getLastErrorText (void)

Get the text of the last error to have occurred in Mama.

Each thread will have its own last error.

Parameters:

error The text of the last error to have occured in this thread.

5.13.3.28 MAMAExpDLL mama_status mama_setBridgeInfoCallback (mamaBridge *bridgeImpl*, bridgeInfoCallback *callback*)

Sets a callback to be invoked whenever an information message is logged at the bridge level.

Messages returned vary depending on the underlying middleware. Currently only implemented for LBM bridges.

5.13.3.29 MAMAExpDLL mama_status mama_addStatsCollector (mamaStatsCollector *statsCollector*)

Add a user stats collector.

5.13.3.30 MAMAExpDLL mama_status mama_removeStatsCollector (mamaStatsCollector *statsCollector*)

Remove a user stats collector.

5.13.4 Variable Documentation

5.13.4.1 typedef mamaBridge

5.14 marketdata.h File Reference

```
#include "mama/config.h"
#include "mama/types.h"
#include "mama/msgtype.h"
```

Typedefs

- typedef enum mamaMdDataType mamaDataType

Enumerations

- enum mamaMdDataType {

  MAMA_MD_DATA_TYPE_STANDARD = 0, MAMA_MD_DATA_TYPE_-ORDER_BOOK = 1, MAMA_MD_DATA_TYPE_NEWS_STORY = 2, MAMA_MD_DATA_TYPE_WORLDVIEW = 3,

  MAMA_MD_DATA_TYPE_PROPERTY = 4, MAMA_MD_DATA_TYPE_-USAGE_LOG = 5, MAMA_MD_DATA_TYPE_NEWS_QUERY = 6, MAMA_MD_DATA_TYPE_TEMPLATE = 7 }
- enum mamaMdMsgType {

  MAMA_MD_MSG_TYPE_GENERAL = 0, MAMA_MD_MSG_TYPE_-CANCEL = 2, MAMA_MD_MSG_TYPE_ERROR = 3, MAMA_MD_MSG_-TYPE_CORRECTION = 4,

  MAMA_MD_MSG_TYPE_CLOSING = 5, MAMA_MD_MSG_TYPE_-SYMBOL_DELETE = 7, MAMA_MD_MSG_TYPE_SYMBOL_ACTION = 8, MAMA_MD_MSG_TYPE_PREOPENING = 12,

  MAMA_MD_MSG_TYPE_QUOTE = 13, MAMA_MD_MSG_TYPE_-TRADE = 14, MAMA_MD_MSG_TYPE_BOOK_UPDATE = 15, MAMA_-MD_MSG_TYPE_BOOK_INITIAL = 16,

  MAMA_MD_MSG_TYPE_BOOK_CLEAR = 18, MAMA_MD_MSG_-TYPE_IMBALANCE = 22, MAMA_MD_MSG_TYPE_SECURITY_STATUS = 23, MAMA_MD_MSG_TYPE_NEWS_HEADLINE = 24,

  MAMA_MD_MSG_TYPE_NEWS_STORY = 25, MAMA_MD_MSG_-TYPE_NEWS_QUERY = 26, MAMA_MD_MSG_TYPE_MISC = 100 }

Functions

- MAMAExpDLL mamaMdMsgType mamaMdMsgType_typeForMsg (const mamaMsg msg)

  Extract the market data message type from the message.

- MAMAExpDLL const char * mamaMdMsgType_stringForMsg (const mamaMsg msg)

  Extract the market data message type as a string from a message.

- MAMAExpDLL const char * mamaMdMsgType_stringForType (mamaMdMsgType type)

  Convert a mamaMdMsgType to a string.

- MAMAExpDLL mamaMsgType mamaMdMsgType_compatMsgType (mamaMdMsgType type)

  Convert a mamaMdMsgType to a reasonable backward-compatible mamaMsgType.

5.14.1 Typedef Documentation

5.14.1.1 typedef enum mamaMdDataType mamaDataType

5.14.2 Enumeration Type Documentation

5.14.2.1 enum mamaMdDataType

Enumerator:

MAMA_MD_DATA_TYPE_STANDARD

MAMA_MD_DATA_TYPE_ORDER_BOOK

MAMA_MD_DATA_TYPE_NEWS_STORY

MAMA_MD_DATA_TYPE_WORLDVIEW

MAMA_MD_DATA_TYPE_PROPERTY

MAMA_MD_DATA_TYPE_USAGE_LOG

MAMA_MD_DATA_TYPE_NEWS_QUERY

MAMA_MD_DATA_TYPE_TEMPLATE

```
{
    MAMA_MD_DATA_TYPE_STANDARD   = 0,
    MAMA_MD_DATA_TYPE_ORDER_BOOK = 1,
    MAMA_MD_DATA_TYPE_NEWS_STORY = 2,
    MAMA_MD_DATA_TYPE_WORLDVIEW  = 3,
    MAMA_MD_DATA_TYPE_PROPERTY   = 4,
    MAMA_MD_DATA_TYPE_USAGE_LOG  = 5,
    MAMA_MD_DATA_TYPE_NEWS_QUERY = 6,
    MAMA_MD_DATA_TYPE_TEMPLATE   = 7
} mamaDataType;
```

5.14.2.2 enum mamaMdMsgType

Enumerator:

MAMA_MD_MSG_TYPE_GENERAL General update (indices, funds).

MAMA_MD_MSG_TYPE_CANCEL Trade cancellation.

MAMA_MD_MSG_TYPE_ERROR Trade error.

MAMA_MD_MSG_TYPE_CORRECTION Trade correction.

MAMA_MD_MSG_TYPE_CLOSING Closing summary.

MAMA_MD_MSG_TYPE_SYMBOL_DELETE Symbol deleted.

MAMA_MD_MSG_TYPE_SYMBOL_ACTION Action related to this symbol, such as a name change or symbol deletion due to option/future expiration, etc.

MAMA_MD_MSG_TYPE_PREOPENING Pre-opening summary (e.g. morning "roll")

MAMA_MD_MSG_TYPE_QUOTE Quote update.

MAMA_MD_MSG_TYPE_TRADE Trade update.

MAMA_MD_MSG_TYPE_BOOK_UPDATE Orderbook update.

MAMA_MD_MSG_TYPE_BOOK_INITIAL Order book initial value. This message is sent rather than MAMA_MD_MSG_TYPE_INITIAL for order books.

MAMA_MD_MSG_TYPE_BOOK_CLEAR Orderbook clear.

MAMA_MD_MSG_TYPE_IMBALANCE Order imbalance or noimbalance update.

MAMA_MD_MSG_TYPE_SECURITY_STATUS Security status update.

MAMA_MD_MSG_TYPE_NEWS_HEADLINE News headline.

MAMA_MD_MSG_TYPE_NEWS_STORY News Story.

MAMA_MD_MSG_TYPE_NEWS_QUERY News query.

MAMA_MD_MSG_TYPE_MISC Miscellaneous.

```
53 {
55     MAMA_MD_MSG_TYPE_GENERAL            = 0,
56
58     MAMA_MD_MSG_TYPE_CANCEL             = 2,
59
61     MAMA_MD_MSG_TYPE_ERROR              = 3,
62
64     MAMA_MD_MSG_TYPE_CORRECTION         = 4,
65
67     MAMA_MD_MSG_TYPE_CLOSING            = 5,
68
70     MAMA_MD_MSG_TYPE_SYMBOL_DELETE      = 7,
71
```

```
    MAMA_MD_MSG_TYPE_SYMBOL_ACTION      = 8,

    MAMA_MD_MSG_TYPE_PREOPENING         = 12,

    MAMA_MD_MSG_TYPE_QUOTE              = 13,

    MAMA_MD_MSG_TYPE_TRADE              = 14,

    MAMA_MD_MSG_TYPE_BOOK_UPDATE        = 15,

    MAMA_MD_MSG_TYPE_BOOK_INITIAL       = 16,

    MAMA_MD_MSG_TYPE_BOOK_CLEAR         = 18,

    MAMA_MD_MSG_TYPE_IMBALANCE          = 22,

     MAMA_MD_MSG_TYPE_SECURITY_STATUS    = 23,

     MAMA_MD_MSG_TYPE_NEWS_HEADLINE      = 24,

     MAMA_MD_MSG_TYPE_NEWS_STORY         = 25,

     MAMA_MD_MSG_TYPE_NEWS_QUERY         = 26,

     MAMA_MD_MSG_TYPE_MISC               = 100

} mamaMdMsgType;
```

5.14.3 Function Documentation

5.14.3.1 MAMAExpDLL mamaMdMsgType mamaMdMsgType_typeForMsg (const mamaMsg *msg*)

Extract the market data message type from the message.

Parameters:

msg The message.

5.14.3.2 MAMAExpDLL const char* mamaMdMsgType_stringForMsg (const mamaMsg *msg*)

Extract the market data message type as a string from a message.

Parameters:

msg The message.

5.14.3.3 MAMAExpDLL const char* mamaMdMsgType_stringForType (mamaMdMsgType *type*)

Convert a mamaMdMsgType to a string.

Parameters:

type The mamaMdMsgType.

5.14.3.4 MAMAExpDLL mamaMsgType mamaMdMsgType_compatMsgType (mamaMdMsgType *type*)

Convert a mamaMdMsgType to a reasonable backward-compatible mamaMsgType.

Parameters:

type The mamaMdMsgType.

5.15 middleware.h File Reference

```
#include <mama/config.h>
```

Typedefs

- typedef enum mamaMiddleware_ mamaMiddleware

  Enum to reference the available MAMA middlewares.

Enumerations

- enum mamaMiddleware_ {

  MAMA_MIDDLEWARE_WMW = 0, MAMA_MIDDLEWARE_LBM = 1, MAMA_MIDDLEWARE_TIBRV = 2, MAMA_MIDDLEWARE_AVIS = 3,

  MAMA_MIDDLEWARE_TICK42BLP = 4, MAMA_MIDDLEWARE_-SOLACE = 5, MAMA_MIDDLEWARE_RAI = 6, MAMA_MIDDLEWARE_-QPID = 7,

  MAMA_MIDDLEWARE_EXEGY = 8, MAMA_MIDDLEWARE_MAX = 9, MAMA_MIDDLEWARE_UNKNOWN = 99 }

  Enum to reference the available MAMA middlewares.

Functions

- MAMAExpDLL mamaMiddleware mamaMiddleware_convertFromString (const char *str)

  Convert a string to a mamaMiddleware value.

- MAMAExpDLL const char * mamaMiddleware_convertToString (mamaMiddleware middleware)

  Convert a mamaMiddleware value to a string.

5.15.1 Typedef Documentation

5.15.1.1 typedef enum mamaMiddleware_ mamaMiddleware

Enum to reference the available MAMA middlewares.

MAMA_MIDDLEWARE_MAX will be incremented when new middlewares become available

5.15.2 Enumeration Type Documentation

5.15.2.1 enum mamaMiddleware_

Enum to reference the available MAMA middlewares.

MAMA_MIDDLEWARE_MAX will be incremented when new middlewares become available

Enumerator:

MAMA_MIDDLEWARE_WMW
MAMA_MIDDLEWARE_LBM
MAMA_MIDDLEWARE_TIBRV
MAMA_MIDDLEWARE_AVIS
MAMA_MIDDLEWARE_TICK42BLP
MAMA_MIDDLEWARE_SOLACE
MAMA_MIDDLEWARE_RAI
MAMA_MIDDLEWARE_QPID
MAMA_MIDDLEWARE_EXEGY
MAMA_MIDDLEWARE_MAX
MAMA_MIDDLEWARE_UNKNOWN

```
{
    MAMA_MIDDLEWARE_WMW     = 0,
    MAMA_MIDDLEWARE_LBM     = 1,
    MAMA_MIDDLEWARE_TIBRV   = 2,
    MAMA_MIDDLEWARE_AVIS    = 3,
    MAMA_MIDDLEWARE_TICK42BLP = 4,
    MAMA_MIDDLEWARE_SOLACE  = 5,
    MAMA_MIDDLEWARE_RAI     = 6,
    MAMA_MIDDLEWARE_QPID    = 7,
    MAMA_MIDDLEWARE_EXEGY   = 8,
    MAMA_MIDDLEWARE_MAX     = 9,
    MAMA_MIDDLEWARE_UNKNOWN = 99
} mamaMiddleware;
```

5.15.3 Function Documentation

5.15.3.1 MAMAExpDLL mamaMiddleware mamaMiddleware_convertFromString (const char * *str*)

Convert a string to a mamaMiddleware value.

Parameters:

str The str to convert.

5.15.3.2 MAMAExpDLL const char* mamaMiddleware_convertToString (mamaMiddleware *middleware*)

Convert a mamaMiddleware value to a string.

Do no attempt to free the string result.

Parameters:

middleware The mamaMiddleware to convert.

5.16 msg.h File Reference

```
#include <mama/types.h>
#include <mama/status.h>
#include <mama/fielddesc.h>
#include <mama/msgtype.h>
#include "wombat/port.h"
#include <stdlib.h>
```

Defines

- #define ENTITLE_FIELD_NAME "wEntitleCode"
- #define ENTITLE_FIELD_ID 496

Typedefs

- typedef enum mamaPayloadType_ mamaPayloadType

  Enum to reference the available MAMA message payloads.

- typedef void(MAMACALLTYPE ∗) mamaMsgIteratorCb (const mamaMsg msg, const mamaMsgField field, void ∗closure)

  Iterator callback method is invoked for each field in the message.

Enumerations

- enum mamaPayloadType_ {

  MAMA_PAYLOAD_WOMBAT_MSG = 'W', MAMA_PAYLOAD_TIBRV = 'R', MAMA_PAYLOAD_FAST = 'F', MAMA_PAYLOAD_V5 = '5',

  MAMA_PAYLOAD_AVIS = 'A', MAMA_PAYLOAD_TICK42BLP = 'B', MAMA_PAYLOAD_RAI = 'I', MAMA_PAYLOAD_EXEGY = 'X',

  MAMA_PAYLOAD_UNKNOWN = 'U' }

  Enum to reference the available MAMA message payloads.

Functions

- MAMAExpDLL const char ∗ mamaPayload_convertToString (mamaPayloadType payloadType)

Convert a mamaPayloadType value to a string.

- MAMAExpDLL mama_status mamaMsg_create (mamaMsg *msg)

  Create a mamaMsg.

- MAMAExpDLL mama_status mamaMsg_createForPayload (mamaMsg *msg, const char id)

  Create a mamaMsg.

- MAMAExpDLL mama_status mamaMsg_createForPayloadBridge (mamaMsg *msg, mamaPayloadBridge payloadBridge)

  Create a mamaMsg.

- MAMAExpDLL mama_status mamaMsg_createForTemplate (mamaMsg *msg, mama_u32_t templateId)

  Create a mamaMsg for a particular template ID.

- MAMAExpDLL mama_status mamaMsg_copy (mamaMsg src, mamaMsg *copy)

  Copy a mamaMsg.

- MAMAExpDLL mama_status mamaMsg_getTempCopy (mamaMsg src, mamaMsg *copy)

  Get a temporary copy of the mamaMsg so to be able to modify the content.

- MAMAExpDLL mama_status mamaMsg_clear (mamaMsg msg)

  Clear a msg.

- MAMAExpDLL mama_status mamaMsg_getSendSubject (const mamaMsg msg, const char **subject)

  TIB/RV only.

- MAMAExpDLL mama_status mamaMsg_destroy (mamaMsg msg)

  Destroy a message and free any resources associated with it.

- MAMAExpDLL mama_status mamaMsg_getPayloadType (mamaMsg msg, mamaPayloadType *payloadType)

  Get the type of the message payload.

- MAMAExpDLL mama_status mamaMsg_getByteSize (const mamaMsg msg, mama_size_t *size)

  Get the size of the underlying message in bytes.

- MAMAExpDLL mama_status mamaMsg_addBool (mamaMsg msg, const char *name, mama_fid_t fid, mama_bool_t value)

  Add a new bool field.

- MAMAExpDLL mama_status mamaMsg_addChar (mamaMsg msg, const char *name, mama_fid_t fid, char value)

  Add a new char field.

- MAMAExpDLL mama_status mamaMsg_addI8 (mamaMsg msg, const char *name, mama_fid_t fid, mama_i8_t value)

  Add a new 8 bit signed int field.

- MAMAExpDLL mama_status mamaMsg_addU8 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u8_t value)

  Add a new 8 bit unsigned int field.

- MAMAExpDLL mama_status mamaMsg_addI16 (mamaMsg msg, const char *name, mama_fid_t fid, mama_i16_t value)

  Add a new 16 bit signed int field.

- MAMAExpDLL mama_status mamaMsg_addU16 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u16_t value)

  Add a new 16 bit unsigned int field.

- MAMAExpDLL mama_status mamaMsg_addI32 (mamaMsg msg, const char *name, mama_fid_t fid, mama_i32_t value)

  Add a new 32 bit unsigned int field.

- MAMAExpDLL mama_status mamaMsg_addU32 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u32_t value)

  Add a new 32 bit signed int field.

- MAMAExpDLL mama_status mamaMsg_addI64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_i64_t value)

  Add a new I64 field.

- MAMAExpDLL mama_status mamaMsg_addU64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u64_t value)

  Add a new U64 field.

- MAMAExpDLL mama_status mamaMsg_addF32 (mamaMsg msg, const char *name, mama_fid_t fid, mama_f32_t value)

  Add a new float (f32) field.

- MAMAExpDLL mama_status mamaMsg_addF64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_f64_t value)

  Add a new f64 field.

- MAMAExpDLL mama_status mamaMsg_addString (mamaMsg msg, const char *name, mama_fid_t fid, const char *value)

  *Add a const char * field.*

- MAMAExpDLL mama_status mamaMsg_addOpaque (mamaMsg msg, const char *name, mama_fid_t fid, const void *value, mama_size_t size)

  Add an opaque field.

- MAMAExpDLL mama_status mamaMsg_addDateTime (mamaMsg msg, const char *name, mama_fid_t fid, const mamaDateTime value)

  Add a MAMA date/time field.

- MAMAExpDLL mama_status mamaMsg_addPrice (mamaMsg msg, const char *name, mama_fid_t fid, const mamaPrice value)

  Add a MAMA price field.

- MAMAExpDLL mama_status mamaMsg_addMsg (mamaMsg msg, const char *name, mama_fid_t fid, const mamaMsg value)

  Add a mamaMsg object to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorBool (mamaMsg msg, const char *name, mama_fid_t fid, const mama_bool_t value[ ], mama_size_t numElements)

  Add an array of booleans to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorChar (mamaMsg msg, const char *name, mama_fid_t fid, const char value[ ], mama_size_t numElements)

  Add an array of characters to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorI8 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_i8_t value[ ], mama_size_t numElements)

  Add an array of signed 8 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorU8 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_u8_t value[ ], mama_size_t numElements)

Add an array of unsigned 8 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorI16 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_i16_t value[ ], mama_size_t numElements)

  Add an array of signed 16 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorU16 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_u16_t value[ ], mama_size_t numElements)

  Add an array of unsigned 16 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorI32 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_i32_t value[ ], mama_size_t numElements)

  Add an array of signed 32 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorU32 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_u32_t value[ ], mama_size_t numElements)

  Add an array of unsigned 32 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorI64 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_i64_t value[ ], mama_size_t numElements)

  Add an array of signed 64 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorU64 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_u64_t value[ ], mama_size_t numElements)

  Add an array of unsigned 64 bit integers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorF32 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_f32_t value[ ], mama_size_t numElements)

  Add an array of 32 bit floating point numbers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorF64 (mamaMsg msg, const char *name, mama_fid_t fid, const mama_f64_t value[ ], mama_size_t numElements)

  Add an array of 64 bit floating point numbers to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorString (mamaMsg msg, const char *name, mama_fid_t fid, const char *value[ ], mama_size_t numElements)

Add an array of strings (char∗) to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorMsg (mamaMsg msg, const char ∗name, mama_fid_t fid, const mamaMsg value[ ], mama_size_t numElements)

  Add an array of mamaMsg objects to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorDateTime (mamaMsg msg, const char ∗name, mama_fid_t fid, const mamaDateTime value[ ], mama_size_t numElements)

  Add an array of timestamps to the message.

- MAMAExpDLL mama_status mamaMsg_addVectorPrice (mamaMsg msg, const char ∗name, mama_fid_t fid, const mamaPrice value[ ], mama_size_t numElements)

  Add an array of prices to the message.

- MAMAExpDLL mama_status mamaMsg_updateBool (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_bool_t value)

  Update the value of an existing bool field.

- MAMAExpDLL mama_status mamaMsg_updateChar (mamaMsg msg, const char ∗name, mama_fid_t fid, char value)

  Update the value of an existing char field.

- MAMAExpDLL mama_status mamaMsg_updateI8 (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_i8_t value)

  Update the value of an existing I8 field.

- MAMAExpDLL mama_status mamaMsg_updateU8 (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_u8_t value)

  Update the value of an existing U8 field.

- MAMAExpDLL mama_status mamaMsg_updateI16 (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_i16_t value)

  Update the value of an existing I16 field.

- MAMAExpDLL mama_status mamaMsg_updateU16 (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_u16_t value)

  Update the value of an existing U16 field.

- MAMAExpDLL mama_status mamaMsg_updateI32 (mamaMsg msg, const char ∗name, mama_fid_t fid, mama_i32_t value)

Update the value of an existing I32 field.

- MAMAExpDLL mama_status mamaMsg_updateU32 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u32_t value)

  Update the value of an existing U32 field.

- MAMAExpDLL mama_status mamaMsg_updateI64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_i64_t value)

  Update the value of an existing I64 field.

- MAMAExpDLL mama_status mamaMsg_updateU64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_u64_t value)

  Update the value of an existing U64 field.

- MAMAExpDLL mama_status mamaMsg_updateF32 (mamaMsg msg, const char *name, mama_fid_t fid, mama_f32_t value)

  Update the value of an existing float field.

- MAMAExpDLL mama_status mamaMsg_updateF64 (mamaMsg msg, const char *name, mama_fid_t fid, mama_f64_t value)

  Update the value of an existing f64 field.

- MAMAExpDLL mama_status mamaMsg_updateString (mamaMsg msg, const char *name, mama_fid_t fid, const char *value)

  Update the value of an existing const char field.*

- MAMAExpDLL mama_status mamaMsg_updateOpaque (mamaMsg msg, const char *name, mama_fid_t fid, const void *value, mama_size_t size)

  Update an opaque field.

- MAMAExpDLL mama_status mamaMsg_updateDateTime (mamaMsg msg, const char *name, mama_fid_t fid, const mamaDateTime value)

  Update a MAMA date/time field.

- MAMAExpDLL mama_status mamaMsg_updatePrice (mamaMsg msg, const char *name, mama_fid_t fid, const mamaPrice value)

  Update a MAMA price field.

- MAMAExpDLL mama_status mamaMsg_getBool (const mamaMsg msg, const char *name, mama_fid_t fid, mama_bool_t *result)

  Get a bool field.

- MAMAExpDLL mama_status mamaMsg_applyMsg (mamaMsg dest, mamaMsg src)

Apply the contents of one message to another as updates.

- MAMAExpDLL mama_status mamaMsg_getNativeMsg (mamaMsg msg, void **nativeMsg)

  Get the native message structure for the underlying message This function is for internal NYSE Technologies use only.

- MAMAExpDLL mama_status mamaMsg_getReplyHandle (mamaMsg msg, mamaMsgReply *replyHandle)

  Get the reply handle for the message and increments the reference count for the handle.

- MAMAExpDLL mama_status mamaMsg_setReplyHandle (mamaMsg msg, mamaMsgReply replyHandle)

  Set the reply handle for the message and increments the reference count for the handle.

- MAMAExpDLL mama_status mamaMsg_copyReplyHandle (mamaMsgReply src, mamaMsgReply *copy)

  Create a new copy of the src reply handle.

- MAMAExpDLL mama_status mamaMsg_destroyReplyHandle (mamaMsgReply replyHandle)

  Destroys a reply handle.

- MAMAExpDLL mama_status mamaMsg_updateSubMsg (mamaMsg msg, const char *fname, mama_fid_t fid, const mamaMsg subMsg)

  Update a sub-message field.

- MAMAExpDLL mama_status mamaMsg_updateVectorMsg (mamaMsg msg, const char *fname, mama_fid_t fid, const mamaMsg msgList[ ], mama_size_t numElements)

  Update a vector message field.

- MAMAExpDLL mama_status mamaMsg_updateVectorString (mamaMsg msg, const char *fname, mama_fid_t fid, const char *strList[ ], mama_size_t numElements)

  Update a vector string field.

- MAMAExpDLL mama_status mamaMsg_updateVectorBool (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_bool_t boolList[ ], mama_size_t numElements)

  Update a vector bool field.

- MAMAExpDLL mama_status mamaMsg_updateVectorChar (mamaMsg msg, const char *fname, mama_fid_t fid, const char charList[ ], mama_size_t numElements)

  Update a vector char field.

- MAMAExpDLL mama_status mamaMsg_updateVectorI8 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_i8_t i8List[ ], mama_size_t numElements)

  Update a vector I8 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorU8 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_u8_t u8List[ ], mama_size_t numElements)

  Update a vector U8 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorI16 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_i16_t i16List[ ], mama_size_t numElements)

  Update a vector I16 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorU16 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_u16_t u16List[ ], mama_size_t numElements)

  Update a vector U16 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorI32 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_i32_t i32List[ ], mama_size_t numElements)

  Update a vector I32 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorU32 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_u32_t u32List[ ], mama_size_t numElements)

  Update a vector U32 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorI64 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_i64_t i64List[ ], mama_size_t numElements)

  Update a vector I64 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorU64 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_u64_t u64List[ ], mama_size_t numElements)

  Update a vector U64 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorF32 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_f32_t f32List[ ], mama_size_t numElements)

  Update a vector F32 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorF64 (mamaMsg msg, const char *fname, mama_fid_t fid, const mama_f64_t f64List[ ], mama_size_t numElements)

  Update a vector F64 field.

- MAMAExpDLL mama_status mamaMsg_updateVectorPrice (mamaMsg msg, const char *fname, mama_fid_t fid, const mamaPrice *priceList[ ], mama_size_t numElements)

  Update a vector price field.

- MAMAExpDLL mama_status mamaMsg_updateVectorTime (mamaMsg msg, const char *fname, mama_fid_t fid, const mamaDateTime timeList[ ], mama_size_t numElements)

  Update a vector mamaDateTime field.

- MAMAExpDLL mama_status mamaMsg_getChar (const mamaMsg msg, const char *name, mama_fid_t fid, char *result)

  Get a char field.

- MAMAExpDLL mama_status mamaMsg_getI8 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_i8_t *result)

  Get a I8, signed 8 bit integer, field.

- MAMAExpDLL mama_status mamaMsg_getU8 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_u8_t *result)

  Get an unsigned 8-bit integer field.

- MAMAExpDLL mama_status mamaMsg_getI16 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_i16_t *result)

  Get a I16, signed 16 bit integer, field.

- MAMAExpDLL mama_status mamaMsg_getU16 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_u16_t *result)

  Get an unsigned 16-bit integer field.

- MAMAExpDLL mama_status mamaMsg_getI32 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_i32_t *result)

Get a I32, signed 32 bit integer, field.

- MAMAExpDLL mama_status mamaMsg_getU32 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_u32_t *result)

  Get an unsigned 32-bit integer field.

- MAMAExpDLL mama_status mamaMsg_getI64 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_i64_t *result)

  Get a I64, signed 64 bit integer, field.

- MAMAExpDLL mama_status mamaMsg_getU64 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_u64_t *result)

  Get an unsigned 64-bit integer field.

- MAMAExpDLL mama_status mamaMsg_getF32 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_f32_t *result)

  Get a float (f32) field.

- MAMAExpDLL mama_status mamaMsg_getF64 (const mamaMsg msg, const char *name, mama_fid_t fid, mama_f64_t *result)

  Get a f64 field.

- MAMAExpDLL mama_status mamaMsg_getString (const mamaMsg msg, const char *name, mama_fid_t fid, const char **result)

  *Get a const char * field.*

- MAMAExpDLL mama_status mamaMsg_getOpaque (const mamaMsg msg, const char *name, mama_fid_t fid, const void **result, mama_size_t *size)

  Get an opaque field.

- MAMAExpDLL mama_status mamaMsg_getField (const mamaMsg msg, const char *name, mama_fid_t fid, mamaMsgField *result)

  Get a MAMA msg field.

- MAMAExpDLL mama_status mamaMsg_getDateTime (const mamaMsg msg, const char *name, mama_fid_t fid, mamaDateTime result)

  Get a MAMA date/time field.

- MAMAExpDLL mama_status mamaMsg_getDateTimeMSec (const mamaMsg msg, const char *name, mama_fid_t fid, mama_u64_t *milliseconds)

  Get the value of a MAMA date/time field in milliseconds.

- MAMAExpDLL mama_status mamaMsg_getPrice (const mamaMsg msg, const char *name, mama_fid_t fid, mamaPrice result)

Get a MAMA price field.

- MAMAExpDLL mama_status mamaMsg_getMsg (const mamaMsg msg, const char *name, mama_fid_t fid, mamaMsg *result)

  Get a submessage.

- MAMAExpDLL mama_status mamaMsg_getVectorBool (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_bool_t **result, mama_size_t *resultLen)

  Get a vector of booleans.

- MAMAExpDLL mama_status mamaMsg_getVectorChar (const mamaMsg msg, const char *name, mama_fid_t fid, const char **result, mama_size_t *resultLen)

  Get a vector of chars.

- MAMAExpDLL mama_status mamaMsg_getVectorI8 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_i8_t **result, mama_size_t *resultLen)

  Get a vector of signed 8 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorU8 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_u8_t **result, mama_size_t *resultLen)

  Get a vector of unsigned 8 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorI16 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_i16_t **result, mama_size_t *resultLen)

  Get a vector of signed 16 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorU16 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_u16_t **result, mama_size_t *resultLen)

  Get a vector of unsigned 16 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorI32 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_i32_t **result, mama_size_t *resultLen)

  Get a vector of signed 32 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorU32 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_u32_t **result, mama_size_t *resultLen)

Get a vector of unsigned 32 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorI64 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_i64_t **result, mama_size_t *resultLen)

  Get a vector of signed 64 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorU64 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_u64_t **result, mama_size_t *resultLen)

  Get a vector of unsigned 64 bit integers.

- MAMAExpDLL mama_status mamaMsg_getVectorF32 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_f32_t **result, mama_size_t *resultLen)

  Get a vector of 32 bit floating point numbers.

- MAMAExpDLL mama_status mamaMsg_getVectorF64 (const mamaMsg msg, const char *name, mama_fid_t fid, const mama_f64_t **result, mama_size_t *resultLen)

  Get a vector of 64 bit floating point numbers.

- MAMAExpDLL mama_status mamaMsg_getVectorString (const mamaMsg msg, const char *name, mama_fid_t fid, const char ***result, mama_size_t *resultLen)

  Get a vector of strings (char).*

- MAMAExpDLL mama_status mamaMsg_getVectorDateTime (const mamaMsg msg, const char *name, mama_fid_t fid, const mamaDateTime *result, mama_size_t *resultLen)

  Get a vector of timestamps.

- MAMAExpDLL mama_status mamaMsg_getVectorPrice (const mamaMsg msg, const char *name, mama_fid_t fid, const mamaPrice *result, mama_size_t *resultLen)

  Get a vector of prices.

- MAMAExpDLL mama_status mamaMsg_getVectorMsg (const mamaMsg msg, const char *name, mama_fid_t fid, const mamaMsg **result, mama_size_t *resultLen)

  Get a vector of submessages.

- MAMAExpDLL mama_status mamaMsg_getNumFields (const mamaMsg msg, mama_size_t *numFields)

Returns the total number of fields in the message.

- MAMAExpDLL const char * mamaMsg_toString (const mamaMsg msg)

  *Return a const char * representation the message.*

- MAMAExpDLL void mamaMsg_freeString (const mamaMsg msg, const char *msgString)

  Free the memory allocated by mamaMsg_toString.

- MAMAExpDLL mama_status mamaMsg_getEntitleCode (const mamaMsg msg, mama_i32_t *code)

  Get the entitle code for this message.

- MAMAExpDLL mama_status mamaMsg_getSeqNum (const mamaMsg msg, mama_seqnum_t *seqNum)

  Get the sequence number for this message if available.

- MAMAExpDLL mamaMsgType mamaMsgType_typeForMsg (const mamaMsg msg)

  Extract the type from the supplied message.

- MAMAExpDLL const char * mamaMsgType_stringForMsg (const mamaMsg msg)

  Return the type name.

- MAMAExpDLL const char * mamaMsgType_stringForType (const mamaMsgType type)

  Return the type name.

- MAMAExpDLL mama_status mamaMsg_getFieldAsString (const mamaMsg msg, const char *fieldName, mama_fid_t fid, char *buf, mama_size_t length)

  Convert the value of the specified field to a string.

- MAMAExpDLL mama_status mamaMsg_iterateFields (const mamaMsg msg, mamaMsgIteratorCb callback, const mamaDictionary dict, void *closure)

  Invoke the specified callback for each field in the message.

- MAMAExpDLL mama_status mamaMsg_getPlatformError (const mamaMsg msg, void **error)

  Return the middleware specific platform error.

- MAMAExpDLL int mamaMsg_isFromInbox (const mamaMsg msg)

  Return true if this message was sent from a client's inbox.

- MAMAExpDLL mama_status mamaMsg_createFromByteBuffer (mamaMsg *msg, const void *buffer, mama_size_t bufferLength)

  Create a mamaMsg from the provided byte buffer.

- MAMAExpDLL mama_status mamaMsg_getByteBuffer (const mamaMsg msg, const void **buffer, mama_size_t *bufferLength)

  Return the internal message buffer as an array of bytes which is suitable for writing to a file.

- MAMAExpDLL mama_status mamaMsg_detach (mamaMsg msg)

  Normally the Mama API owns incoming mamaMsg objects and they go out of scope when the message callback returns.

- MAMAExpDLL mama_status mamaMsg_getIsDefinitelyDuplicate (mamaMsg msg, int *result)

  Return true if this message is definitely a duplicate message.

- MAMAExpDLL mama_status mamaMsg_getIsPossiblyDuplicate (mamaMsg msg, int *result)

  Return true if this message is possibly a duplicate message.

- MAMAExpDLL mama_status mamaMsg_getIsPossiblyDelayed (mamaMsg msg, int *result)

  Return true if the message is possibly delayed.

- MAMAExpDLL mama_status mamaMsg_getIsDefinitelyDelayed (mamaMsg msg, int *result)

  Return true if the message is delayed.

- MAMAExpDLL mama_status mamaMsg_getIsOutOfSequence (mamaMsg msg, int *result)

  Return true when the FH sends the message out of sequence.

- MAMAExpDLL mama_status mamaMsg_setNewBuffer (mamaMsg msg, void *buffer, mama_size_t size)

  Set a new buffer for an existing mamaMsg.

- MAMAExpDLL mama_status mamaMsg_getNativeHandle (const mamaMsg msg, void **result)

  Get the native middleware message handle.

- MAMAExpDLL mama_status mamaMsgIterator_create (mamaMsgIterator *iterator, mamaDictionary dict)

Creates a new iterator for use with a mamaMsg.

- MAMAExpDLL mama_status mamaMsgIterator_associate (mamaMsgIterator iterator, mamaMsg msg)

  Associate an iterator for use with a mamaMsg.

- MAMAExpDLL mama_status mamaMsgIterator_setDict (mamaMsgIterator iterator, mamaDictionary dict)

  Associate a mamaDictionary for use with an iterator.

- MAMAExpDLL mamaMsgField mamaMsgIterator_next (mamaMsgIterator iterator)

  Return the next field in the mamaMsg.

- MAMAExpDLL mama_bool_t mamaMsgIterator_hasNext (mamaMsgIterator iterator)

  Return whether there is a next next field in the mamaMsg.

- MAMAExpDLL mamaMsgField mamaMsgIterator_begin (mamaMsgIterator iterator)

  Set iterator to first field in mamaMsg.

- MAMAExpDLL mamaMsgField mamaMsgIterator_end (mamaMsgIterator iterator)

  Set iterator to last field in mamaMsg.

- MAMAExpDLL mama_status mamaMsgIterator_destroy (mamaMsgIterator iterator)

  Destroy iterator.

5.16.1 Define Documentation

5.16.1.1 #define ENTITLE_FIELD_NAME "wEntitleCode"

5.16.1.2 #define ENTITLE_FIELD_ID 496

5.16.2 Typedef Documentation

5.16.2.1 typedef enum mamaPayloadType_ mamaPayloadType

Enum to reference the available MAMA message payloads.

5.16.2.2 typedef void(MAMACALLTYPE ∗) mamaMsgIteratorCb(const mamaMsg msg, const mamaMsgField field, void ∗closure)

Iterator callback method is invoked for each field in the message.

Parameters:

msg The message.

field The field.

closure The closure specified in mamaMsg_iterateFields().

5.16.3 Enumeration Type Documentation

5.16.3.1 enum mamaPayloadType_

Enum to reference the available MAMA message payloads.

Enumerator:

MAMA_PAYLOAD_WOMBAT_MSG

MAMA_PAYLOAD_TIBRV

MAMA_PAYLOAD_FAST

MAMA_PAYLOAD_V5

MAMA_PAYLOAD_AVIS

MAMA_PAYLOAD_TICK42BLP

MAMA_PAYLOAD_RAI

MAMA_PAYLOAD_EXEGY

MAMA_PAYLOAD_UNKNOWN

```
{
    MAMA_PAYLOAD_WOMBAT_MSG = 'W',
    MAMA_PAYLOAD_TIBRV      = 'R',
    MAMA_PAYLOAD_FAST       = 'F',
    MAMA_PAYLOAD_V5         = '5',
    MAMA_PAYLOAD_AVIS       = 'A',
    MAMA_PAYLOAD_TICK42BLP  = 'B',
    MAMA_PAYLOAD_RAI        = 'I',
    MAMA_PAYLOAD_EXEGY      = 'X',
    MAMA_PAYLOAD_UNKNOWN    = 'U'
} mamaPayloadType;
```

5.16.4 Function Documentation

5.16.4.1 MAMAExpDLL const char* mamaPayload_convertToString (mamaPayloadType *payloadType*)

Convert a mamaPayloadType value to a string.

Do no attempt to free the string result.

Parameters:

payloadType The payloadType to convert.

5.16.4.2 MAMAExpDLL mama_status mamaMsg_create (mamaMsg * *msg*)

Create a mamaMsg.

Parameters:

msg The location of a mamaMsg where to store the result.

5.16.4.3 MAMAExpDLL mama_status mamaMsg_createForPayload (mamaMsg * *msg*, const char *id*)

Create a mamaMsg.

Parameters:

msg The location of a mamaMsg where to store the result.

id The identifier of the payload to be used.

5.16.4.4 MAMAExpDLL mama_status mamaMsg_createForPayloadBridge (mamaMsg * *msg*, mamaPayloadBridge *payloadBridge*)

Create a mamaMsg.

Parameters:

msg The location of a mamaMsg where to store the result.

id The payload bridge to be used.

5.16.4.5 MAMAExpDLL mama_status mamaMsg_createForTemplate (mamaMsg * *msg*, mama_u32_t *templateId*)

Create a mamaMsg for a particular template ID.

Parameters:

msg The location of a mamaMsg where to store the result.

templateID The wombatMsg to encapsulate.

5.16.4.6 MAMAExpDLL mama_status mamaMsg_copy (mamaMsg *src*, mamaMsg * *copy*)

Copy a mamaMsg.

If mamaMsg_create() has not been called for for the destination message object then this will be done by the function.

Parameters:

src The message to copy.

copy A pointer to the destination message.

5.16.4.7 MAMAExpDLL mama_status mamaMsg_getTempCopy (mamaMsg *src*, mamaMsg * *copy*)

Get a temporary copy of the mamaMsg so to be able to modify the content.

If the message can be modified directly, the message itself is returned. If the message cannot be modified then only one copy is performed the first time this function is called and then the same copy is returned when this function is called again. The copy is destroyed when the original message is destroyed.

Parameters:

src The message to copy.

copy A pointer to the destination message.

5.16.4.8 MAMAExpDLL mama_status mamaMsg_clear (mamaMsg *msg*)

Clear a msg.

All fields are removed.

Parameters:

msg The mamaMsg to clear.

5.16.4.9 MAMAExpDLL mama_status mamaMsg_getSendSubject (const mamaMsg *msg*, const char ** *subject*)

TIB/RV only.

Get the TIB/RV send subject from the message. For other message types MAMA_-STATUS_NOT_IMPLEMENTED will be returned.

Parameters:

src The message to copy.

copy A pointer to the destination message.

5.16.4.10 MAMAExpDLL mama_status mamaMsg_destroy (mamaMsg *msg*)

Destroy a message and free any resources associated with it.

Parameters:

msg The message.

5.16.4.11 MAMAExpDLL mama_status mamaMsg_getPayloadType (mamaMsg *msg*, mamaPayloadType * *payloadType*)

Get the type of the message payload.

Parameters:

msg The message.

payloadType The payload type.

5.16.4.12 MAMAExpDLL mama_status mamaMsg_getByteSize (const mamaMsg *msg*, mama_size_t * *size*)

Get the size of the underlying message in bytes.

Only works for tibrvMsg and wombatmsg types.

Parameters:

msg The MAMA message

size (out) The size of the message in bytes

Returns:

Resulting status of the call

5.16.4.13 MAMAExpDLL mama_status mamaMsg_addBool (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_bool_t *value*)

Add a new bool field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.14 MAMAExpDLL mama_status mamaMsg_addChar (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, char *value*)

Add a new char field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.15 MAMAExpDLL mama_status mamaMsg_addI8 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i8_t *value*)

Add a new 8 bit signed int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.16 MAMAExpDLL mama_status mamaMsg_addU8 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u8_t *value*)

Add a new 8 bit unsigned int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.17 MAMAExpDLL mama_status mamaMsg_addI16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i16_t *value*)

Add a new 16 bit signed int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.18 MAMAExpDLL mama_status mamaMsg_addU16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u16_t *value*)

Add a new 16 bit unsigned int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.19 MAMAExpDLL mama_status mamaMsg_addI32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i32_t *value*)

Add a new 32 bit unsigned int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.20 MAMAExpDLL mama_status mamaMsg_addU32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u32_t *value*)

Add a new 32 bit signed int field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.21 MAMAExpDLL mama_status mamaMsg_addI64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i64_t *value*)

Add a new I64 field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.22 MAMAExpDLL mama_status mamaMsg_addU64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u64_t *value*)

Add a new U64 field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.23 MAMAExpDLL mama_status mamaMsg_addF32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f32_t *value*)

Add a new float (f32) field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.24 MAMAExpDLL mama_status mamaMsg_addF64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f64_t *value*)

Add a new f64 field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.25 MAMAExpDLL mama_status mamaMsg_addString (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char * *value*)

Add a const char * field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.26 MAMAExpDLL mama_status mamaMsg_addOpaque (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const void * *value*, mama_size_t *size*)

Add an opaque field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

size The size of the opaque in bytes

5.16.4.27 MAMAExpDLL mama_status mamaMsg_addDateTime (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaDateTime *value*)

Add a MAMA date/time field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.28 MAMAExpDLL mama_status mamaMsg_addPrice (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaPrice *value*)

Add a MAMA price field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.29 MAMAExpDLL mama_status mamaMsg_addMsg (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaMsg *value*)

Add a mamaMsg object to the message.

Parameters:

msg The message to which the field is being added.

name The name identifier for the field (optional).

fid The field identifier

value The value of the field

Returns:

mama_status MAMA_STATUS_OK if the function executes successfully

5.16.4.30 MAMAExpDLL mama_status mamaMsg_addVectorBool (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_bool_t *value*[ ], mama_size_t *numElements*)

Add an array of booleans to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of booleans.

numElements The number of elements in the array.

5.16.4.31 MAMAExpDLL mama_status mamaMsg_addVectorChar (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char *value*[ ], mama_size_t *numElements*)

Add an array of characters to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of characters.

numElements The number of elements in the array.

5.16.4.32 MAMAExpDLL mama_status mamaMsg_addVectorI8 (mamaMsg *msg*, const char ∗ *name*, mama_fid_t *fid*, const mama_i8_t *value*[ ], mama_size_t *numElements*)

Add an array of signed 8 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of signed 8 bit integers.

numElements The number of elements in the array.

5.16.4.33 MAMAExpDLL mama_status mamaMsg_addVectorU8 (mamaMsg *msg*, const char ∗ *name*, mama_fid_t *fid*, const mama_u8_t *value*[ ], mama_size_t *numElements*)

Add an array of unsigned 8 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of unsigned 8 bit integers.

numElements The number of elements in the array.

5.16.4.34 MAMAExpDLL mama_status mamaMsg_addVectorI16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i16_t *value*[ ], mama_size_t *numElements*)

Add an array of signed 16 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of signed 16 bit integers.

numElements The number of elements in the array.

5.16.4.35 MAMAExpDLL mama_status mamaMsg_addVectorU16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u16_t *value*[ ], mama_size_t *numElements*)

Add an array of unsigned 16 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of unsigned 16 bit integers.

numElements The number of elements in the array.

5.16.4.36 MAMAExpDLL mama_status mamaMsg_addVectorI32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i32_t *value*[ ], mama_size_t *numElements*)

Add an array of signed 32 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of signed 32 bit integers.

numElements The number of elements in the array.

5.16.4.37 MAMAExpDLL mama_status mamaMsg_addVectorU32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u32_t *value*[ ], mama_size_t *numElements*)

Add an array of unsigned 32 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of unsigned 32 bit integers.

numElements The number of elements in the array.

5.16.4.38 MAMAExpDLL mama_status mamaMsg_addVectorI64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i64_t *value*[ ], mama_size_t *numElements*)

Add an array of signed 64 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of signed 64 bit integers.

numElements The number of elements in the array.

5.16.4.39 MAMAExpDLL mama_status mamaMsg_addVectorU64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u64_t *value*[ ], mama_size_t *numElements*)

Add an array of unsigned 64 bit integers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of unsigned 64 bit integers.

numElements The number of elements in the array.

5.16.4.40 MAMAExpDLL mama_status mamaMsg_addVectorF32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_f32_t *value*[ ], mama_size_t *numElements*)

Add an array of 32 bit floating point numbers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of 32 bit floating point numbers.

numElements The number of elements in the array.

5.16.4.41 MAMAExpDLL mama_status mamaMsg_addVectorF64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_f64_t *value*[ ], mama_size_t *numElements*)

Add an array of 64 bit floating point numbers to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of 64 bit floating point numbers.

numElements The number of elements in the array.

5.16.4.42 MAMAExpDLL mama_status mamaMsg_addVectorString (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char * *value*[ ], mama_size_t *numElements*)

Add an array of strings (char*) to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of strings (char*).

numElements The number of elements in the array.

5.16.4.43 MAMAExpDLL mama_status mamaMsg_addVectorMsg (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaMsg *value*[ ], mama_size_t *numElements*)

Add an array of mamaMsg objects to the message.

Parameters:

msg The message to which the field is being added.

name The name identifier for the field (optional).

fid The field identifier

value[ ] The value of the field.

numElements The number of elements in the mamaMsg array.

Returns:

mama_status MAMA_STATUS_OK if the function returns successfully.

5.16.4.44 MAMAExpDLL mama_status mamaMsg_addVectorDateTime (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaDateTime *value*[ ], mama_size_t *numElements*)

Add an array of timestamps to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of timestamps.

numElements The number of elements in the array.

5.16.4.45 MAMAExpDLL mama_status mamaMsg_addVectorPrice (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaPrice *value*[ ], mama_size_t *numElements*)

Add an array of prices to the message.

Parameters:

msg The mamaMsg

name The name identifier for the field (optional).

fid The field identifier.

value[ ] Array of values.

numElements The number of elements in the array.

5.16.4.46 MAMAExpDLL mama_status mamaMsg_updateBool (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_bool_t *value*)

Update the value of an existing bool field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.47 MAMAExpDLL mama_status mamaMsg_updateChar (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, char *value*)

Update the value of an existing char field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.48 MAMAExpDLL mama_status mamaMsg_updateI8 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i8_t *value*)

Update the value of an existing I8 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.49 MAMAExpDLL mama_status mamaMsg_updateU8 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u8_t *value*)

Update the value of an existing U8 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.50 MAMAExpDLL mama_status mamaMsg_updateI16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i16_t *value*)

Update the value of an existing I16 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.51 MAMAExpDLL mama_status mamaMsg_updateU16 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u16_t *value*)

Update the value of an existing U16 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.52 MAMAExpDLL mama_status mamaMsg_updateI32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i32_t *value*)

Update the value of an existing I32 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.53 MAMAExpDLL mama_status mamaMsg_updateU32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u32_t *value*)

Update the value of an existing U32 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.54 MAMAExpDLL mama_status mamaMsg_updateI64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i64_t *value*)

Update the value of an existing I64 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.55 MAMAExpDLL mama_status mamaMsg_updateU64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u64_t *value*)

Update the value of an existing U64 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.56 MAMAExpDLL mama_status mamaMsg_updateF32 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f32_t *value*)

Update the value of an existing float field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.57 MAMAExpDLL mama_status mamaMsg_updateF64 (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f64_t *value*)

Update the value of an existing f64 field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.58 MAMAExpDLL mama_status mamaMsg_updateString (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char * *value*)

Update the value of an existing const char* field.

If the field does not exist it is added.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The new value.

5.16.4.59 MAMAExpDLL mama_status mamaMsg_updateOpaque (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const void * *value*, mama_size_t *size*)

Update an opaque field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

size The size of the opaque in bytes

5.16.4.60 MAMAExpDLL mama_status mamaMsg_updateDateTime (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaDateTime *value*)

Update a MAMA date/time field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.61 MAMAExpDLL mama_status mamaMsg_updatePrice (mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaPrice *value*)

Update a MAMA price field.

Parameters:

msg The message.

name The name.

fid The field identifier.

value The value.

5.16.4.62 MAMAExpDLL mama_status mamaMsg_getBool (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_bool_t * *result*)

Get a bool field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.63 MAMAExpDLL mama_status mamaMsg_applyMsg (mamaMsg *dest*, mamaMsg *src*)

Apply the contents of one message to another as updates.

Parameters:

dest The message to update.

src The source messages.

5.16.4.64 MAMAExpDLL mama_status mamaMsg_getNativeMsg (mamaMsg *msg*, void ** *nativeMsg*)

Get the native message structure for the underlying message This function is for internal NYSE Technologies use only.

Parameters:

msg The message

nativeMsg The resulting native handle.

Returns:

mama_status Returns MAMA_STATUS_OK if the call was successful.

5.16.4.65 MAMAExpDLL mama_status mamaMsg_getReplyHandle (mamaMsg *msg*, mamaMsgReply * *replyHandle*)

Get the reply handle for the message and increments the reference count for the handle.

Parameters:

msg The message

replyHandle The resulting reply handle.

Returns:

mama_status Returns MAMA_STATUS_OK if the call was successful.

5.16.4.66 MAMAExpDLL mama_status mamaMsg_setReplyHandle (mamaMsg *msg*, mamaMsgReply *replyHandle*)

Set the reply handle for the message and increments the reference count for the handle.

Parameters:

msg The message

replyHandle The reply handle for this message.

Returns:

mama_status Returns MAMA_STATUS_OK if the call was successful.

5.16.4.67 MAMAExpDLL mama_status mamaMsg_copyReplyHandle (mamaMsgReply *src*, mamaMsgReply ∗ *copy*)

Create a new copy of the src reply handle.

Parameters:

src The reply handler to copy.

copy A pointer to the destination reply handle.

Returns:

mama_status Returns MAMA_STATUS_OK if the call was successful.

5.16.4.68 MAMAExpDLL mama_status mamaMsg_destroyReplyHandle (mamaMsgReply *replyHandle*)

Destroys a reply handle.

Parameters:

replyHandle The reply handle for this message.

Returns:

mama_status Returns MAMA_STATUS_OK if the call was successful.

5.16.4.69 MAMAExpDLL mama_status mamaMsg_updateSubMsg (mamaMsg *msg*, const char ∗ *fname*, mama_fid_t *fid*, const mamaMsg *subMsg*)

Update a sub-message field.

Parameters:

msg The message.

fname The name

fid The field identifier

subMsg The new value.

5.16.4.70 MAMAExpDLL mama_status mamaMsg_updateVectorMsg (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mamaMsg *msgList*[ ], mama_size_t *numElements*)

Update a vector message field.

Parameters:

msg The message.

fname The name

fid The field identifier

msgList The new value.

numElements the number of elements in the vector

5.16.4.71 MAMAExpDLL mama_status mamaMsg_updateVectorString (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const char * *strList*[ ], mama_size_t *numElements*)

Update a vector string field.

Parameters:

msg The message.

fname The name

fid The field identifier

strList The new value.

numElements the number of elements in the vector

5.16.4.72 MAMAExpDLL mama_status mamaMsg_updateVectorBool (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_bool_t *boolList*[ ], mama_size_t *numElements*)

Update a vector bool field.

Parameters:

msg The message.

fname The name

fid The field identifier

boolList The new value.

numElements the number of elements in the vector

5.16.4.73 MAMAExpDLL mama_status mamaMsg_updateVectorChar (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const char *charList*[ ], mama_size_t *numElements*)

Update a vector char field.

Parameters:

msg The message.

fname The name

fid The field identifier

charList The new value.

numElements the number of elements in the vector

5.16.4.74 MAMAExpDLL mama_status mamaMsg_updateVectorI8 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_i8_t *i8List*[ ], mama_size_t *numElements*)

Update a vector I8 field.

Parameters:

msg The message.

fname The name

fid The field identifier

i8List The new value.

numElements the number of elements in the vector

5.16.4.75 MAMAExpDLL mama_status mamaMsg_updateVectorU8 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_u8_t *u8List*[ ], mama_size_t *numElements*)

Update a vector U8 field.

Parameters:

msg The message.

fname The name

fid The field identifier

u8List The new value.

numElements the number of elements in the vector

5.16.4.76 MAMAExpDLL mama_status mamaMsg_updateVectorI16 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_i16_t *i16List*[ ], mama_size_t *numElements*)

Update a vector I16 field.

Parameters:

msg The message.

fname The name

fid The field identifier

i16List The new value.

numElements the number of elements in the vector

5.16.4.77 MAMAExpDLL mama_status mamaMsg_updateVectorU16 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_u16_t *u16List*[ ], mama_size_t *numElements*)

Update a vector U16 field.

Parameters:

msg The message.

fname The name

fid The field identifier

u16List The new value.

numElements the number of elements in the vector

5.16.4.78 MAMAExpDLL mama_status mamaMsg_updateVectorI32 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_i32_t *i32List*[ ], mama_size_t *numElements*)

Update a vector I32 field.

Parameters:

msg The message.

fname The name

fid The field identifier

i32List The new value.

numElements the number of elements in the vector

5.16.4.79 MAMAExpDLL mama_status mamaMsg_updateVectorU32 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_u32_t *u32List*[ ], mama_size_t *numElements*)

Update a vector U32 field.

Parameters:

msg The message.

fname The name

fid The field identifier

u32List The new value.

numElements the number of elements in the vector

5.16.4.80 MAMAExpDLL mama_status mamaMsg_updateVectorI64 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_i64_t *i64List*[ ], mama_size_t *numElements*)

Update a vector I64 field.

Parameters:

msg The message.

fname The name

fid The field identifier

i64List The new value.

numElements the number of elements in the vector

5.16.4.81 MAMAExpDLL mama_status mamaMsg_updateVectorU64 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_u64_t *u64List*[ ], mama_size_t *numElements*)

Update a vector U64 field.

Parameters:

msg The message.

fname The name

fid The field identifier

u64List The new value.

numElements the number of elements in the vector

5.16.4.82 MAMAExpDLL mama_status mamaMsg_updateVectorF32 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_f32_t *f32List*[ ], mama_size_t *numElements*)

Update a vector F32 field.

Parameters:

msg The message.

fname The name

fid The field identifier

f32List The new value.

numElements the number of elements in the vector

5.16.4.83 MAMAExpDLL mama_status mamaMsg_updateVectorF64 (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mama_f64_t *f64List*[ ], mama_size_t *numElements*)

Update a vector F64 field.

Parameters:

msg The message.

fname The name

fid The field identifier

f64List The new value.

numElements the number of elements in the vector

5.16.4.84 MAMAExpDLL mama_status mamaMsg_updateVectorPrice (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mamaPrice * *priceList*[ ], mama_size_t *numElements*)

Update a vector price field.

Parameters:

msg The message.

fname The name

fid The field identifier

priceList The new value.

numElements the number of elements in the vector

5.16.4.85 MAMAExpDLL mama_status mamaMsg_updateVectorTime (mamaMsg *msg*, const char * *fname*, mama_fid_t *fid*, const mamaDateTime *timeList*[ ], mama_size_t *numElements*)

Update a vector mamaDateTime field.

Parameters:

msg The message.

fname The name

fid The field identifier

timeList The new value.

numElements the number of elements in the vector

5.16.4.86 MAMAExpDLL mama_status mamaMsg_getChar (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, char * *result*)

Get a char field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.87 MAMAExpDLL mama_status mamaMsg_getI8 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i8_t * *result*)

Get a I8, signed 8 bit integer, field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.88 MAMAExpDLL mama_status mamaMsg_getU8 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u8_t * *result*)

Get an unsigned 8-bit integer field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.89 MAMAExpDLL mama_status mamaMsg_getI16 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i16_t * *result*)

Get a I16, signed 16 bit integer, field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.90 MAMAExpDLL mama_status mamaMsg_getU16 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u16_t * *result*)

Get an unsigned 16-bit integer field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.91 MAMAExpDLL mama_status mamaMsg_getI32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i32_t * *result*)

Get a I32, signed 32 bit integer, field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.92 MAMAExpDLL mama_status mamaMsg_getU32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u32_t * *result*)

Get an unsigned 32-bit integer field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.93 MAMAExpDLL mama_status mamaMsg_getI64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_i64_t * *result*)

Get a I64, signed 64 bit integer, field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.94 MAMAExpDLL mama_status mamaMsg_getU64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u64_t * *result*)

Get an unsigned 64-bit integer field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.95 MAMAExpDLL mama_status mamaMsg_getF32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f32_t * *result*)

Get a float (f32) field.

Parameters:

msg The message.

name The name.

fid The field identifier.

result (out) The double value.

5.16.4.96 MAMAExpDLL mama_status mamaMsg_getF64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_f64_t * *result*)

Get a f64 field.

Parameters:

msg The message.

name The name.

fid The field identifier.

result (out) The double value.

5.16.4.97 MAMAExpDLL mama_status mamaMsg_getString (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char ** *result*)

Get a const char * field.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result (out) the string value.

5.16.4.98 MAMAExpDLL mama_status mamaMsg_getOpaque (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const void ** *result*, mama_size_t * *size*)

Get an opaque field.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result (out) The opaque value.

size (out) Length in bytes of the opaque field.

5.16.4.99 MAMAExpDLL mama_status mamaMsg_getField (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mamaMsgField * *result*)

Get a MAMA msg field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result. The result contains the reusable field object of the mamaMsg object. Applications calling this method will receive the same reusable object for repeated calls on same mamaMsg object.

5.16.4.100 MAMAExpDLL mama_status mamaMsg_getDateTime (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mamaDateTime *result*)

Get a MAMA date/time field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.101 MAMAExpDLL mama_status mamaMsg_getDateTimeMSec (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mama_u64_t * *milliseconds*)

Get the value of a MAMA date/time field in milliseconds.

Parameters:

msg The message.

name The name

fid The field identifier

milliseconds (out) Pointer to the value in milliseconds

5.16.4.102 MAMAExpDLL mama_status mamaMsg_getPrice (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mamaPrice *result*)

Get a MAMA price field.

Parameters:

msg The message.

name The name

fid The field identifier

result (out) Pointer to the result

5.16.4.103 MAMAExpDLL mama_status mamaMsg_getMsg (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, mamaMsg * *result*)

Get a submessage.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result. result when the method returns..

5.16.4.104 MAMAExpDLL mama_status mamaMsg_getVectorBool (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_bool_t ** *result*, mama_size_t * *resultLen*)

Get a vector of booleans.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.105 MAMAExpDLL mama_status mamaMsg_getVectorChar (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char ** *result*, mama_size_t * *resultLen*)

Get a vector of chars.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.106 MAMAExpDLL mama_status mamaMsg_getVectorI8 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i8_t ** *result*, mama_size_t * *resultLen*)

Get a vector of signed 8 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.107 MAMAExpDLL mama_status mamaMsg_getVectorU8 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u8_t ** *result*, mama_size_t * *resultLen*)

Get a vector of unsigned 8 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.108 MAMAExpDLL mama_status mamaMsg_getVectorI16 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i16_t ** *result*, mama_size_t * *resultLen*)

Get a vector of signed 16 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.109 MAMAExpDLL mama_status mamaMsg_getVectorU16 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u16_t ** *result*, mama_size_t * *resultLen*)

Get a vector of unsigned 16 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.110 MAMAExpDLL mama_status mamaMsg_getVectorI32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i32_t ** *result*, mama_size_t * *resultLen*)

Get a vector of signed 32 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.111 MAMAExpDLL mama_status mamaMsg_getVectorU32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u32_t ** *result*, mama_size_t * *resultLen*)

Get a vector of unsigned 32 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.112 MAMAExpDLL mama_status mamaMsg_getVectorI64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_i64_t ** *result*, mama_size_t * *resultLen*)

Get a vector of signed 64 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.113 MAMAExpDLL mama_status mamaMsg_getVectorU64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_u64_t ** *result*, mama_size_t * *resultLen*)

Get a vector of unsigned 64 bit integers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.114 MAMAExpDLL mama_status mamaMsg_getVectorF32 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_f32_t ** *result*, mama_size_t * *resultLen*)

Get a vector of 32 bit floating point numbers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.115 MAMAExpDLL mama_status mamaMsg_getVectorF64 (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mama_f64_t ** *result*, mama_size_t * *resultLen*)

Get a vector of 64 bit floating point numbers.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.116 MAMAExpDLL mama_status mamaMsg_getVectorString (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const char *** *result*, mama_size_t * *resultLen*)

Get a vector of strings (char*).

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.117 MAMAExpDLL mama_status mamaMsg_getVectorDateTime (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaDateTime * *result*, mama_size_t * *resultLen*)

Get a vector of timestamps.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.118 MAMAExpDLL mama_status mamaMsg_getVectorPrice (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaPrice * *result*, mama_size_t * *resultLen*)

Get a vector of prices.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.119 MAMAExpDLL mama_status mamaMsg_getVectorMsg (const mamaMsg *msg*, const char * *name*, mama_fid_t *fid*, const mamaMsg ** *result*, mama_size_t * *resultLen*)

Get a vector of submessages.

Parameters:

msg The message.

name The field name.

fid The field identifier.

result A pointer to the result.

resultLen An integer pointer that will contain the length of the result when the method returns..

5.16.4.120 MAMAExpDLL mama_status mamaMsg_getNumFields (const mamaMsg *msg*, mama_size_t ∗ *numFields*)

Returns the total number of fields in the message.

Sub-messages count as a single field.

Parameters:

msg The message.

numFields (out) The number of fields in the message.

5.16.4.121 MAMAExpDLL const char∗ mamaMsg_toString (const mamaMsg *msg*)

Return a const char ∗ representation the message.

Must call mamaMsg_freeString() to free memory allocated for string.

Parameters:

msg The message.

Returns:

A string representation of the message.

5.16.4.122 MAMAExpDLL void mamaMsg_freeString (const mamaMsg *msg*, const char ∗ *msgString*)

Free the memory allocated by mamaMsg_toString.

Parameters:

msg The message.

msgString The string allocated by mamaMsg_toString

5.16.4.123 MAMAExpDLL mama_status mamaMsg_getEntitleCode (const mamaMsg *msg*, mama_i32_t ∗ *code*)

Get the entitle code for this message.

The result defaults to 0 (no entitlement) if the field is not present.

Parameters:

msg The message.

code (out) The entitle code

5.16.4.124 MAMAExpDLL mama_status mamaMsg_getSeqNum (const mamaMsg *msg*, mama_seqnum_t ∗ *seqNum*)

Get the sequence number for this message if available.

The result defaults to 0 and MAMA_STATUS_NOT_FOUND retuned if the field is not present.

Parameters:

msg The message.

code (out) The sequence number.

5.16.4.125 MAMAExpDLL mamaMsgType mamaMsgType_typeForMsg (const mamaMsg *msg*)

Extract the type from the supplied message.

Parameters:

msg The message.

Returns:

The type.

5.16.4.126 MAMAExpDLL const char∗ mamaMsgType_stringForMsg (const mamaMsg *msg*)

Return the type name.

Parameters:

msg The message.

Returns:

The type name.

5.16.4.127 MAMAExpDLL const char∗ mamaMsgType_stringForType (const mamaMsgType *type*)

Return the type name.

Parameters:

type The message.

Returns:

The type name.

5.16.4.128 MAMAExpDLL mama_status mamaMsg_getFieldAsString (const mamaMsg *msg*, const char ∗ *fieldName*, mama_fid_t *fid*, char ∗ *buf*, mama_size_t *length*)

Convert the value of the specified field to a string.

Caller must provide a buffer for the value.

Parameters:

msg The message.

fieldName The field name.

fid The field identifier.

buf The buffer where the resulting string will be copied.

length The length of the caller supplied buffer.

5.16.4.129 MAMAExpDLL mama_status mamaMsg_iterateFields (const mamaMsg *msg*, mamaMsgIteratorCb *callback*, const mamaDictionary *dict*, void ∗ *closure*)

Invoke the specified callback for each field in the message.

Parameters:

msg The message.

callback The msgIteratorCb to invoke.

dict The dictionary.

closure The user specified closure passed to the callback.

5.16.4.130 MAMAExpDLL mama_status mamaMsg_getPlatformError (const mamaMsg *msg*, void ∗∗ *error*)

Return the middleware specific platform error.

When a mamaSubscription_ method returns MAMA_STATUS_PLATFORM the error will be the result from the underlying platform.

Parameters:

msg The message.

error (out) The platform dependent result.

5.16.4.131 MAMAExpDLL int mamaMsg_isFromInbox (const mamaMsg *msg*)

Return true if this message was sent from a client's inbox.

5.16.4.132 MAMAExpDLL mama_status mamaMsg_createFromByteBuffer (mamaMsg ∗ *msg*, const void ∗ *buffer*, mama_size_t *bufferLength*)

Create a mamaMsg from the provided byte buffer.

The application is responsible for destroying the message. In this function a copy of the buffer isn't made and the client must maintain it until after the message has been destroyed using the mamaMsg_destroy function. The function can determine from the buffer whether it is a wombatmsg or the native format for the transport being used.

Any transport differences are detailed below.

rv: The bufferLength parameter is not required.

Parameters:

msg The mamaMsg object

buffer The byte array containing the wire format of the message

bufferLength The length, in bytes, of the supplied buffer

Returns:

mama_status The outcome of the operation

5.16.4.133 MAMAExpDLL mama_status mamaMsg_getByteBuffer (const mamaMsg *msg*, const void ∗∗ *buffer*, mama_size_t ∗ *bufferLength*)

Return the internal message buffer as an array of bytes which is suitable for writing to a file.

Do not modify the returned byte buffer as this is the internal message buffer and remains part of the message until it is destroyed. The buffer will be written to the address supplied by the client. The buffer will contain the wire format of the underlying message implementation but does not contain delivery information such as source, topic etc.

Parameters:

msg The mamaMsg object

buffer The byte array containing the message wire format

bufferLength The length, in bytes, of the returned buffer

Returns:

mama_status The outcome of the operation

5.16.4.134 MAMAExpDLL mama_status mamaMsg_detach (mamaMsg *msg*)

Normally the Mama API owns incoming mamaMsg objects and they go out of scope when the message callback returns.

Calling this method from the message callback transfers ownership, and responsibility for calling mamaMsg_destroy() to the caller.

5.16.4.135 MAMAExpDLL mama_status mamaMsg_getIsDefinitelyDuplicate (mamaMsg *msg*, int ∗ *result*)

Return true if this message is definitely a duplicate message.

This condition will not occur with the current feed handlers.

5.16.4.136 MAMAExpDLL mama_status mamaMsg_getIsPossiblyDuplicate (mamaMsg *msg*, int ∗ *result*)

Return true if this message is possibly a duplicate message.

This may occur in the event of a fault tolerant feed handler take over where the feed handler replays messages to prevent gaps.

5.16.4.137 MAMAExpDLL mama_status mamaMsg_getIsPossiblyDelayed (mamaMsg *msg*, int ∗ *result*)

Return true if the message is possibly delayed.

This condition may be true during a fault-tolerant take over.

5.16.4.138 MAMAExpDLL mama_status mamaMsg_getIsDefinitelyDelayed (mamaMsg *msg*, int ∗ *result*)

Return true if the message is delayed.

This condition may be true during a faul tolerant take over.

5.16.4.139 MAMAExpDLL mama_status mamaMsg_getIsOutOfSequence (mamaMsg *msg*, int ∗ *result*)

Return true when the FH sends the message out of sequence.

5.16.4.140 MAMAExpDLL mama_status mamaMsg_setNewBuffer (mamaMsg *msg*, void ∗ *buffer*, mama_size_t *size*)

Set a new buffer for an existing mamaMsg.

This approach is faster than creating a new message for buffers as the message can reuse memory allocated during previous use.

Parameters:

msg The mamaMsg that will receive the new buffer.

buffer The new buffer to use for this message.

size The size of the buffer.

5.16.4.141 MAMAExpDLL mama_status mamaMsg_getNativeHandle (const mamaMsg *msg*, void ∗∗ *result*)

Get the native middleware message handle.

This is only intended for internal use.

5.16.4.142 MAMAExpDLL mama_status mamaMsgIterator_create (mamaMsgIterator * *iterator*, mamaDictionary *dict*)

Creates a new iterator for use with a mamaMsg.

Parameters:

iterator Pointer to memory to be used for iterator

dict The mamaDictionary to be associated with the iterator

Returns:

mama_status The outcome of the operation

5.16.4.143 MAMAExpDLL mama_status mamaMsgIterator_associate (mamaMsgIterator *iterator*, mamaMsg *msg*)

Associate an iterator for use with a mamaMsg.

Parameters:

msg The mamaMsg that will use the iterator

iterator iterator for use with msg

Returns:

mama_status The outcome of the operation

5.16.4.144 MAMAExpDLL mama_status mamaMsgIterator_setDict (mamaMsgIterator *iterator*, mamaDictionary *dict*)

Associate a mamaDictionary for use with an iterator.

Parameters:

iterator The iterator that will use the dictionary

dict The dictionary to be used

Returns:

mama_status The outcome of the operation.

5.16.4.145 MAMAExpDLL mamaMsgField mamaMsgIterator_next (mamaMsgIterator *iterator*)

Return the next field in the mamaMsg.

Parameters:

iterator The iterator being used

Returns:

mamaMsgField The next field in the message

5.16.4.146 MAMAExpDLL mama_bool_t mamaMsgIterator_hasNext (mamaMsgIterator *iterator*)

Return whether there is a next next field in the mamaMsg.

Parameters:

iterator The iterator being used

Returns:

bool Does the message have another field

5.16.4.147 MAMAExpDLL mamaMsgField mamaMsgIterator_begin (mamaMsgIterator *iterator*)

Set iterator to first field in mamaMsg.

Parameters:

iterator The iterator being used

Returns:

mamaMsgField The first field in the message

5.16.4.148 MAMAExpDLL mamaMsgField mamaMsgIterator_end (mamaMsgIterator *iterator*)

Set iterator to last field in mamaMsg.

Parameters:

iterator The iterator being used

Returns:

mamaMsgField The last field in the message

5.16.4.149 MAMAExpDLL mama_status mamaMsgIterator_destroy (mamaMsgIterator *iterator*)

Destroy iterator.

Parameters:

iterator The iterator being destroyed

5.17 msgfield.h File Reference

```
#include <mama/types.h>
#include <mama/status.h>
#include <mama/fielddesc.h>
#include <stdlib.h>
#include "wombat/port.h"
```

Functions

- MAMAExpDLL mama_status mamaMsgField_create (mamaMsgField *msgField)

  Create a field.

- MAMAExpDLL mama_status mamaMsgField_destroy (mamaMsgField msgField)

  Destroy a message and free any resources associated with it.

- MAMAExpDLL mama_status mamaMsgField_getDescriptor (const mamaMsgField msgField, mamaFieldDescriptor *result)

  Get the field descriptor for this field.

- MAMAExpDLL mama_status mamaMsgField_getFid (const mamaMsgField msgField, mama_fid_t *result)

  Get the field ID for this field.

- MAMAExpDLL mama_status mamaMsgField_getName (const mamaMsgField msgField, const char **result)

  Get the field name for this field.

- MAMAExpDLL mama_status mamaMsgField_getType (const mamaMsgField msgField, mamaFieldType *result)

  Get the field type for this field.

- MAMAExpDLL mama_status mamaMsgField_getTypeName (const mamaMsgField msgField, const char **result)

  Get the type name for this field.

- MAMAExpDLL mama_status mamaMsgField_getBool (const mamaMsgField msgField, mama_bool_t *result)

  Get a boolean field.

- MAMAExpDLL mama_status mamaMsgField_getChar (const mamaMsgField msgField, char *result)

  Get a character field.

- MAMAExpDLL mama_status mamaMsgField_getI8 (const mamaMsgField msgField, mama_i8_t *result)

  Get a I8, signed 8 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getU8 (const mamaMsgField msgField, mama_u8_t *result)

  Get a U8, unsigned 8 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getI16 (const mamaMsgField msgField, mama_i16_t *result)

  Get a I16, signed 16 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getU16 (const mamaMsgField msgField, mama_u16_t *result)

  Get a U16, unsigned 16 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getI32 (const mamaMsgField msgField, mama_i32_t *result)

  Get a I32, signed 32 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getU32 (const mamaMsgField msgField, mama_u32_t *result)

  Get a U32, unsigned 32 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getI64 (const mamaMsgField msgField, mama_i64_t *result)

  Get a I64, signed 64 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getU64 (const mamaMsgField msgField, mama_u64_t *result)

  Get a U64, unsigned 64 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getF32 (const mamaMsgField msgField, mama_f32_t *result)

  Get a F32, floating point 32 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getF64 (const mamaMsgField msgField, mama_f64_t *result)

Get a F64, floating point 64 bit integer, field.

- MAMAExpDLL mama_status mamaMsgField_getString (const mamaMsgField msgField, const char **result)

  *Get a const char * field.*

- MAMAExpDLL mama_status mamaMsgField_getOpaque (const mamaMsgField msgField, const void **result, mama_size_t *size)

  Get an opaque field.

- MAMAExpDLL mama_status mamaMsgField_getDateTime (const mamaMsgField msgField, mamaDateTime result)

  Get a MAMA date/time field.

- MAMAExpDLL mama_status mamaMsgField_getPrice (const mamaMsgField msgField, mamaPrice result)

  Get a MAMA price field.

- MAMAExpDLL mama_status mamaMsgField_getMsg (const mamaMsgField msgField, mamaMsg *result)

  Get a MAMA submessage field.

- MAMAExpDLL mama_status mamaMsgField_getVectorBool (const mamaMsgField msgField, const mama_bool_t **result, mama_size_t *size)

  Get a vector of booleans.

- MAMAExpDLL mama_status mamaMsgField_getVectorChar (const mamaMsgField msgField, const char **result, mama_size_t *size)

  Get a vector of characters.

- MAMAExpDLL mama_status mamaMsgField_getVectorI8 (const mamaMsgField msgField, const mama_i8_t **result, mama_size_t *size)

  Get a vector of signed 8 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorU8 (const mamaMsgField msgField, const mama_u8_t **result, mama_size_t *size)

  Get a vector of unsigned 8 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorI16 (const mamaMsgField msgField, const mama_i16_t **result, mama_size_t *size)

  Get a vector of signed 16 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorU16 (const mamaMsgField msgField, const mama_u16_t **result, mama_size_t *size)

Get a vector of unsigned 16 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorI32 (const mamaMsgField msgField, const mama_i32_t **result, mama_size_t *size)

  Get a vector of signed 32 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorU32 (const mamaMsgField msgField, const mama_u32_t **result, mama_size_t *size)

  Get a vector of unsigned 32 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorI64 (const mamaMsgField msgField, const mama_i64_t **result, mama_size_t *size)

  Get a vector of signed 64 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorU64 (const mamaMsgField msgField, const mama_u64_t **result, mama_size_t *size)

  Get a vector of unsigned 64 bit integers.

- MAMAExpDLL mama_status mamaMsgField_getVectorF32 (const mamaMsgField msgField, const mama_f32_t **result, mama_size_t *size)

  Get a vector of 32 bit floating point numbers.

- MAMAExpDLL mama_status mamaMsgField_getVectorF64 (const mamaMsgField msgField, const mama_f64_t **result, mama_size_t *size)

  Get a vector of 64 bit floating point numbers.

- MAMAExpDLL mama_status mamaMsgField_getVectorString (const mamaMsgField msgField, const char ***result, mama_size_t *size)

  Get a vector of strings (char).*

- MAMAExpDLL mama_status mamaMsgField_getVectorMsg (const mamaMsgField msgField, const mamaMsg **result, mama_size_t *size)

  Get a vector of submessages.

- MAMAExpDLL mama_status mamaMsgField_getAsString (const mamaMsgField field, char *buf, mama_size_t length)

  Convert the value of the specified field to a string.

- MAMAExpDLL mama_status mamaMsgField_updateBool (mamaMsgField field, mama_bool_t value)

  Update the specified field with a new bool value.

- MAMAExpDLL mama_status mamaMsgField_updateChar (mamaMsgField field, char value)

Update the specified field with a new char value.

- MAMAExpDLL mama_status mamaMsgField_updateI8 (mamaMsgField field, mama_i8_t value)

  Update the specified field with a new I8 value.

- MAMAExpDLL mama_status mamaMsgField_updateU8 (mamaMsgField field, mama_u8_t value)

  Update the specified field with a new U8 value.

- MAMAExpDLL mama_status mamaMsgField_updateI16 (mamaMsgField field, mama_i16_t value)

  Update the specified field with a new I16 value.

- MAMAExpDLL mama_status mamaMsgField_updateU16 (mamaMsgField field, mama_u16_t value)

  Update the specified field with a new U16 value.

- MAMAExpDLL mama_status mamaMsgField_updateI32 (mamaMsgField field, mama_i32_t value)

  Update the specified field with a new I32 value.

- MAMAExpDLL mama_status mamaMsgField_updateU32 (mamaMsgField field, mama_u32_t value)

  Update the specified field with a new U32 value.

- MAMAExpDLL mama_status mamaMsgField_updateI64 (mamaMsgField field, mama_i64_t value)

  Update the specified field with a new I64 value.

- MAMAExpDLL mama_status mamaMsgField_updateU64 (mamaMsgField field, mama_u64_t value)

  Update the specified field with a new U64 value.

- MAMAExpDLL mama_status mamaMsgField_updateF32 (mamaMsgField field, mama_f32_t value)

  Update the specified field with a new F32 value.

- MAMAExpDLL mama_status mamaMsgField_updateF64 (mamaMsgField field, mama_f64_t value)

  Update the specified field with a new F64 value.

- MAMAExpDLL mama_status mamaMsgField_updateDateTime (mamaMsgField field, const mamaDateTime value)

Update the specified field with a new MamaDateTime value.

- MAMAExpDLL mama_status mamaMsgField_updatePrice (mamaMsgField field, const mamaPrice value)

  Update the specified field with a new MamaDateTime value.

5.17.1 Function Documentation

5.17.1.1 MAMAExpDLL mama_status mamaMsgField_create (mamaMsgField * *msgField*)

Create a field.

Parameters:

msgField A pointer to the result.

5.17.1.2 MAMAExpDLL mama_status mamaMsgField_destroy (mamaMsgField *msgField*)

Destroy a message and free any resources associated with it.

Parameters:

msgField The field.

5.17.1.3 MAMAExpDLL mama_status mamaMsgField_getDescriptor (const mamaMsgField *msgField*, mamaFieldDescriptor * *result*)

Get the field descriptor for this field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.4 MAMAExpDLL mama_status mamaMsgField_getFid (const mamaMsgField *msgField*, mama_fid_t * *result*)

Get the field ID for this field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.5 MAMAExpDLL mama_status mamaMsgField_getName (const mamaMsgField *msgField*, const char ** *result*)

Get the field name for this field.

Parameters:

msgField The field.

result (out) Pointer to the result.

5.17.1.6 MAMAExpDLL mama_status mamaMsgField_getType (const mamaMsgField *msgField*, mamaFieldType * *result*)

Get the field type for this field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.7 MAMAExpDLL mama_status mamaMsgField_getTypeName (const mamaMsgField *msgField*, const char ** *result*)

Get the type name for this field.

This is a human readable representation of the type.

Parameters:

msgField The field.

result (out) Pointer to the result.

5.17.1.8 MAMAExpDLL mama_status mamaMsgField_getBool (const mamaMsgField *msgField*, mama_bool_t * *result*)

Get a boolean field.

Parameters:

msgField The field.
result (out) Pointer to the result

5.17.1.9 MAMAExpDLL mama_status mamaMsgField_getChar (const mamaMsgField *msgField*, char ∗ *result*)

Get a character field.

Parameters:

msgField The field.
result (out) Pointer to the result

5.17.1.10 MAMAExpDLL mama_status mamaMsgField_getI8 (const mamaMsgField *msgField*, mama_i8_t ∗ *result*)

Get a I8, signed 8 bit integer, field.

Parameters:

msgField The field.
result (out) Pointer to the result

5.17.1.11 MAMAExpDLL mama_status mamaMsgField_getU8 (const mamaMsgField *msgField*, mama_u8_t ∗ *result*)

Get a U8, unsigned 8 bit integer, field.

Parameters:

msgField The field.
result (out) Pointer to the result

5.17.1.12 MAMAExpDLL mama_status mamaMsgField_getI16 (const mamaMsgField *msgField*, mama_i16_t ∗ *result*)

Get a I16, signed 16 bit integer, field.

Parameters:

msgField The field.
result (out) Pointer to the result

5.17.1.13 MAMAExpDLL mama_status mamaMsgField_getU16 (const mamaMsgField *msgField*, mama_u16_t * *result*)

Get a U16, unsigned 16 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.14 MAMAExpDLL mama_status mamaMsgField_getI32 (const mamaMsgField *msgField*, mama_i32_t * *result*)

Get a I32, signed 32 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.15 MAMAExpDLL mama_status mamaMsgField_getU32 (const mamaMsgField *msgField*, mama_u32_t * *result*)

Get a U32, unsigned 32 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.16 MAMAExpDLL mama_status mamaMsgField_getI64 (const mamaMsgField *msgField*, mama_i64_t * *result*)

Get a I64, signed 64 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.17 MAMAExpDLL mama_status mamaMsgField_getU64 (const mamaMsgField *msgField*, mama_u64_t * *result*)

Get a U64, unsigned 64 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.18 MAMAExpDLL mama_status mamaMsgField_getF32 (const mamaMsgField *msgField*, mama_f32_t * *result*)

Get a F32, floating point 32 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.19 MAMAExpDLL mama_status mamaMsgField_getF64 (const mamaMsgField *msgField*, mama_f64_t * *result*)

Get a F64, floating point 64 bit integer, field.

Parameters:

msgField The field.

result (out) Pointer to the result

5.17.1.20 MAMAExpDLL mama_status mamaMsgField_getString (const mamaMsgField *msgField*, const char ** *result*)

Get a const char * field.

Parameters:

msgField The field.

result (out) the string value.

5.17.1.21 MAMAExpDLL mama_status mamaMsgField_getOpaque (const mamaMsgField *msgField*, const void ∗∗ *result*, mama_size_t ∗ *size*)

Get an opaque field.

Parameters:

msgField The field.

result (out) The opaque value.

size The size of the resulting opaque value.

5.17.1.22 MAMAExpDLL mama_status mamaMsgField_getDateTime (const mamaMsgField *msgField*, mamaDateTime *result*)

Get a MAMA date/time field.

Parameters:

msgField The field.

result (out) the output value.

5.17.1.23 MAMAExpDLL mama_status mamaMsgField_getPrice (const mamaMsgField *msgField*, mamaPrice *result*)

Get a MAMA price field.

Parameters:

msgField The field.

result (out) the output value.

5.17.1.24 MAMAExpDLL mama_status mamaMsgField_getMsg (const mamaMsgField *msgField*, mamaMsg ∗ *result*)

Get a MAMA submessage field.

Parameters:

msgField The field.

result (out) the mamaMsg output value.

5.17.1.25 MAMAExpDLL mama_status mamaMsgField_getVectorBool (const mamaMsgField *msgField*, const mama_bool_t ∗∗ *result*, mama_size_t ∗ *size*)

Get a vector of booleans.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.26 MAMAExpDLL mama_status mamaMsgField_getVectorChar (const mamaMsgField *msgField*, const char ∗∗ *result*, mama_size_t ∗ *size*)

Get a vector of characters.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.27 MAMAExpDLL mama_status mamaMsgField_getVectorI8 (const mamaMsgField *msgField*, const mama_i8_t ∗∗ *result*, mama_size_t ∗ *size*)

Get a vector of signed 8 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.28 MAMAExpDLL mama_status mamaMsgField_getVectorU8 (const mamaMsgField *msgField*, const mama_u8_t ** *result*, mama_size_t * *size*)

Get a vector of unsigned 8 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.29 MAMAExpDLL mama_status mamaMsgField_getVectorI16 (const mamaMsgField *msgField*, const mama_i16_t ** *result*, mama_size_t * *size*)

Get a vector of signed 16 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.30 MAMAExpDLL mama_status mamaMsgField_getVectorU16 (const mamaMsgField *msgField*, const mama_u16_t ** *result*, mama_size_t * *size*)

Get a vector of unsigned 16 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.31 MAMAExpDLL mama_status mamaMsgField_getVectorI32 (const mamaMsgField *msgField*, const mama_i32_t ** *result*, mama_size_t * *size*)

Get a vector of signed 32 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.32 MAMAExpDLL mama_status mamaMsgField_getVectorU32 (const mamaMsgField *msgField*, const mama_u32_t ** *result*, mama_size_t * *size*)

Get a vector of unsigned 32 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.33 MAMAExpDLL mama_status mamaMsgField_getVectorI64 (const mamaMsgField *msgField*, const mama_i64_t ** *result*, mama_size_t * *size*)

Get a vector of signed 64 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.34 MAMAExpDLL mama_status mamaMsgField_getVectorU64 (const mamaMsgField *msgField*, const mama_u64_t ** *result*, mama_size_t * *size*)

Get a vector of unsigned 64 bit integers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.35 MAMAExpDLL mama_status mamaMsgField_getVectorF32 (const mamaMsgField *msgField*, const mama_f32_t ** *result*, mama_size_t * *size*)

Get a vector of 32 bit floating point numbers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.36 MAMAExpDLL mama_status mamaMsgField_getVectorF64 (const mamaMsgField *msgField*, const mama_f64_t ** *result*, mama_size_t * *size*)

Get a vector of 64 bit floating point numbers.

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.37 MAMAExpDLL mama_status mamaMsgField_getVectorString (const mamaMsgField *msgField*, const char ∗∗∗ *result*, mama_size_t ∗ *size*)

Get a vector of strings (char∗).

Parameters:

msgField The message field.

result A pointer to the result.

size An integer pointer that will contain the length of the result when the method returns..

5.17.1.38 MAMAExpDLL mama_status mamaMsgField_getVectorMsg (const mamaMsgField *msgField*, const mamaMsg ∗∗ *result*, mama_size_t ∗ *size*)

Get a vector of submessages.

Parameters:

msgField The field.

result A pointer to the result.

size A pointer for the length of the result.

5.17.1.39 MAMAExpDLL mama_status mamaMsgField_getAsString (const mamaMsgField *field*, char ∗ *buf*, mama_size_t *length*)

Convert the value of the specified field to a string.

Caller must provide a buffer for the value.

Parameters:

field The field.

buf Buffer where the result will be copied.

length The length of the caller supplied buffer.

5.17.1.40 MAMAExpDLL mama_status mamaMsgField_updateBool (mamaMsgField *field*, mama_bool_t *value*)

Update the specified field with a new bool value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type bool.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.41 MAMAExpDLL mama_status mamaMsgField_updateChar (mamaMsgField *field*, char *value*)

Update the specified field with a new char value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type char.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.42 MAMAExpDLL mama_status mamaMsgField_updateI8 (mamaMsgField *field*, mama_i8_t *value*)

Update the specified field with a new I8 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type I8.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.43 MAMAExpDLL mama_status mamaMsgField_updateU8 (mamaMsgField *field*, mama_u8_t *value*)

Update the specified field with a new U8 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type U8.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.44 MAMAExpDLL mama_status mamaMsgField_updateI16 (mamaMsgField *field*, mama_i16_t *value*)

Update the specified field with a new I16 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type I16.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.45 MAMAExpDLL mama_status mamaMsgField_updateU16 (mamaMsgField *field*, mama_u16_t *value*)

Update the specified field with a new U16 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type U16.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.46 MAMAExpDLL mama_status mamaMsgField_updateI32 (mamaMsgField *field*, mama_i32_t *value*)

Update the specified field with a new I32 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type I32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.47 MAMAExpDLL mama_status mamaMsgField_updateU32 (mamaMsgField *field*, mama_u32_t *value*)

Update the specified field with a new U32 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type U32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.48 MAMAExpDLL mama_status mamaMsgField_updateI64 (mamaMsgField *field*, mama_i64_t *value*)

Update the specified field with a new I64 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type I64.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.49 MAMAExpDLL mama_status mamaMsgField_updateU64 (mamaMsgField *field*, mama_u64_t *value*)

Update the specified field with a new U64 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type U64.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.50 MAMAExpDLL mama_status mamaMsgField_updateF32 (mamaMsgField *field*, mama_f32_t *value*)

Update the specified field with a new F32 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type F32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.51 MAMAExpDLL mama_status mamaMsgField_updateF64 (mamaMsgField *field*, mama_f64_t *value*)

Update the specified field with a new F64 value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type F32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.52 MAMAExpDLL mama_status mamaMsgField_updateDateTime (mamaMsgField *field*, const mamaDateTime *value*)

Update the specified field with a new MamaDateTime value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type F32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.17.1.53 MAMAExpDLL mama_status mamaMsgField_updatePrice (mamaMsgField *field*, const mamaPrice *value*)

Update the specified field with a new MamaDateTime value.

Returns MAMA_WRONG_FIELD_TYPE if the existing field is not of type F32.

Parameters:

field The mamaMsgField to be updated.

value The new value for the field.

Returns:

MAMA_STATUS_OK The function returned successfully.
MAMA_STATUS_WRONG_FIELD_TYPE The existing field type does not match the type of the update function called.
MAMA_STATUS_NULL_ARG The field passed to the function is NULL.
MAMA_STATUS_INVALID_ARG The underlying bridge field is NULL.

5.18 msgqualifier.h File Reference

```
#include <mama/config.h>
#include <mama/types.h>
#include <mama/status.h>
```

Defines

- #define MAMA_MSG_QUAL_MAX_STR_LEN (100)
- #define MAMA_MSG_QUAL_POSSIBLY_DUPLICATE 0x0001
- #define MAMA_MSG_QUAL_DEFINITELY_DUPLICATE 0x0002
- #define MAMA_MSG_QUAL_POSSIBLY_DELAYED 0x0004
- #define MAMA_MSG_QUAL_DEFINITELY_DELAYED 0x0008
- #define MAMA_MSG_QUAL_OUT_OF_SEQUENCE 0x0010
- #define MAMA_MSG_QUAL_NOT_USED 0x1000

Functions

- MAMAExpDLL mama_status mamaMsgQual_create (mamaMsgQual *msgQual)

  Create a mamaMsgQual object.

- MAMAExpDLL mama_status mamaMsgQual_destroy (mamaMsgQual msgQual)

  Destroy a mamaMsgQual object.

- MAMAExpDLL mama_status mamaMsgQual_clear (mamaMsgQual msgQual)

  Clear a mamaMsgQual object.

- MAMAExpDLL mama_status mamaMsgQual_copy (mamaMsgQual dest, const mamaMsgQual src)

  Copy a mamaMsgQual object.

- MAMAExpDLL int mamaMsgQual_equal (const mamaMsgQual lhs, const mamaMsgQual rhs)

  Check for equality between two objects.

- MAMAExpDLL mama_status mamaMsgQual_setValue (mamaMsgQual msgQual, mama_u16_t value)

  Set the value.

- MAMAExpDLL mama_status mamaMsgQual_setIsDefinitelyDuplicate (mamaMsgQual msgQual, int state)

  Set the bit associated with the Definitely Duplicate condition.

- MAMAExpDLL mama_status mamaMsgQual_setIsPossiblyDuplicate (mamaMsgQual msgQual, int state)

  Set the bit associated with the Possibly Duplicate condition.

- MAMAExpDLL mama_status mamaMsgQual_setIsDefinitelyDelayed (mamaMsgQual msgQual, int state)

  Set the bit associated with the Definitely Delayed condition.

- MAMAExpDLL mama_status mamaMsgQual_setIsPossiblyDelayed (mamaMsgQual msgQual, int state)

  Set the bit associated with the Possibly Delayed condition.

- MAMAExpDLL mama_status mamaMsgQual_setIsOutOfSequence (mamaMsgQual msgQual, int state)

  Set the bit associated with the Out Of Sequence condition.

- MAMAExpDLL mama_status mamaMsgQual_getIsDefinitelyDuplicate (mamaMsgQual msgQual, int *result)

  Determine whether the message qualifier indicates that the "msg" is Definitely Duplicate.

- MAMAExpDLL mama_status mamaMsgQual_getIsPossiblyDuplicate (mamaMsgQual msgQual, int *result)

  Determine whether the message qualifier indicates that the "msg" is Possibly Duplicate.

- MAMAExpDLL mama_status mamaMsgQual_getIsDefinitelyDelayed (mamaMsgQual msgQual, int *result)

  Determine whether the message qualifier indicates that the "msg" is Definitely Delayed.

- MAMAExpDLL mama_status mamaMsgQual_getIsPossiblyDelayed (mamaMsgQual msgQual, int *result)

  Determine whether the message qualifier indicates that the "msg" is Possibly Delayed.

- MAMAExpDLL mama_status mamaMsgQual_getIsOutOfSequence (mamaMsgQual msgQual, int *result)

  Determine whether the message qualifier indicates that the "msg" is Out Of Sequence Delayed.

- MAMAExpDLL mama_status mamaMsgQual_getValue (const mamaMsgQual msgQual, mama_u16_t *value)

  Get the message qualifier value.

- MAMAExpDLL mama_status mamaMsgQual_getAsString (const mamaMsgQual msgQual, const char *delimiter, char *str, mama_size_t maxLen)

  Get the message qualifier value as a delimited string.

- MAMAExpDLL mama_status mamaMsgQual_valueToString (mama_u16_t value, const char *delimiter, char *str, mama_size_t maxLen)

  Get the message qualifier value as a delimited string.

5.18.1 Define Documentation

5.18.1.1 #define MAMA_MSG_QUAL_MAX_STR_LEN (100)

5.18.1.2 #define MAMA_MSG_QUAL_POSSIBLY_DUPLICATE 0x0001

5.18.1.3 #define MAMA_MSG_QUAL_DEFINITELY_DUPLICATE 0x0002

5.18.1.4 #define MAMA_MSG_QUAL_POSSIBLY_DELAYED 0x0004

5.18.1.5 #define MAMA_MSG_QUAL_DEFINITELY_DELAYED 0x0008

5.18.1.6 #define MAMA_MSG_QUAL_OUT_OF_SEQUENCE 0x0010

5.18.1.7 #define MAMA_MSG_QUAL_NOT_USED 0x1000

5.18.2 Function Documentation

5.18.2.1 MAMAExpDLL mama_status mamaMsgQual_create (mamaMsgQual * *msgQual*)

Create a mamaMsgQual object.

Parameters:

msgQual The location of a mamaMsgQual where to store the result.

5.18.2.2 MAMAExpDLL mama_status mamaMsgQual_destroy (mamaMsgQual *msgQual*)

Destroy a mamaMsgQual object.

Parameters:

msgQual The object to destroy.

5.18.2.3 MAMAExpDLL mama_status mamaMsgQual_clear (mamaMsgQual *msgQual*)

Clear a mamaMsgQual object.

Parameters:

msgQual The object to clear.

5.18.2.4 MAMAExpDLL mama_status mamaMsgQual_copy (mamaMsgQual *dest*, const mamaMsgQual *src*)

Copy a mamaMsgQual object.

The destination object must have already been allocated using mamaMsgQual_-create().

Parameters:

dest The destination mamaMsgQual.

src The mamaMsgQual to copy.

5.18.2.5 MAMAExpDLL int mamaMsgQual_equal (const mamaMsgQual *lhs*, const mamaMsgQual *rhs*)

Check for equality between two objects.

Parameters:

lhs The first object to compare.

rhs The second object to compare.

5.18.2.6 MAMAExpDLL mama_status mamaMsgQual_setValue (mamaMsgQual *msgQual*, mama_u16_t *value*)

Set the value.

Parameters:

msgQual The object to set.

value The value to set to.

5.18.2.7 MAMAExpDLL mama_status mamaMsgQual_setIsDefinitelyDuplicate (mamaMsgQual *msgQual*, int *state*)

Set the bit associated with the Definitely Duplicate condition.

Parameters:

msgQual The object to set.

state Zero value clears Definitely Duplicate bit. Non-zero value sets Definitely Duplicate bit.

5.18.2.8 MAMAExpDLL mama_status mamaMsgQual_setIsPossiblyDuplicate (mamaMsgQual *msgQual*, int *state*)

Set the bit associated with the Possibly Duplicate condition.

Parameters:

msgQual The object to set.

state Zero value clears Possibly Duplicate bit. Non-zero value sets Possibly Duplicate bit.

5.18.2.9 MAMAExpDLL mama_status mamaMsgQual_setIsDefinitelyDelayed (mamaMsgQual *msgQual*, int *state*)

Set the bit associated with the Definitely Delayed condition.

Parameters:

msgQual The object to set.

state Zero value clears Definitely Delayed bit. Non-zero value sets Definitely Delayed bit.

5.18.2.10 MAMAExpDLL mama_status mamaMsgQual_setIsPossiblyDelayed (mamaMsgQual *msgQual*, int *state*)

Set the bit associated with the Possibly Delayed condition.

Parameters:

msgQual The object to set.

state Zero value clears Possibly Delayed bit. Non-zero value sets Possibly Delayed bit.

5.18.2.11 MAMAExpDLL mama_status mamaMsgQual_setIsOutOfSequence (mamaMsgQual *msgQual*, int *state*)

Set the bit associated with the Out Of Sequence condition.

Parameters:

msgQual The object to set.

state Zero value clears Out Of Sequence bit. Non-zero value sets Out Of Sequence bit.

5.18.2.12 MAMAExpDLL mama_status mamaMsgQual_getIsDefinitelyDuplicate (mamaMsgQual *msgQual*, int * *result*)

Determine whether the message qualifier indicates that the "msg" is Definitely Duplicate.

Parameters:

msgQual The mamaMsgQual object.

result (out) Pointer to the result. "1" indicates a positive result. "0" indicates a negative result.

5.18.2.13 MAMAExpDLL mama_status mamaMsgQual_getIsPossiblyDuplicate (mamaMsgQual *msgQual*, int * *result*)

Determine whether the message qualifier indicates that the "msg" is Possibly Duplicate.

Parameters:

msgQual The mamaMsgQual object.

result (out) Pointer to the result. "1" indicates a positive result. "0" indicates a negative result.

5.18.2.14 MAMAExpDLL mama_status mamaMsgQual_getIsDefinitelyDelayed (mamaMsgQual *msgQual*, int * *result*)

Determine whether the message qualifier indicates that the "msg" is Definitely Delayed.

Parameters:

msgQual The mamaMsgQual object.

result (out) Pointer to the result. "1" indicates a positive result. "0" indicates a negative result.

5.18.2.15 MAMAExpDLL mama_status mamaMsgQual_getIsPossiblyDelayed (mamaMsgQual *msgQual*, int * *result*)

Determine whether the message qualifier indicates that the "msg" is Possibly Delayed.

Parameters:

msgQual The mamaMsgQual object.

result (out) Pointer to the result. "1" indicates a positive result. "0" indicates a negative result.

5.18.2.16 MAMAExpDLL mama_status mamaMsgQual_getIsOutOfSequence (mamaMsgQual *msgQual*, int * *result*)

Determine whether the message qualifier indicates that the "msg" is Out Of Sequence Delayed.

Parameters:

msgQual The mamaMsgQual object.

result (out) Pointer to the result. "1" indicates a positive result. "0" indicates a negative result.

5.18.2.17 MAMAExpDLL mama_status mamaMsgQual_getValue (const mamaMsgQual *msgQual*, mama_u16_t * *value*)

Get the message qualifier value.

Parameters:

msgQual The mamaMsgQual object.

value The integer value of the message qualifier.

5.18.2.18 MAMAExpDLL mama_status mamaMsgQual_getAsString (const mamaMsgQual *msgQual*, const char * *delimiter*, char * *str*, mama_size_t *maxLen*)

Get the message qualifier value as a delimited string.

Parameters:

msgQual The mamaMsgQual object.

delimiter The delimiter to separate individual condition strings.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.18.2.19 MAMAExpDLL mama_status mamaMsgQual_valueToString (mama_u16_t *value*, const char * *delimiter*, char * *str*, mama_size_t *maxLen*)

Get the message qualifier value as a delimited string.

Parameters:

value The value of the message qualifier.

delimiter The delimiter to separate individual conditions strings.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.19 msgstatus.h File Reference

```
#include <mama/msg.h>
```

Typedefs

- typedef enum mamaMsgStatus_ mamaMsgStatus

  Utility class (enum) for interpreting message status.

Enumerations

- enum mamaMsgStatus_ {

  MAMA_MSG_STATUS_OK = 0, MAMA_MSG_STATUS_LINE_DOWN = 1, MAMA_MSG_STATUS_NO_SUBSCRIBERS = 2, MAMA_MSG_STATUS_-BAD_SYMBOL = 3,

  MAMA_MSG_STATUS_EXPIRED = 4, MAMA_MSG_STATUS_TIMEOUT = 5, MAMA_MSG_STATUS_MISC = 6, MAMA_MSG_STATUS_STALE = 7,

  MAMA_MSG_STATUS_TIBRV_STATUS = 8, MAMA_MSG_STATUS_-PLATFORM_STATUS = 8, MAMA_MSG_STATUS_NOT_ENTITLED = 9, MAMA_MSG_STATUS_NOT_FOUND = 10,

  MAMA_MSG_STATUS_POSSIBLY_STALE = 11, MAMA_MSG_STATUS_-NOT_PERMISSIONED = 12, MAMA_MSG_STATUS_TOPIC_CHANGE = 13, MAMA_MSG_STATUS_BANDWIDTH_EXCEEDED = 14,

  MAMA_MSG_STATUS_DUPLICATE = 15, MAMA_MSG_STATUS_-UNKNOWN = 99 }

  Utility class (enum) for interpreting message status.

Functions

- MAMAExpDLL mamaMsgStatus mamaMsgStatus_statusForMsg (const mamaMsg msg)

  Extract the status from the supplied message.

- MAMAExpDLL const char * mamaMsgStatus_stringForMsg (const mamaMsg msg)

  Return the status as a string given a message.

- MAMAExpDLL const char * mamaMsgStatus_stringForStatus (mamaMsg-Status type)

Return a text description of the message's status.

5.19.1 Typedef Documentation

5.19.1.1 typedef enum mamaMsgStatus_ mamaMsgStatus

Utility class (enum) for interpreting message status.

5.19.2 Enumeration Type Documentation

5.19.2.1 enum mamaMsgStatus_

Utility class (enum) for interpreting message status.

Enumerator:

MAMA_MSG_STATUS_OK OK.

MAMA_MSG_STATUS_LINE_DOWN The feed handler has detected a Line Down.

MAMA_MSG_STATUS_NO_SUBSCRIBERS The feed handler does not have any subscribers to the subject.

MAMA_MSG_STATUS_BAD_SYMBOL The symbol does not exist.

MAMA_MSG_STATUS_EXPIRED Expired.

MAMA_MSG_STATUS_TIMEOUT A time out occurred.

MAMA_MSG_STATUS_MISC Miscellaneous status.

Not an error

MAMA_MSG_STATUS_STALE The subject is stale.

Messages may have been dropped

MAMA_MSG_STATUS_TIBRV_STATUS

MAMA_MSG_STATUS_PLATFORM_STATUS Error in the underlying messaging API.

MAMA_MSG_STATUS_NOT_ENTITLED Not entitled to a subject.

MAMA_MSG_STATUS_NOT_FOUND Not found.

MAMA_MSG_STATUS_POSSIBLY_STALE Messages may have been dropped.

MAMA_MSG_STATUS_NOT_PERMISSIONED Not permissioned for the subject.

MAMA_MSG_STATUS_TOPIC_CHANGE Topic renamed.

MAMA_MSG_STATUS_BANDWIDTH_EXCEEDED Bandwidth exceeded.

MAMA_MSG_STATUS_DUPLICATE Message with duplicate sequence number.

MAMA_MSG_STATUS_UNKNOWN Unknown status.

```
{
    MAMA_MSG_STATUS_OK                  =   0,

    MAMA_MSG_STATUS_LINE_DOWN           =   1,

    MAMA_MSG_STATUS_NO_SUBSCRIBERS      =   2,

    MAMA_MSG_STATUS_BAD_SYMBOL          =   3,

    MAMA_MSG_STATUS_EXPIRED             =   4,

    MAMA_MSG_STATUS_TIMEOUT             =   5,

    MAMA_MSG_STATUS_MISC                =   6,

    MAMA_MSG_STATUS_STALE               =   7,

    MAMA_MSG_STATUS_TIBRV_STATUS        =   8,

    MAMA_MSG_STATUS_PLATFORM_STATUS     =   8,

    MAMA_MSG_STATUS_NOT_ENTITLED        =   9,

    MAMA_MSG_STATUS_NOT_FOUND           =   10,

    MAMA_MSG_STATUS_POSSIBLY_STALE      =   11,

    MAMA_MSG_STATUS_NOT_PERMISSIONED    =   12,

    MAMA_MSG_STATUS_TOPIC_CHANGE        =   13,

    MAMA_MSG_STATUS_BANDWIDTH_EXCEEDED  =   14,

    MAMA_MSG_STATUS_DUPLICATE           =   15,

    MAMA_MSG_STATUS_UNKNOWN             =   99

} mamaMsgStatus;
```

5.19.3 Function Documentation

5.19.3.1 MAMAExpDLL mamaMsgStatus mamaMsgStatus_statusForMsg (const mamaMsg *msg*)

Extract the status from the supplied message.

Parameters:

msg The message.

5.19.3.2 MAMAExpDLL const char* mamaMsgStatus_stringForMsg (const mamaMsg *msg*)

Return the status as a string given a message.

Returns:

The string.

5.19.3.3 MAMAExpDLL const char* mamaMsgStatus_stringForStatus (mamaMsgStatus *type*)

Return a text description of the message's status.

Returns:

The description.

5.20 msgtype.h File Reference

```
#include "mama/types.h"
```

Enumerations

- enum mamaMsgType {

  MAMA_MSG_TYPE_UPDATE = 0, MAMA_MSG_TYPE_INITIAL = 1, MAMA_MSG_TYPE_CANCEL = 2, MAMA_MSG_TYPE_ERROR = 3,

  MAMA_MSG_TYPE_CORRECTION = 4, MAMA_MSG_TYPE_CLOSING = 5, MAMA_MSG_TYPE_RECAP = 6, MAMA_MSG_TYPE_DELETE = 7,

  MAMA_MSG_TYPE_EXPIRE = 8, MAMA_MSG_TYPE_SNAPSHOT = 9, MAMA_MSG_TYPE_PREOPENING = 12, MAMA_MSG_TYPE_QUOTE = 13,

  MAMA_MSG_TYPE_TRADE = 14, MAMA_MSG_TYPE_ORDER = 15, MAMA_MSG_TYPE_BOOK_INITIAL = 16, MAMA_MSG_TYPE_BOOK_-UPDATE = 17,

  MAMA_MSG_TYPE_BOOK_CLEAR = 18, MAMA_MSG_TYPE_BOOK_-RECAP = 19, MAMA_MSG_TYPE_BOOK_SNAPSHOT = 20, MAMA_-MSG_TYPE_NOT_PERMISSIONED = 21,

  MAMA_MSG_TYPE_NOT_FOUND = 22, MAMA_MSG_TYPE_END_OF_-INITIALS = 23, MAMA_MSG_TYPE_WOMBAT_REQUEST = 24, MAMA_-MSG_TYPE_WOMBAT_CALC = 25,

  MAMA_MSG_TYPE_SEC_STATUS = 26, MAMA_MSG_TYPE_DDICT_-SNAPSHOT = 50, MAMA_MSG_TYPE_MISC = 100, MAMA_MSG_TYPE_-TIBRV = 101,

  MAMA_MSG_TYPE_FEATURE_SET = 150, MAMA_MSG_TYPE_SYNC_-REQUEST = 170, MAMA_MSG_TYPE_REFRESH = 171, MAMA_MSG_-TYPE_WORLD_VIEW = 172,

  MAMA_MSG_TYPE_NEWS_QUERY = 173, MAMA_MSG_TYPE_NULL = 175, MAMA_MSG_TYPE_ENTITLEMENTS_REFRESH = 176, MAMA_-MSG_TYPE_UNKNOWN = 199 }

Functions

- MAMAExpDLL mamaMsgType mamaMsgType_typeForMsg (const mamaMsg msg)

  Extract the type from the message.

- MAMAExpDLL const char * mamaMsgType_stringForMsg (const mamaMsg msg)

Extract the type as a string from a message.

- MAMAExpDLL const char * mamaMsgType_stringForType (mamaMsgType type)

  Convert a mamaMsgType to a string.

5.20.1 Enumeration Type Documentation

5.20.1.1 enum mamaMsgType

Enumerator:

MAMA_MSG_TYPE_UPDATE General update (includes, funds).

For direct feeds, the handler sends more specific message types for stocks and order books; however, aggregated feeds will send MAMA_MSG_TYPE_-UPDATE for stocks and order books as well.

MAMA_MSG_TYPE_INITIAL Initial value.

The initial image (full record) for normal subscriptions. The initial value is the first message (several messages for multi-part initialvalues) to arrive with all of the information in the cache. Multi-part initial values only occur on Mama for TIBRV. If the requiresInitial parameter to createSubscription is 0, no initial value is sent.

MAMA_MSG_TYPE_CANCEL Trade cancel.

MAMA_MSG_TYPE_ERROR An error occurred .

MAMA_MSG_TYPE_CORRECTION Trade correction.

MAMA_MSG_TYPE_CLOSING Closing summary.

MAMA_MSG_TYPE_RECAP Refresh/recap of some/all fields.

When the client detects a sequence number gap, it requests a recap from the feed handler. The feed handler may also send recaps in the event of a correction or other event that requires publishing the full record.

MAMA_MSG_TYPE_DELETE The feed handler will not send any more updates for the symbol, to the client in question.

MAMA_MSG_TYPE_EXPIRE Expired option or future.

MAMA_MSG_TYPE_SNAPSHOT A snapshot is the same as an initial value; however, the client will not receive any subsequent updates.

Clients request snapshots by creating snapshot subscriptions.

MAMA_MSG_TYPE_PREOPENING Pre-opening summary (e.g.

morning "roll").

MAMA_MSG_TYPE_QUOTE Quote updates.

MAMA_MSG_TYPE_TRADE Trade updates.

MAMA_MSG_TYPE_ORDER Order updates.

MAMA_MSG_TYPE_BOOK_INITIAL Order book initial value.

This message is sent rather than MAMA_MSG_TYPE_INITIAL for order books.

MAMA_MSG_TYPE_BOOK_UPDATE Order book update.

Sent for order books rather than MAMA_MSG_TYPE_UPDATE.

MAMA_MSG_TYPE_BOOK_CLEAR Order book clear.

All the entries should be removed from the book.

MAMA_MSG_TYPE_BOOK_RECAP Order book recap.

Sent rather than MAMA_MSG_TYPE_RECAP for order books

MAMA_MSG_TYPE_BOOK_SNAPSHOT Order book recap.

Sent rather than MAMA_MSG_TYPE_SNAPSHOT for order books

MAMA_MSG_TYPE_NOT_PERMISSIONED Not permissioned on the feed.

MAMA_MSG_TYPE_NOT_FOUND The symbols was not found but may show up later.

This indicates that the symbol is not currently in the feed handler's cache, but may get added later. The feed handler must be configured with OrderBookNotFoundAction and RecordNotFoundAction set to not_found to enable this behavior.

MAMA_MSG_TYPE_END_OF_INITIALS End of group of initial values.

Marks the last initial value for group subscriptions.

MAMA_MSG_TYPE_WOMBAT_REQUEST A service request.

MAMA_MSG_TYPE_WOMBAT_CALC A calculated result.

MAMA_MSG_TYPE_SEC_STATUS Security status update.

MAMA_MSG_TYPE_DDICT_SNAPSHOT Data dictionary.

This message contains the data dictionary.

MAMA_MSG_TYPE_MISC Miscellaneous.

MAMA_MSG_TYPE_TIBRV Returned if an RV error is encountered the MsgStatus will be the tibrv_status.

MAMA_MSG_TYPE_FEATURE_SET The set of features and related params for a particular publisher.

MAMA_MSG_TYPE_SYNC_REQUEST Subscription synchronization request.

MAMA_MSG_TYPE_REFRESH Subscription refresh.

MAMA_MSG_TYPE_WORLD_VIEW World View request.

MAMA_MSG_TYPE_NEWS_QUERY News query.

MAMA_MSG_TYPE_NULL Keep alive message.

MAMA_MSG_TYPE_ENTITLEMENTS_REFRESH

MAMA_MSG_TYPE_UNKNOWN Unknown, not covered by any of the above.

```
36 {
43     MAMA_MSG_TYPE_UPDATE            =   0,
44
53     MAMA_MSG_TYPE_INITIAL           =   1,
54
56     MAMA_MSG_TYPE_CANCEL            =   2,
57
59     MAMA_MSG_TYPE_ERROR             =   3,
60
62     MAMA_MSG_TYPE_CORRECTION        =   4,
63
65     MAMA_MSG_TYPE_CLOSING           =   5,
66
73     MAMA_MSG_TYPE_RECAP             =   6,
74
79     MAMA_MSG_TYPE_DELETE            =   7,
80
82     MAMA_MSG_TYPE_EXPIRE            =   8,
83
89     MAMA_MSG_TYPE_SNAPSHOT          =   9,
90
92     MAMA_MSG_TYPE_PREOPENING        =   12,
93
95     MAMA_MSG_TYPE_QUOTE             =   13,
96
98     MAMA_MSG_TYPE_TRADE             =   14,
99
101     MAMA_MSG_TYPE_ORDER             =   15,
102
107     MAMA_MSG_TYPE_BOOK_INITIAL      =   16,
108
112     MAMA_MSG_TYPE_BOOK_UPDATE       =   17,
113
115     MAMA_MSG_TYPE_BOOK_CLEAR        =   18,
116
119     MAMA_MSG_TYPE_BOOK_RECAP        =   19,
120
123     MAMA_MSG_TYPE_BOOK_SNAPSHOT     =   20,
124
126     MAMA_MSG_TYPE_NOT_PERMISSIONED  =   21,
127
135     MAMA_MSG_TYPE_NOT_FOUND         =   22,
136
141     MAMA_MSG_TYPE_END_OF_INITIALS   =   23,
142
144     MAMA_MSG_TYPE_WOMBAT_REQUEST    =   24,
145
147     MAMA_MSG_TYPE_WOMBAT_CALC       =   25,
148
150     MAMA_MSG_TYPE_SEC_STATUS        =   26,
151
153     MAMA_MSG_TYPE_DDICT_SNAPSHOT    =   50,
154
156     MAMA_MSG_TYPE_MISC              =   100,
157
```

```
    MAMA_MSG_TYPE_TIBRV             =   101,

    MAMA_MSG_TYPE_FEATURE_SET           =   150,

    MAMA_MSG_TYPE_SYNC_REQUEST          =   170,

    MAMA_MSG_TYPE_REFRESH               =   171,

    MAMA_MSG_TYPE_WORLD_VIEW            =   172,

    MAMA_MSG_TYPE_NEWS_QUERY            =   173,

    MAMA_MSG_TYPE_NULL                  =   175,

    MAMA_MSG_TYPE_ENTITLEMENTS_REFRESH  =   176,

    MAMA_MSG_TYPE_UNKNOWN               =   199
} mamaMsgType;
```

5.20.2 Function Documentation

5.20.2.1 MAMAExpDLL mamaMsgType mamaMsgType_typeForMsg (const mamaMsg *msg*)

Extract the type from the message.

Parameters:

msg The message.

5.20.2.2 MAMAExpDLL const char* mamaMsgType_stringForMsg (const mamaMsg *msg*)

Extract the type as a string from a message.

Parameters:

msg The message.

5.20.2.3 MAMAExpDLL const char* mamaMsgType_stringForType (mamaMsgType *type*)

Convert a mamaMsgType to a string.

Parameters:

type The mamaMsgType.

5.21 price.h File Reference

```
#include <mama/config.h>
#include <mama/status.h>
#include <mama/types.h>
```

Defines

- #define MAMA_PRICE_HINTS_NONE ((mamaPriceHints)0x00)
- #define MAMA_PRICE_MAX_STR_LEN (32)
- #define MAMA_PRICE_EPSILON ((mama_f64_t)0.00000000001)

  MAMA_PRICE_EPSILON a value that is considered equivalent to zero.

Typedefs

- typedef enum mamaPricePrecision_ mamaPricePrecision
- typedef mama_u8_t mamaPriceHints

Enumerations

- enum mamaPricePrecision_ {

  MAMA_PRICE_PREC_UNKNOWN = 0, MAMA_PRICE_PREC_10 = 1, MAMA_PRICE_PREC_100 = 2, MAMA_PRICE_PREC_1000 = 3,

  MAMA_PRICE_PREC_10000 = 4, MAMA_PRICE_PREC_100000 = 5, MAMA_PRICE_PREC_1000000 = 6, MAMA_PRICE_PREC_10000000 = 7,

  MAMA_PRICE_PREC_100000000 = 8, MAMA_PRICE_PREC_1000000000 = 9, MAMA_PRICE_PREC_10000000000 = 10, MAMA_PRICE_PREC_INT = 16,

  MAMA_PRICE_PREC_DIV_2 = 17, MAMA_PRICE_PREC_DIV_4 = 18, MAMA_PRICE_PREC_DIV_8 = 19, MAMA_PRICE_PREC_DIV_16 = 20,

  MAMA_PRICE_PREC_DIV_32 = 21, MAMA_PRICE_PREC_DIV_64 = 22, MAMA_PRICE_PREC_DIV_128 = 23, MAMA_PRICE_PREC_DIV_256 = 24,

  MAMA_PRICE_PREC_DIV_512 = 25, MAMA_PRICE_PREC_TICK_32 = 26, MAMA_PRICE_PREC_HALF_32 = 27, MAMA_PRICE_PREC_-QUARTER_32 = 28,

  MAMA_PRICE_PREC_TICK_64 = 29, MAMA_PRICE_PREC_HALF_64 = 30, MAMA_PRICE_PREC_CENTS = MAMA_PRICE_PREC_100, MAMA_-PRICE_PREC_PENNIES = MAMA_PRICE_PREC_100 }

Functions

- MAMAExpDLL mama_status mamaPrice_create (mamaPrice *price)

  Create a mamaPrice object.

- MAMAExpDLL mama_status mamaPrice_destroy (mamaPrice price)

  Destroy a mamaPrice object.

- MAMAExpDLL mama_status mamaPrice_clear (mamaPrice price)

  Clear a mamaPrice object.

- MAMAExpDLL mama_status mamaPrice_copy (mamaPrice dest, const mamaPrice src)

  Copy a price object.

- MAMAExpDLL mama_status mamaPrice_add (mamaPrice dest, const mamaPrice rhs)

  Add to a price object.

- MAMAExpDLL mama_status mamaPrice_subtract (mamaPrice dest, const mamaPrice rhs)

  Subtract from a price object.

- MAMAExpDLL int mamaPrice_equal (const mamaPrice lhs, const mamaPrice rhs)

  Check for equality between two price objects.

- MAMAExpDLL double mamaPrice_compare (const mamaPrice lhs, const mamaPrice rhs)

  Compare two price objects.

- MAMAExpDLL mama_status mamaPrice_setValue (mamaPrice price, double value)

  Set the price value only (no hint information).

- MAMAExpDLL mama_status mamaPrice_setPrecision (mamaPrice price, mamaPricePrecision precision)

  Set the price precision.

- MAMAExpDLL mama_status mamaPrice_setIsValidPrice (mamaPrice price, mama_bool_t valid)

  Set if price contains a valid value.

- MAMAExpDLL mama_status mamaPrice_setHints (mamaPrice price, mamaPriceHints hints)

  Set all of the price hints.

- MAMAExpDLL mama_status mamaPrice_setWithHints (mamaPrice price, double value, mamaPriceHints hints)

  Set the price with hint information.

- MAMAExpDLL mama_status mamaPrice_setFromString (mamaPrice price, const char *str)

  Set the price from a string representation, preserving any detectable hints.

- MAMAExpDLL mama_status mamaPrice_getValue (const mamaPrice price, double *value)

  Get the price value only.

- MAMAExpDLL mama_status mamaPrice_getRoundedValue (const mamaPrice price, const mamaPricePrecision precision, double *value)

  This function obtains the double value of a price rounded to the supplied precision.

- MAMAExpDLL mama_status mamaPrice_getPrecision (const mamaPrice price, mamaPricePrecision *precision)

  Get the price precision.

- MAMAExpDLL mama_status mamaPrice_getIsValidPrice (const mamaPrice price, mama_bool_t *valid)

  Check if price contains a valid value.

- MAMAExpDLL mama_status mamaPrice_getHints (const mamaPrice price, mamaPriceHints *hints)

  Get the price hints.

- MAMAExpDLL mama_status mamaPrice_getWithHints (const mamaPrice price, double *value, mamaPricePrecision *precision)

  Get the price with special, optional hints.

- MAMAExpDLL mama_status mamaPrice_getAsString (const mamaPrice price, char *str, mama_size_t maxLen)

  Get the price as a string, formatted according to the hints provided.

- MAMAExpDLL mama_status mamaPrice_negate (mamaPrice price)

  Negate the price value.

- MAMAExpDLL mama_status mamaPrice_isZero (const mamaPrice price, mama_bool_t *result)

  Return whether the price has a value equivalent to zero.

- MAMAExpDLL mamaPricePrecision mamaPrice_decimals2Precision (mama_i32_t places)

  Return the appropriate precision code for a given number of decimal places.

- MAMAExpDLL mamaPricePrecision mamaPrice_denom2Precision (mama_i32_t denominator)

  Return the appropriate precision code for a given fractional denominator.

- MAMAExpDLL mama_i32_t mamaPrice_precision2Decimals (mamaPricePrecision precision)

  Return the number of decimal places for a given precision code.

- MAMAExpDLL mama_i32_t mamaPrice_precision2Denom (mamaPricePrecision precision)

  Return the fractional denominator for a given precision code.

5.21.1 Define Documentation

5.21.1.1 #define MAMA_PRICE_HINTS_NONE ((mamaPriceHints)0x00)

5.21.1.2 #define MAMA_PRICE_MAX_STR_LEN (32)

5.21.1.3 #define MAMA_PRICE_EPSILON ((mama_f64_t)0.00000000001)

MAMA_PRICE_EPSILON a value that is considered equivalent to zero.

5.21.2 Typedef Documentation

5.21.2.1 typedef enum mamaPricePrecision_ mamaPricePrecision

5.21.2.2 typedef mama_u8_t mamaPriceHints

5.21.3 Enumeration Type Documentation

5.21.3.1 enum mamaPricePrecision_

Enumerator:

MAMA_PRICE_PREC_UNKNOWN

MAMA_PRICE_PREC_10

MAMA_PRICE_PREC_100

MAMA_PRICE_PREC_1000

MAMA_PRICE_PREC_10000

MAMA_PRICE_PREC_100000

MAMA_PRICE_PREC_1000000

MAMA_PRICE_PREC_10000000

MAMA_PRICE_PREC_100000000

MAMA_PRICE_PREC_1000000000

MAMA_PRICE_PREC_10000000000

MAMA_PRICE_PREC_INT

MAMA_PRICE_PREC_DIV_2

MAMA_PRICE_PREC_DIV_4

MAMA_PRICE_PREC_DIV_8

MAMA_PRICE_PREC_DIV_16

MAMA_PRICE_PREC_DIV_32

MAMA_PRICE_PREC_DIV_64

MAMA_PRICE_PREC_DIV_128

MAMA_PRICE_PREC_DIV_256

MAMA_PRICE_PREC_DIV_512

MAMA_PRICE_PREC_TICK_32

MAMA_PRICE_PREC_HALF_32

MAMA_PRICE_PREC_QUARTER_32

MAMA_PRICE_PREC_TICK_64

MAMA_PRICE_PREC_HALF_64

MAMA_PRICE_PREC_CENTS

MAMA_PRICE_PREC_PENNIES

```
{
    MAMA_PRICE_PREC_UNKNOWN      = 0,
    MAMA_PRICE_PREC_10           = 1,
    MAMA_PRICE_PREC_100          = 2,
    MAMA_PRICE_PREC_1000         = 3,
    MAMA_PRICE_PREC_10000        = 4,
    MAMA_PRICE_PREC_100000       = 5,
    MAMA_PRICE_PREC_1000000      = 6,
    MAMA_PRICE_PREC_10000000     = 7,
    MAMA_PRICE_PREC_100000000    = 8,
    MAMA_PRICE_PREC_1000000000   = 9,
    MAMA_PRICE_PREC_10000000000  = 10,
```

```
    MAMA_PRICE_PREC_INT          = 16,
    MAMA_PRICE_PREC_DIV_2        = 17,
    MAMA_PRICE_PREC_DIV_4        = 18,
    MAMA_PRICE_PREC_DIV_8        = 19,
    MAMA_PRICE_PREC_DIV_16       = 20,
    MAMA_PRICE_PREC_DIV_32       = 21,
    MAMA_PRICE_PREC_DIV_64       = 22,
    MAMA_PRICE_PREC_DIV_128      = 23,
    MAMA_PRICE_PREC_DIV_256      = 24,
    MAMA_PRICE_PREC_DIV_512      = 25,
    MAMA_PRICE_PREC_TICK_32      = 26,
    MAMA_PRICE_PREC_HALF_32      = 27,
    MAMA_PRICE_PREC_QUARTER_32   = 28,
    MAMA_PRICE_PREC_TICK_64      = 29,
    MAMA_PRICE_PREC_HALF_64      = 30,
    MAMA_PRICE_PREC_CENTS        = MAMA_PRICE_PREC_100,
    MAMA_PRICE_PREC_PENNIES      = MAMA_PRICE_PREC_100
} mamaPricePrecision;
```

5.21.4 Function Documentation

5.21.4.1 MAMAExpDLL mama_status mamaPrice_create (mamaPrice * *price*)

Create a mamaPrice object.

Parameters:

price The location of a mamaPrice where to store the result.

5.21.4.2 MAMAExpDLL mama_status mamaPrice_destroy (mamaPrice *price*)

Destroy a mamaPrice object.

Parameters:

price The price object to destroy.

5.21.4.3 MAMAExpDLL mama_status mamaPrice_clear (mamaPrice *price*)

Clear a mamaPrice object.

Parameters:

price The price object to clear.

5.21.4.4 MAMAExpDLL mama_status mamaPrice_copy (mamaPrice *dest*, const mamaPrice *src*)

Copy a price object.

The destination object must have already been allocated using mamaPrice_create().

Parameters:

copy The destination price.

src The price to copy.

5.21.4.5 MAMAExpDLL mama_status mamaPrice_add (mamaPrice *dest*, const mamaPrice *rhs*)

Add to a price object.

The destination object must have already been allocated using mamaPrice_create().

Parameters:

dest The price to add to.

rhs The price to add.

5.21.4.6 MAMAExpDLL mama_status mamaPrice_subtract (mamaPrice *dest*, const mamaPrice *rhs*)

Subtract from a price object.

The destination object must have already been allocated using mamaPrice_create().

Parameters:

dest The price to subtract from.

src The price to subtract.

5.21.4.7 MAMAExpDLL int mamaPrice_equal (const mamaPrice *lhs*, const mamaPrice *rhs*)

Check for equality between two price objects.

Parameters:

lhs The first price to compare.

rhs The second price to compare.

5.21.4.8 MAMAExpDLL double mamaPrice_compare (const mamaPrice *lhs*, const mamaPrice *rhs*)

Compare two price objects.

The return value is negative if lhs is earlier than rhs, positive if lhs is greater than rhs and zero if the two are equal.

Parameters:

lhs The first price to compare.

rhs The second price to compare.

5.21.4.9 MAMAExpDLL mama_status mamaPrice_setValue (mamaPrice *price*, double *value*)

Set the price value only (no hint information).

Parameters:

price The price to set.

value The value to set.

5.21.4.10 MAMAExpDLL mama_status mamaPrice_setPrecision (mamaPrice *price*, mamaPricePrecision *precision*)

Set the price precision.

Parameters:

price The price to set.

precision The precision for the price.

5.21.4.11 MAMAExpDLL mama_status mamaPrice_setIsValidPrice (mamaPrice *price*, mama_bool_t *valid*)

Set if price contains a valid value.

(0 may be a valid value)

Parameters:

price The price to set.

valid If price contains a valid value.

5.21.4.12 MAMAExpDLL mama_status mamaPrice_setHints (mamaPrice *price*, mamaPriceHints *hints*)

Set all of the price hints.

Parameters:

price The price to set.

hints The hints for the price.

5.21.4.13 MAMAExpDLL mama_status mamaPrice_setWithHints (mamaPrice *price*, double *value*, mamaPriceHints *hints*)

Set the price with hint information.

Parameters:

price The price to set.

value The value to set.

hints The price hints.

5.21.4.14 MAMAExpDLL mama_status mamaPrice_setFromString (mamaPrice *price*, const char * *str*)

Set the price from a string representation, preserving any detectable hints.

Parameters:

price The price to set.

str The string representation of some price.

5.21.4.15 MAMAExpDLL mama_status mamaPrice_getValue (const mamaPrice *price*, double * *value*)

Get the price value only.

Parameters:

price The price to get.

value The the decimal value of the price.

5.21.4.16 MAMAExpDLL mama_status mamaPrice_getRoundedValue (const mamaPrice *price*, const mamaPricePrecision *precision*, double * *value*)

This function obtains the double value of a price rounded to the supplied precision.

Parameters:

← ***price*** The price to round.

← ***precision*** The precision to use for rounding.

→ ***value*** The rounded value.

Returns:

Indicates whether the function succeeded or failed and could be one of:

- MAMA_STATUS_INVALID_ARG
- MAMA_STATUS_OK

5.21.4.17 MAMAExpDLL mama_status mamaPrice_getPrecision (const mamaPrice *price*, mamaPricePrecision * *precision*)

Get the price precision.

Parameters:

price The price to get.

precision The price precision hint.

5.21.4.18 MAMAExpDLL mama_status mamaPrice_getIsValidPrice (const mamaPrice *price*, mama_bool_t * *valid*)

Check if price contains a valid value.

Parameters:

price The price to get.

valid If the value is valid.

5.21.4.19 MAMAExpDLL mama_status mamaPrice_getHints (const mamaPrice *price*, mamaPriceHints * *hints*)

Get the price hints.

Parameters:

price The price to get.

hints The price hints.

5.21.4.20 MAMAExpDLL mama_status mamaPrice_getWithHints (const mamaPrice *price*, double * *value*, mamaPricePrecision * *precision*)

Get the price with special, optional hints.

Parameters:

price The price to get.

value The decimal value of the price.

precision The precision of the price.

5.21.4.21 MAMAExpDLL mama_status mamaPrice_getAsString (const mamaPrice *price*, char * *str*, mama_size_t *maxLen*)

Get the price as a string, formatted according to the hints provided.

Parameters:

price The price to set.

str The string buffer to update.

maxLen The maximum size of the string buffer (including trailing '\0').

5.21.4.22 MAMAExpDLL mama_status mamaPrice_negate (mamaPrice *price*)

Negate the price value.

Hints and precisions are not affected.

5.21.4.23 MAMAExpDLL mama_status mamaPrice_isZero (const mamaPrice *price*, mama_bool_t * *result*)

Return whether the price has a value equivalent to zero.

It may not be exactly 0.0, but we check against +/- epsilon.

5.21.4.24 MAMAExpDLL mamaPricePrecision mamaPrice_-decimals2Precision (mama_i32_t *places*)

Return the appropriate precision code for a given number of decimal places.

5.21.4.25 MAMAExpDLL mamaPricePrecision mamaPrice_denom2Precision (mama_i32_t *denominator*)

Return the appropriate precision code for a given fractional denominator.

5.21.4.26 MAMAExpDLL mama_i32_t mamaPrice_precision2Decimals (mamaPricePrecision *precision*)

Return the number of decimal places for a given precision code.

5.21.4.27 MAMAExpDLL mama_i32_t mamaPrice_precision2Denom (mamaPricePrecision *precision*)

Return the fractional denominator for a given precision code.

5.22 publisher.h File Reference

Typedefs

- typedef mamaMsg msg
- typedef mamaMsg mama_status status
- typedef mamaMsg mama_status void ∗ closure

Functions

- typedef void (MAMACALLTYPE ∗mamaThrottledSendCompleteCb)(mamaPublisher publisher

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

- MAMAExpDLL mama_status mamaPublisher_create (mamaPublisher ∗result, mamaTransport tport, const char ∗symbol, const char ∗source, const char ∗root)

  Create a mama publisher for the corresponding transport.

- MAMAExpDLL mama_status mamaPublisher_send (mamaPublisher publisher, mamaMsg msg)

  Send a message from the specified publisher.

- MAMAExpDLL mama_status mamaPublisher_sendWithThrottle (mamaPublisher publisher, mamaMsg msg, mamaThrottledSendCompleteCb sendCompleteCb, void ∗closure)

  Send a message with the throttle.

- MAMAExpDLL mama_status mamaPublisher_sendFromInboxWithThrottle (mamaPublisher publisher, mamaInbox inbox, mamaMsg msg, mamaThrottledSendCompleteCb sendCompleteCb, void ∗closure)
- MAMAExpDLL mama_status mamaPublisher_sendFromInbox (mamaPublisher publisher, mamaInbox inbox, mamaMsg msg)
- MAMAExpDLL mama_status mamaPublisher_sendReplyToInboxHandle (mamaPublisher publisher, mamaMsgReply replyAddress, mamaMsg reply)

  Send a reply in response to a request to an inbox.

- MAMAExpDLL mama_status mamaPublisher_sendReplyToInbox (mamaPublisher publisher, mamaMsg request, mamaMsg reply)

  Send a reply in response to a request to an inbox.

- MAMAExpDLL mama_status mamaPublisher_destroy (mamaPublisher publisher)

  Destroy the publisher.

5.22.1 Typedef Documentation

5.22.1.1 typedef mamaMsg msg

5.22.1.2 typedef mamaMsg mama_status status

5.22.1.3 typedef mamaMsg mama_status void∗ closure

5.22.2 Function Documentation

5.22.2.1 typedef void (MAMACALLTYPE ∗ *mamaThrottledSendCompleteCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

Messages sent on the throttle queue are no longer destroyed by the API. It is the responsibility of the application developer to manage the lifecycle of any messages sent on the throttle.

Parameters:

publisher The publisher object used to send the message.

msg The mamaMsg which has been sent from the throttle queue.

status Whether the message was successfully sent from the throttle. A value of MAMA_STATUS_OK indicates that the send was successful.

closure User supplied context data.

5.22.2.2 MAMAExpDLL mama_status mamaPublisher_create (mamaPublisher ∗ *result*, mamaTransport *tport*, const char ∗ *symbol*, const char ∗ *source*, const char ∗ *root*)

Create a mama publisher for the corresponding transport.

If the transport is a marketdata transport, as opposed to a "basic" transport, the topic corresponds to the symbol. For a basic transport, the source and root get ignored.

Parameters:

result A pointer to hold the resulting mamaPublisher.

tport The transport.

symbol Symbol on which to publish.

source The source for market data publishers. (e.g. source.symbol)

root The root for market data publishers. Used internally.

5.22.2.3 MAMAExpDLL mama_status mamaPublisher_send (mamaPublisher *publisher*, mamaMsg *msg*)

Send a message from the specified publisher.

Parameters:

publisher The publisher from which to send a message.

msg The mamaMsg to send.

5.22.2.4 MAMAExpDLL mama_status mamaPublisher_sendWithThrottle (mamaPublisher *publisher*, mamaMsg *msg*, mamaThrottledSendCompleteCb *sendCompleteCb*, void * *closure*)

Send a message with the throttle.

The lifecycle of the message sent is controlled by the user of the API. The callback indicates when the API is no longer using the message and can be destroyed/reused by the application.

Parameters:

publisher The publish from which to send the throttled message.

msg The mamaMsg to send.

sendCompleteCb Callback invoked once the message has been send on the throttle.

closure User supplied data returned in the send complete callback.

5.22.2.5 MAMAExpDLL mama_status mamaPublisher_sendFromInboxWithThrottle (mamaPublisher *publisher*, mamaInbox *inbox*, mamaMsg *msg*, mamaThrottledSendCompleteCb *sendCompleteCb*, void * *closure*)

5.22.2.6 MAMAExpDLL mama_status mamaPublisher_sendFromInbox (mamaPublisher *publisher*, mamaInbox *inbox*, mamaMsg *msg*)

5.22.2.7 MAMAExpDLL mama_status mamaPublisher_sendReplyToInboxHandle (mamaPublisher *publisher*, mamaMsgReply *replyAddress*, mamaMsg *reply*)

Send a reply in response to a request to an inbox.

Parameters:

publisher The mamaPublisher from which to send the response.

request The reply address to which you are responding.

reply The mamaMsg to be sent as the reply.

5.22.2.8 MAMAExpDLL mama_status mamaPublisher_sendReplyToInbox (mamaPublisher *publisher*, mamaMsg *request*, mamaMsg *reply*)

Send a reply in response to a request to an inbox.

Parameters:

publisher The mamaPublisher from which to send the response.

request The mamaMsg to which you are responding.

reply The mamaMsg to be sent as the reply.

5.22.2.9 MAMAExpDLL mama_status mamaPublisher_destroy (mamaPublisher *publisher*)

Destroy the publisher.

Parameters:

publisher The mamaPublisher to destroy.

5.23 quality.h File Reference

```
#include "mama/config.h"
```

Typedefs

- typedef enum mamaQuality_ mamaQuality

Enumerations

- enum mamaQuality_ {

  MAMA_QUALITY_OK = 0, MAMA_QUALITY_MAYBE_STALE = 1, MAMA_QUALITY_STALE = 2, MAMA_QUALITY_PARTIAL_STALE = 3,

  MAMA_QUALITY_FORCED_STALE = 4, MAMA_QUALITY_-DUPLICATE = 5, MAMA_QUALITY_UNKNOWN = 99 }

Functions

- MAMAExpDLL mamaQuality mamaQuality_convertFromString (const char *str)

  Convert a string to a mamaQuality value.

- MAMAExpDLL const char * mamaQuality_convertToString (mamaQuality quality)

  Convert a mamaQuality value to a string.

5.23.1 Typedef Documentation

5.23.1.1 typedef enum mamaQuality_ mamaQuality

5.23.2 Enumeration Type Documentation

5.23.2.1 enum mamaQuality_

Enumerator:

MAMA_QUALITY_OK

MAMA_QUALITY_MAYBE_STALE

MAMA_QUALITY_STALE

MAMA_QUALITY_PARTIAL_STALE

MAMA_QUALITY_FORCED_STALE

MAMA_QUALITY_DUPLICATE

MAMA_QUALITY_UNKNOWN

```
{
    MAMA_QUALITY_OK             =  0,
    MAMA_QUALITY_MAYBE_STALE    =  1,
    MAMA_QUALITY_STALE          =  2,
    MAMA_QUALITY_PARTIAL_STALE  =  3,
    MAMA_QUALITY_FORCED_STALE   =  4,
    MAMA_QUALITY_DUPLICATE      =  5,
    MAMA_QUALITY_UNKNOWN        = 99
} mamaQuality;
```

5.23.3 Function Documentation

5.23.3.1 MAMAExpDLL mamaQuality mamaQuality_convertFromString (const char * *str*)

Convert a string to a mamaQuality value.

Parameters:

str The str to convert.

5.23.3.2 MAMAExpDLL const char* mamaQuality_convertToString (mamaQuality *quality*)

Convert a mamaQuality value to a string.

Do no attempt to free the string result.

Parameters:

quality The mamaQuality to convert.

5.24 queue.h File Reference

```
#include <mama/mama.h>
#include <mama/types.h>
#include <mama/status.h>
```

Data Structures

- struct mamaQueueMonitorCallbacks_

  callbacks which may be invoked in response to certain conditions on the specified queue being met.

Typedefs

- typedef size_t size
- typedef size_t void * closure
- typedef size_t size
- typedef size_t void * closure
- typedef mamaQueueMonitorCallbacks_ mamaQueueMonitorCallbacks

  callbacks which may be invoked in response to certain conditions on the specified queue being met.

- typedef void(MAMACALLTYPE *) mamaQueueEnqueueCB (mamaQueue queue, void *closure)

  Function invoked when an event is enqueued on the queue for which this function was registered.

- typedef void(MAMACALLTYPE *) mamaQueueEventCB (mamaQueue queue, void *closure)

  Function invoked when a user added event fires.

Functions

- typedef void (MAMACALLTYPE *mamaQueueHighWatermarkExceeded-Cb)(mamaQueue queue

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL mama_status mamaQueue_create (mamaQueue *queue, mamaBridge bridgeImpl)

  Create a queue.

- MAMAExpDLL mama_status mamaQueue_create_usingNative (mamaQueue *queue, mamaBridge bridgeImpl, void *nativeQueue)
- MAMAExpDLL mama_status mamaQueue_canDestroy (mamaQueue queue)

  Check to see if a queue can be destroyed.

- MAMAExpDLL mama_status mamaQueue_destroy (mamaQueue queue)

  Destroy a queue.

- MAMAExpDLL mama_status mamaQueue_destroyWait (mamaQueue queue)

  Destroy a queue.

- MAMAExpDLL mama_status mamaQueue_destroyTimedWait (mamaQueue queue, long timeout)

  Destroy a queue.

- MAMAExpDLL mama_status mamaQueue_setHighWatermark (mamaQueue queue, size_t highWatermark)

  Specify a high watermark for events on the queue.

- MAMAExpDLL mama_status mamaQueue_getHighWatermark (mamaQueue queue, size_t *highWatermark)

  Get the value of the high water mark for the specified queue.

- MAMAExpDLL mama_status mamaQueue_setLowWatermark (mamaQueue queue, size_t lowWatermark)

  Set the low water mark for the queue.

- MAMAExpDLL mama_status mamaQueue_getLowWatermark (mamaQueue queue, size_t *lowWatermark)

  Get the value of the low water mark for the specified queue.

- MAMAExpDLL mama_status mamaQueue_setQueueMonitorCallbacks (mamaQueue queue, mamaQueueMonitorCallbacks *queueMonitorCallbacks, void *closure)

  Specify a set of callbacks which may be invoked in response to certain conditions arising on the queue.

- MAMAExpDLL mama_status mamaQueue_getEventCount (mamaQueue queue, size_t *count)

Writes the number of events currently on the specified queue to the address specified by count.

- MAMAExpDLL mama_status mamaQueue_setQueueName (mamaQueue queue, const char *name)

  Associate a name identifier with the event queue.

- MAMAExpDLL mama_status mamaQueue_getQueueName (mamaQueue queue, const char **name)

  Get the string name identifier for the specified event queue.

- MAMAExpDLL mama_status mamaQueue_getQueueBridgeName (mamaQueue queue, const char **name)

  Get the string name identifier of the bridge for the specified event queue.

- MAMAExpDLL mama_status mamaQueue_dispatch (mamaQueue queue)

  Dispatch messages from the queue.

- MAMAExpDLL mama_status mamaQueue_timedDispatch (mamaQueue queue, uint64_t timeout)

  Dispatch messages from the queue.

- MAMAExpDLL mama_status mamaQueue_dispatchEvent (mamaQueue queue)

  Dispatch a single event from the specified queue.

- MAMAExpDLL mama_status mamaQueue_enqueueEvent (mamaQueue queue, mamaQueueEventCB callback, void *closure)

  Add an user event to a queue.

- MAMAExpDLL mama_status mamaQueue_stopDispatch (mamaQueue queue)

  Unblock the queue as soon as possible.

- MAMAExpDLL mama_status mamaQueue_setEnqueueCallback (mamaQueue queue, mamaQueueEnqueueCB callback, void *closure)

  Register the specified callback function to receive a callback each time an event is enqueued on the specified mamaQueue.

- MAMAExpDLL mama_status mamaQueue_removeEnqueueCallback (mamaQueue queue)

  If the specified queue has a registered enqueue callback it is unregistered and the previously supplied callback function will no longer receive callbacks for enqueue events.

- MAMAExpDLL mama_status mamaQueue_getNativeHandle (mamaQueue queue, void **nativeHandle)

  Get the native middleware implementation queue handle (if applicable for the implementation).

- MAMAExpDLL mama_status mamaDispatcher_create (mamaDispatcher *result, mamaQueue queue)

  Create a mamaDispatcher.

- MAMAExpDLL mama_status mamaDispatcher_getQueue (mamaDispatcher dispatcher, mamaQueue *result)

  Return the queue associated with the dispatcher.

- MAMAExpDLL mama_status mamaQueue_enableStats (mamaQueue queue)

  Enable stats logging on queue.

- MAMAExpDLL mama_status mamaDispatcher_destroy (mamaDispatcher dispatcher)

  Destroy the dispatcher and stop dispatching events.

- MAMAExpDLL mama_status mamaQueue_getClosure (mamaQueue queue, void **closure)
- MAMAExpDLL mama_status mamaQueue_setClosure (mamaQueue queue, void *closure)

5.24.1 Typedef Documentation

5.24.1.1 typedef size_t size

5.24.1.2 typedef size_t void* closure

5.24.1.3 typedef size_t size

5.24.1.4 typedef size_t void* closure

5.24.1.5 typedef struct mamaQueueMonitorCallbacks_ mamaQueueMonitorCallbacks

callbacks which may be invoked in response to certain conditions on the specified queue being met.

5.24.1.6 typedef void(MAMACALLTYPE *) mamaQueueEnqueueCB(mamaQueue queue, void *closure)

Function invoked when an event is enqueued on the queue for which this function was registered.

LBM Bridge: NB! Users may not dispatch events from this function when using with mamaQueue_setEnqueueCallback() The function is invoked from an LBM internal thread. Attempts to dispatch from here will result in a deadlock.

Parameters:

queue The mamaQueue on which the function was registered.

closure The user data supplied in the call to setEnqueueCallback.

5.24.1.7 typedef void(MAMACALLTYPE *) mamaQueueEventCB(mamaQueue queue, void *closure)

Function invoked when a user added event fires.

Events are added to a queue using the `mamaQuque_enqueueEvent()`.

Parameters:

queue The MamaQueue on which the event was enqueued.

closure The user specified data associated with this event.

5.24.2 Function Documentation

5.24.2.1 typedef void (MAMACALLTYPE * *mamaQueueLowWatermarkCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

Parameters:

queue The mamaQueue for which the size limit has been exceeded. NULL if the queue is the default internal MAMA queue.

size The current number of events outstanding on the queue (if supported on the underlying middleware)

closure User supplied data set when the callback was registered. NULL in the case of the default MAMA queue as no closure can be specified when registering the data quality callbacks.

5.24.2.2 MAMAExpDLL mama_status mamaQueue_create (mamaQueue * *queue*, mamaBridge *bridgeImpl*)

Create a queue.

Queues allow applications to dispatch events in order with multiple threads using a single mamaDispatcher for each queue. A queue must be associated with a particular middleware.

Parameters:

queue A pointer to the resulting queue.

bridgeImpl A valid bridge implementation for which this queue is being created.

Returns:

MAMA_STATUS_OK if the call succeeds.
MAMA_STATUS_NO_BRIDGE_IMPL if the bridgeImpl parameter is not valid.

5.24.2.3 MAMAExpDLL mama_status mamaQueue_create_usingNative (mamaQueue * *queue*, mamaBridge *bridgeImpl*, void * *nativeQueue*)

5.24.2.4 MAMAExpDLL mama_status mamaQueue_canDestroy (mamaQueue *queue*)

Check to see if a queue can be destroyed.

The queue cannot be destroyed if there are currently open event objects on it.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the queue can be destroyed. MAMA_STATUS_-QUEUE_OPEN_OBJECTS if there are still objects open against the queue. MAMA_STATUS_NULL_ARG

5.24.2.5 MAMAExpDLL mama_status mamaQueue_destroy (mamaQueue *queue*)

Destroy a queue.

Note that the queue can only be destroyed if all of the objects created on it, (timers, subscriptions etc), have been destroyed.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful. MAMA_STATUS_QUEUE_-OPEN_OBJECTS if there are still objects open against the queue.

5.24.2.6 MAMAExpDLL mama_status mamaQueue_destroyWait (mamaQueue *queue*)

Destroy a queue.

Note that the queue can only be destroyed if all of the objects created on it, (timers, subscriptions etc), have been destroyed. This function will block until all of the objects have been destroyed and will then destroy the queue.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.7 MAMAExpDLL mama_status mamaQueue_destroyTimedWait (mamaQueue *queue*, long *timeout*)

Destroy a queue.

Note that the queue can only be destroyed if all of the objects created on it, (timers, subscriptions etc), have been destroyed. This function will block for the specified time or until all of the objects have been destroyed and will then destroy the queue.

Parameters:

queue The queue.

timeout The time to block for in ms.

Returns:

MAMA_STATUS_OK if the call is successful. MAMA_STATUS_TIMEOUT if the time elapsed.

5.24.2.8 MAMAExpDLL mama_status mamaQueue_setHighWatermark (mamaQueue *queue*, size_t *highWatermark*)

Specify a high watermark for events on the queue.

The behaviour for setting this value varies depending on the underlying middleware.

LBM: LBM uses an unbounded event queue. Setting this values allows users of the API to receive a callback if the value is exceeded. (See mamaQueue_setQueueMonitorCallback() for setting queue related callbacks) The default behaviour is for the queue to grow unbounded without notifications. The high watermark for LBM can be set for all queues at once by setting the mama.lbm.eventqueuemonitor.queue_size_warning property for the API. Calls to this function will override the value specified in mama.properties at runtime. Callbacks can be disabled by setting this value to 0, effectively disabling high watermark checking.

RV: This will set a queue limit policy of TIBRVQUEUE_DISCARD_FIRST whereby the oldest events in the queue are discarded first. The discard amount will be set with a value of 1, i.e. events will be dropped from the queue one at a time. The default behaviour is an unlimited queue which does not discard events.

Parameters:

queue The mamaQueue for which the high watermark is being set.

highWatermark The size of the queue, beyond which, results in notification of activity.

Returns:

MAMA_STATUS_OK if the function returns successfully.

5.24.2.9 MAMAExpDLL mama_status mamaQueue_getHighWatermark (mamaQueue *queue*, size_t * *highWatermark*)

Get the value of the high water mark for the specified queue.

A value of 0 will be returned if no high water mark was previously specified.

Parameters:

queue The mamaQueue for which the high water mark is being retrieved

highWatermark Address to which the high water mark will be written.

5.24.2.10 MAMAExpDLL mama_status mamaQueue_setLowWatermark (mamaQueue *queue*, size_t *lowWatermark*)

Set the low water mark for the queue.

Only supported by Wombat TCP middleware.

The low watermark must be >1 and < highWaterMark otherwise this method returns MAMA_STATUS_INVALID_ARG. For this reason the high water mark must be set before invoking this method.

Parameters:

queue The queue.

lowWatermark the low watermark.

5.24.2.11 MAMAExpDLL mama_status mamaQueue_getLowWatermark (mamaQueue *queue*, size_t * *lowWatermark*)

Get the value of the low water mark for the specified queue.

A value of 1 will be returned if no low water mark was previously specified.

Parameters:

queue The mamaQueue for which the low water mark is being retrieved.

lowWatermark Address to which the low water mark will be written.

5.24.2.12 MAMAExpDLL mama_status mamaQueue_setQueueMonitorCallbacks (mamaQueue *queue*, mamaQueueMonitorCallbacks * *queueMonitorCallbacks*, void * *closure*)

Specify a set of callbacks which may be invoked in response to certain conditions arising on the queue.

The behaviour here is middleware specific as not all will support all callbacks.

LBM: When the high watermark is exceeded the mamaQueueHighWatermarkExceededCb callback will invoked each time an event on the queue is dispatched until such time as the number of events on the queue falls below the high watermark.

5.24.2.13 MAMAExpDLL mama_status mamaQueue_getEventCount (mamaQueue *queue*, size_t * *count*)

Writes the number of events currently on the specified queue to the address specified by count.

Parameters:

queue The queue.

count Address to where the number of events on the queue will be written

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.14 MAMAExpDLL mama_status mamaQueue_setQueueName (mamaQueue *queue*, const char * *name*)

Associate a name identifier with the event queue.

This will be used in queue related logging statements. The string is copied by the API.

Parameters:

queue The event queue for which the name is being specified.

name The string identifier for the queue.

Returns:

MAMA_STATUS_OK The function call succeeded.
MAMA_STATUS_NULL_ARG The queue parameter is NULL
MAMA_STATUS_INVALID_ARG The name parameter is NULL
MAMA_STATUS_NO_MEM The name could not be copied.

5.24.2.15 MAMAExpDLL mama_status mamaQueue_getQueueName (mamaQueue *queue*, const char ** *name*)

Get the string name identifier for the specified event queue.

Parameters:

queue The event queue for which the name is being sought.

name Address to which the name will be written.

Returns:

MAMA_STATUS_OK The function call succeeded.
MAMA_STATUS_NULL_ARG The queue parameter was NULL
MAMA_STATUS_INVALID_ARG The name parameter was NULL

5.24.2.16 MAMAExpDLL mama_status mamaQueue_getQueueBridgeName (mamaQueue *queue*, const char ** *name*)

Get the string name identifier of the bridge for the specified event queue.

Name will be either "wmw", "tibrv", or "lbm".

Parameters:

queue The event queue for which the bridge name is being sought.

name Address to which the name will be written.

Returns:

MAMA_STATUS_OK The function call succeeded.
MAMA_STATUS_NULL_ARG The queue parameter was NULL

5.24.2.17 MAMAExpDLL mama_status mamaQueue_dispatch (mamaQueue *queue*)

Dispatch messages from the queue.

This call blocks and dispatches until mamaQueue_stopDispatch() is called.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.18 MAMAExpDLL mama_status mamaQueue_timedDispatch (mamaQueue *queue*, uint64_t *timeout*)

Dispatch messages from the queue.

This call blocks and dispatches until timeout has elapsed.

Parameters:

queue The queue.

timeout The number of milliseconds to block for before the function returns.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.19 MAMAExpDLL mama_status mamaQueue_dispatchEvent (mamaQueue *queue*)

Dispatch a single event from the specified queue.

If there is no event on the queue simply return and do nothing.

Parameters:

queue The queue from which to dispatch the event.

Returns:

MAMA_STATUS_OK if the function succeeds.

5.24.2.20 MAMAExpDLL mama_status mamaQueue_enqueueEvent (mamaQueue *queue*, mamaQueueEventCB *callback*, void * *closure*)

Add an user event to a queue.

Currently only supported using Wombat Middleware.

Parameters:

queue The queue to which the event is to be added

callback The function to be invoked when the event fires.

closure Optional arbitrary user supplied data. Passed back to callback function.

Returns:

MAMA_STATUS_OK if the function succeeds.

5.24.2.21 MAMAExpDLL mama_status mamaQueue_stopDispatch (mamaQueue *queue*)

Unblock the queue as soon as possible.

This will cause mamaDispatchers to exit. Creating a new dispatcher will resume dispatching events.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.22 MAMAExpDLL mama_status mamaQueue_setEnqueueCallback (mamaQueue *queue*, mamaQueueEnqueueCB *callback*, void * *closure*)

Register the specified callback function to receive a callback each time an event is enqueued on the specified mamaQueue.

Parameters:

queue The mamaQueue on which the callback should be registered.

callback The function which should be invoked for each enqueue operation

closure Optional arbitrary user supplied data. Passed back to callback function.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.23 MAMAExpDLL mama_status mamaQueue_removeEnqueueCallback (mamaQueue *queue*)

If the specified queue has a registered enqueue callback it is unregistered and the previously supplied callback function will no longer receive callbacks for enqueue events.

Parameters:

queue The mamaQueue for which the callback function should be removed.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.24 MAMAExpDLL mama_status mamaQueue_getNativeHandle (mamaQueue *queue*, void ** *nativeHandle*)

Get the native middleware implementation queue handle (if applicable for the implementation).

This function is for internal use only.

Parameters:

queue The mamaQueue for which the native handle is requested.

nativeHandle The resulting native handle.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.25 MAMAExpDLL mama_status mamaDispatcher_create (mamaDispatcher * *result*, mamaQueue *queue*)

Create a mamaDispatcher.

The dispatcher spawns a thread to dispatch events from a queue. It will continue to dispatch events until it is destroyed or mamaQueue_stopDispatch is called.

Only a single dispatcher can be created for a given queue. Attempting to create multiple dispatchers for a queue will result in and error. Dispatching message from a single queue with multiple threads results in messages arriving out of order and sequence number gaps for market data subscriptions.

Parameters:

result A pointer to the resulting mamaDispatcher.

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.26 MAMAExpDLL mama_status mamaDispatcher_getQueue (mamaDispatcher *dispatcher*, mamaQueue * *result*)

Return the queue associated with the dispatcher.

Parameters:

dispatcher The dispatcher.

result The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.27 MAMAExpDLL mama_status mamaQueue_enableStats (mamaQueue *queue*)

Enable stats logging on queue.

Parameters:

queue The queue.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.28 MAMAExpDLL mama_status mamaDispatcher_destroy (mamaDispatcher *dispatcher*)

Destroy the dispatcher and stop dispatching events.

If mamaDispatcher_createQueue() was used then the underlying queue will be destroyed as well.

Parameters:

dispatcher The dispatcher.

Returns:

MAMA_STATUS_OK if the call is successful.

5.24.2.29 MAMAExpDLL mama_status mamaQueue_getClosure (mamaQueue *queue*, void ∗∗ *closure*)

5.24.2.30 MAMAExpDLL mama_status mamaQueue_setClosure (mamaQueue *queue*, void ∗ *closure*)

5.25 reservedfields.h File Reference

```
#include "mama/mama.h"
#include "mama/config.h"
```

Data Structures

- struct MamaReservedField_

Typedefs

- typedef MamaReservedField_ MamaReservedField

Variables

- MAMAExpDLL const long WOMBAT_MAX_RESERVED_FID
- MAMAExpDLL const MamaReservedField MamaFieldMsgType
- MAMAExpDLL const MamaReservedField MamaFieldMsgStatus
- MAMAExpDLL const MamaReservedField MamaFieldFieldIndex
- MAMAExpDLL const MamaReservedField MamaFieldSubscMsgTypeOld
- MAMAExpDLL const MamaReservedField MamaFieldSubscSubjectOld
- MAMAExpDLL const MamaReservedField MamaFieldMsgNum
- MAMAExpDLL const MamaReservedField MamaFieldMsgTotal
- MAMAExpDLL const MamaReservedField MamaFieldTibrvResult
- MAMAExpDLL const MamaReservedField MamaFieldSeqNum
- MAMAExpDLL const MamaReservedField MamaFieldFeedName
- MAMAExpDLL const MamaReservedField MamaFieldFeedHost
- MAMAExpDLL const MamaReservedField MamaFieldFeedGroup
- MAMAExpDLL const MamaReservedField MamaFieldSyncPattern
- MAMAExpDLL const MamaReservedField MamaFieldItemSeqNum
- MAMAExpDLL const MamaReservedField MamaFieldSendTime
- MAMAExpDLL const MamaReservedField MamaFieldAppDataType
- MAMAExpDLL const MamaReservedField MamaFieldAppMsgType
- MAMAExpDLL const MamaReservedField MamaFieldSenderId
- MAMAExpDLL const MamaReservedField MamaFieldMsgQual
- MAMAExpDLL const MamaReservedField MamaFieldConflateCount
- MAMAExpDLL const MamaReservedField MamaFieldConflateQuoteCount
- MAMAExpDLL const MamaReservedField MamaFieldConflateTradeCount
- MAMAExpDLL const MamaReservedField MamaFieldRvDaemon
- MAMAExpDLL const MamaReservedField MamaFieldRvNetwork
- MAMAExpDLL const MamaReservedField MamaFieldRvService

- MAMAExpDLL const MamaReservedField MamaFieldElvinSource
- MAMAExpDLL const MamaReservedField MamaFieldElvinSourceOld
- MAMAExpDLL const MamaReservedField MamaFieldSubscriptionType
- MAMAExpDLL const MamaReservedField MamaFieldSubscMsgType
- MAMAExpDLL const MamaReservedField MamaFieldSubscSourceHost
- MAMAExpDLL const MamaReservedField MamaFieldSubscSourceApp
- MAMAExpDLL const MamaReservedField MamaFieldSubscSourceUser
- MAMAExpDLL const MamaReservedField MamaFieldServiceLevel
- MAMAExpDLL const MamaReservedField MamaFieldSubscSourceIp
- MAMAExpDLL const MamaReservedField MamaFieldSubscSourceAppClass
- MAMAExpDLL const MamaReservedField MamaFieldInterfaceVersion
- MAMAExpDLL const MamaReservedField MamaFieldUpdateTopic
- MAMAExpDLL const MamaReservedField MamaFieldSubscSymbol
- MAMAExpDLL const MamaReservedField MamaFieldTportName
- MAMAExpDLL const MamaReservedField MamaFieldSubscNamespace
- MAMAExpDLL const MamaReservedField MamaFieldSymbolList
- MAMAExpDLL const MamaReservedField MamaFieldTemplate
- MAMAExpDLL const MamaReservedField MamaFieldTemplateLen
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldMsgType
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldMsgStatus
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldFieldIndex
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldMsgNum
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldMsgTotal
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldSeqNum
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldFeedName
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldFeedHost
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldFeedGroup
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldItemSeqNum
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldSendTime
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldAppDataType
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldAppMsgType
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldSenderId
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldMsgQual
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldConflateQuoteCount
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldEntitleCode
- MAMAExpDLL mamaFieldDescriptor MamaReservedFieldSymbolList

5.25.1 Typedef Documentation

5.25.1.1 typedef struct MamaReservedField_ MamaReservedField

5.25.2 Variable Documentation

5.25.2.1 MAMAExpDLL const long WOMBAT_MAX_RESERVED_FID

5.25.2.2 MAMAExpDLL const MamaReservedField MamaFieldMsgType

5.25.2.3 MAMAExpDLL const MamaReservedField MamaFieldMsgStatus

5.25.2.4 MAMAExpDLL const MamaReservedField MamaFieldFieldIndex

5.25.2.5 MAMAExpDLL const MamaReservedField MamaFieldSubscMsgTypeOld

5.25.2.6 MAMAExpDLL const MamaReservedField MamaFieldSubscSubjectOld

5.25.2.7 MAMAExpDLL const MamaReservedField MamaFieldMsgNum

5.25.2.8 MAMAExpDLL const MamaReservedField MamaFieldMsgTotal

5.25.2.9 MAMAExpDLL const MamaReservedField MamaFieldTibrvResult

5.25.2.10 MAMAExpDLL const MamaReservedField MamaFieldSeqNum

5.25.2.11 MAMAExpDLL const MamaReservedField MamaFieldFeedName

5.25.2.12 MAMAExpDLL const MamaReservedField MamaFieldFeedHost

5.25.2.13 MAMAExpDLL const MamaReservedField MamaFieldFeedGroup

5.25.2.14 MAMAExpDLL const MamaReservedField MamaFieldSyncPattern

5.25.2.15 MAMAExpDLL const MamaReservedField MamaFieldItemSeqNum

5.25.2.16 MAMAExpDLL const MamaReservedField MamaFieldSendTime

5.25.2.17 MAMAExpDLL const MamaReservedField MamaFieldAppDataType

5.25.2.18 MAMAExpDLL const MamaReservedField MamaFieldAppMsgType

5.25.2.19 MAMAExpDLL const MamaReservedField MamaFieldSenderId

5.25.2.20 MAMAExpDLL const MamaReservedField MamaFieldMsgQual

5.25.2.21 MAMAExpDLL const MamaReservedField MamaFieldConflateCount

5.25.2.22 MAMAExpDLL const MamaReservedField MamaFieldConflateQuoteCount

5.25.2.23 MAMAExpDLL const MamaReservedField

5.26 senderId.h File Reference

```
#include "mama/mama.h"
```

Functions

- MAMAExpDLL void mamaSenderId_getIpAddr (uint64_t senderId, char *buffer, size_t maxLen)

  Determine the IP address (as a string) of the sender ID in the buffer provided.

- MAMAExpDLL void mamaSenderId_getPid (uint64_t senderId, uint16_t *result)

  Determine the process id in the buffer provided.

- MAMAExpDLL void mamaSenderId_getPid64 (uint64_t senderId, uint32_t *result)
- MAMAExpDLL uint64_t mamaSenderId_getSelf (void)

  Return a sender ID for possible use when publishing messages.

- MAMAExpDLL void mamaSenderId_setSelf (uint64_t self)

  Manually set a sender ID for the process.

5.26.1 Function Documentation

5.26.1.1 MAMAExpDLL void mamaSenderId_getIpAddr (uint64_t *senderId*, char * *buffer*, size_t *maxLen*)

Determine the IP address (as a string) of the sender ID in the buffer provided.

The sender ID is a field sent in many MAMA messages (see the set of reserved fields).

5.26.1.2 MAMAExpDLL void mamaSenderId_getPid (uint64_t *senderId*, uint16_t * *result*)

Determine the process id in the buffer provided.

The sender ID is a field sent in many MAMA messages (see the set of reserved fields).

5.26.1.3 MAMAExpDLL void mamaSenderId_getPid64 (uint64_t *senderId*, uint32_t * *result*)

5.26.1.4 MAMAExpDLL uint64_t mamaSenderId_getSelf (void)

Return a sender ID for possible use when publishing messages.

The sender ID is a field sent in many MAMA messages (see the set of reserved fields).

5.26.1.5 MAMAExpDLL void mamaSenderId_setSelf (uint64_t *self*)

Manually set a sender ID for the process.

If a sender_id is set this way, then as of that point, mamaSenderId_getSelf() will always return that value.

5.27 servicelevel.h File Reference

```
#include "mama/types.h"
```

Typedefs

- typedef enum mamaServiceLevel_ mamaServiceLevel

  mamaServiceLevel is an enumeration that represents the service level (a) requested by a subscribing application, or (b) provided by the publisher.

Enumerations

- enum mamaServiceLevel_ {

  MAMA_SERVICE_LEVEL_REAL_TIME = 0, MAMA_SERVICE_LEVEL_-SNAPSHOT = 1, MAMA_SERVICE_LEVEL_REPEATING_SNAPSHOT = 2, MAMA_SERVICE_LEVEL_CONFLATED = 5,

  MAMA_SERVICE_LEVEL_UNKNOWN = 99 }

  mamaServiceLevel is an enumeration that represents the service level (a) requested by a subscribing application, or (b) provided by the publisher.

Functions

- MAMAExpDLL mamaServiceLevel mamaServiceLevel_getFromMsg (const mamaMsg msg)

  Extract the subscription service level from a message.

- MAMAExpDLL const char * mamaServiceLevel_getFromMsgAsString (const mamaMsg msg)

  Extract the subscription service level as a string from a message.

- MAMAExpDLL const char * mamaServiceLevel_toString (mamaServiceLevel type)

  Convert a mamaServiceLevel to a string.

5.27.1 Typedef Documentation

5.27.1.1 typedef enum mamaServiceLevel_ mamaServiceLevel

mamaServiceLevel is an enumeration that represents the service level (a) requested by a subscribing application, or (b) provided by the publisher.

In the future, the service level provided to the application might change dynamically, depending upon infrastructure capacity. So, even though the requested service level is "real time", if service deteriorates for any reason, the subscription may automatically switch to a conflated level (after notifying the application of the switch).

5.27.2 Enumeration Type Documentation

5.27.2.1 enum mamaServiceLevel_

mamaServiceLevel is an enumeration that represents the service level (a) requested by a subscribing application, or (b) provided by the publisher.

In the future, the service level provided to the application might change dynamically, depending upon infrastructure capacity. So, even though the requested service level is "real time", if service deteriorates for any reason, the subscription may automatically switch to a conflated level (after notifying the application of the switch).

Enumerator:

MAMA_SERVICE_LEVEL_REAL_TIME Real time updates.

MAMA_SERVICE_LEVEL_SNAPSHOT Single snapshot image only.

MAMA_SERVICE_LEVEL_REPEATING_SNAPSHOT Repeating snapshot image.

When creating subscriptions, the serviceLevelOpt parameter should be set to the number of milliseconds representing the interval between snapshots. The first snapshot will be obtained within some random period between zero and the interval. Only one snapshot request will be allowed to be outstanding at a time.

MAMA_SERVICE_LEVEL_CONFLATED Conflated updates (future).

MAMA_SERVICE_LEVEL_UNKNOWN Unknown level.

```
47 {
51     MAMA_SERVICE_LEVEL_REAL_TIME = 0,
52
56     MAMA_SERVICE_LEVEL_SNAPSHOT = 1,
57
66     MAMA_SERVICE_LEVEL_REPEATING_SNAPSHOT = 2,
67
71     MAMA_SERVICE_LEVEL_CONFLATED = 5,
72
```

```
    MAMA_SERVICE_LEVEL_UNKNOWN = 99

} mamaServiceLevel;
```

5.27.3 Function Documentation

5.27.3.1 MAMAExpDLL mamaServiceLevel mamaServiceLevel_getFromMsg (const mamaMsg *msg*)

Extract the subscription service level from a message.

Parameters:

msg The message.

5.27.3.2 MAMAExpDLL const char* mamaServiceLevel_getFromMsgAsString (const mamaMsg *msg*)

Extract the subscription service level as a string from a message.

Parameters:

msg The message.

5.27.3.3 MAMAExpDLL const char* mamaServiceLevel_toString (mamaServiceLevel *type*)

Convert a mamaServiceLevel to a string.

Parameters:

type The mamaServiceLevel.

5.28 source.h File Reference

```
#include "mama/config.h"
#include "mama/status.h"
#include "mama/sourceman.h"
#include "mama/types.h"
#include "mama/quality.h"
```

Typedefs

- typedef enum mamaSourceState_ mamaSourceState

Enumerations

- enum mamaSourceState_ { MAMA_SOURCE_STATE_OFF = 0, MAMA_-SOURCE_STATE_OK = 1, MAMA_SOURCE_STATE_UNKNOWN = 99 }

Functions

- MAMAExpDLL mama_status mamaSource_create (mamaSource *source)

  Create a mamaSource object.

- MAMAExpDLL mama_status mamaSource_destroy (mamaSource source)

  Destroy a mamaSource object.

- MAMAExpDLL mama_status mamaSource_clear (mamaSource source)

  Clear a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setId (mamaSource source, const char *id)

  Set the ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setMappedId (mamaSource source, const char *id)

  Set the mapped ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setDisplayId (mamaSource source, const char *id)

  Set the display ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setQuality (mamaSource source, mamaQuality quality)

  Set the quality of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setState (mamaSource source, mamaSourceState quality)

  Set the state of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setParent (mamaSource source, mamaSource parent)

  Set the parent for a mamaSource object.

- MAMAExpDLL mama_status mamaSource_setSymbolNamespace (mamaSource source, const char *symbolNamespace)

  Set the publisher specific source name for this source.

- MAMAExpDLL mama_status mamaSource_setTransportName (mamaSource source, const char *transportName)

  Set the name of the mamaTransport on which this describes a valid source of data.

- MAMAExpDLL mama_status mamaSource_setTransport (mamaSource source, mamaTransport transport)

  Associate a mamaTransport object with the source.

- MAMAExpDLL mama_status mamaSource_setSymbology (mamaSource source, mamaSymbology symbology)

  Associate a mamaSymbology object with the source.

- MAMAExpDLL mama_status mamaSource_getId (const mamaSource source, const char **id)

  Get the ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_getMappedId (const mamaSource source, const char **id)

  Get the mapped ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_getDisplayId (const mamaSource source, const char **id)

  Get the display ID of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_getQuality (const mamaSource source, mamaQuality *quality)

  Get the quality of a mamaSource object.

- MAMAExpDLL mama_status mamaSource_getState (const mamaSource source, mamaSourceState *state)

  Get the state of a mamaSource object.

- MAMAExpDLL const char * mamaSource_getStateAsString (const mamaSource source)
- MAMAExpDLL mama_status mamaSource_getParent (const mamaSource source, mamaSource *parent)

  Get the parent source for a mamaSource object.

- MAMAExpDLL mama_status mamaSource_getSymbolNamespace (const mamaSource source, const char **symbolNamespace)

  Get the source name for the publisher which this mamaSource represents.

- MAMAExpDLL mama_status mamaSource_getTransportName (const mamaSource source, const char **transportName)

  Get the name of the mamaTransport on which this source is valid.

- MAMAExpDLL mama_status mamaSource_getTransport (const mamaSource source, mamaTransport *transport)

  Get the mamaTransport associated with this source.

- MAMAExpDLL mama_status mamaSource_getSymbology (const mamaSource source, mamaSymbology *symbology)

  Get the mamaSymbology associated with this source.

- MAMAExpDLL mama_status mamaSource_getSubSourceManager (const mamaSource source, mamaSourceManager *subSourceManager)

  A mamaSource can have sub-sources.

5.28.1 Typedef Documentation

5.28.1.1 typedef enum mamaSourceState_ mamaSourceState

5.28.2 Enumeration Type Documentation

5.28.2.1 enum mamaSourceState_

Enumerator:

MAMA_SOURCE_STATE_OFF

MAMA_SOURCE_STATE_OK

MAMA_SOURCE_STATE_UNKNOWN

```
{
    MAMA_SOURCE_STATE_OFF     = 0,
    MAMA_SOURCE_STATE_OK      = 1,
    MAMA_SOURCE_STATE_UNKNOWN = 99
} mamaSourceState;
```

5.28.3 Function Documentation

5.28.3.1 MAMAExpDLL mama_status mamaSource_create (mamaSource * *source*)

Create a mamaSource object.

Parameters:

source The location of a mamaSource to store the result.

5.28.3.2 MAMAExpDLL mama_status mamaSource_destroy (mamaSource *source*)

Destroy a mamaSource object.

Parameters:

source The source object to destroy.

5.28.3.3 MAMAExpDLL mama_status mamaSource_clear (mamaSource *source*)

Clear a mamaSource object.

Parameters:

source The source object to clear.

5.28.3.4 MAMAExpDLL mama_status mamaSource_setId (mamaSource *source*, const char * *id*)

Set the ID of a mamaSource object.

Parameters:

source The source object to update.

id The new ID for the source object.

5.28.3.5 MAMAExpDLL mama_status mamaSource_setMappedId (mamaSource *source*, const char * *id*)

Set the mapped ID of a mamaSource object.

The mapped ID is the ID that the parent source manager has mapped this source as.

Parameters:

source The source object to update.

id The new mapped ID for the source object.

5.28.3.6 MAMAExpDLL mama_status mamaSource_setDisplayId (mamaSource *source*, const char * *id*)

Set the display ID of a mamaSource object.

Parameters:

source The source object to update.

id The new display ID for the source object.

5.28.3.7 MAMAExpDLL mama_status mamaSource_setQuality (mamaSource *source*, mamaQuality *quality*)

Set the quality of a mamaSource object.

Parameters:

source The source object to update.

quality The new quality for the source object.

5.28.3.8 MAMAExpDLL mama_status mamaSource_setState (mamaSource *source*, mamaSourceState *quality*)

Set the state of a mamaSource object.

Parameters:

source The source object to update.

quality The new state for the source object.

5.28.3.9 MAMAExpDLL mama_status mamaSource_setParent (mamaSource *source*, mamaSource *parent*)

Set the parent for a mamaSource object.

The current mamaSource is added to the parents sub source manager.

See also:

mamaSource_getSubSourceManager()

Parameters:

source The source object to update.

parent The parent source for this source object.

5.28.3.10 MAMAExpDLL mama_status mamaSource_setSymbolNamespace (mamaSource *source*, const char * *symbolNamespace*)

Set the publisher specific source name for this source.

e.g. This could be "NASDAQ" for a UTP PAPA publisher.

Parameters:

source The source object to update.

symbolNamespace The namespace for the publisher this mamaSource object describes.

5.28.3.11 MAMAExpDLL mama_status mamaSource_setTransportName (mamaSource *source*, const char * *transportName*)

Set the name of the mamaTransport on which this describes a valid source of data.

Parameters:

source The source object to update.

transportName The name of the mamaTransport for which this source is valid.

5.28.3.12 MAMAExpDLL mama_status mamaSource_setTransport (mamaSource *source*, mamaTransport *transport*)

Associate a mamaTransport object with the source.

Parameters:

source The source to update.

transport The mamaTransport to associate with the source.

5.28.3.13 MAMAExpDLL mama_status mamaSource_setSymbology (mamaSource *source*, mamaSymbology *symbology*)

Associate a mamaSymbology object with the source.

Parameters:

source The source to update.

symbology The mamaSymbology to associate with the source.

5.28.3.14 MAMAExpDLL mama_status mamaSource_getId (const mamaSource *source*, const char ** *id*)

Get the ID of a mamaSource object.

Parameters:

source The source object to check.

id Location of the result for the ID of the source.

5.28.3.15 MAMAExpDLL mama_status mamaSource_getMappedId (const mamaSource *source*, const char ** *id*)

Get the mapped ID of a mamaSource object.

Parameters:

source The source object to check.

id Location of the result for the mapped ID of the source.

5.28.3.16 MAMAExpDLL mama_status mamaSource_getDisplayId (const mamaSource *source*, const char ∗∗ *id*)

Get the display ID of a mamaSource object.

Parameters:

source The source object to check.

id Location of the result for the display ID of the source.

5.28.3.17 MAMAExpDLL mama_status mamaSource_getQuality (const mamaSource *source*, mamaQuality ∗ *quality*)

Get the quality of a mamaSource object.

Parameters:

source The source object to check.

quality Location of the result for the quality of the source.

5.28.3.18 MAMAExpDLL mama_status mamaSource_getState (const mamaSource *source*, mamaSourceState ∗ *state*)

Get the state of a mamaSource object.

Parameters:

source The source object to check.

state Location of the result for the state of the source.

5.28.3.19 MAMAExpDLL const char∗ mamaSource_getStateAsString (const mamaSource *source*)

5.28.3.20 MAMAExpDLL mama_status mamaSource_getParent (const mamaSource *source*, mamaSource ∗ *parent*)

Get the parent source for a mamaSource object.

Parameters:

source The source object to check.

parent Location to store the address of the parent for this source.

5.28.3.21 MAMAExpDLL mama_status mamaSource_getSymbolNamespace (const mamaSource *source*, const char ∗∗ *symbolNamespace*)

Get the source name for the publisher which this mamaSource represents.

Parameters:

source The source object to check.

symbolNamespace The source name for the publisher.

5.28.3.22 MAMAExpDLL mama_status mamaSource_getTransportName (const mamaSource *source*, const char ∗∗ *transportName*)

Get the name of the mamaTransport on which this source is valid.

Parameters:

source The source object to check.

transportName The location to store the pointer to the transportName.

5.28.3.23 MAMAExpDLL mama_status mamaSource_getTransport (const mamaSource *source*, mamaTransport ∗ *transport*)

Get the mamaTransport associated with this source.

Parameters:

source The source object to check.

transport The mamaTransport associated with this source.

5.28.3.24 MAMAExpDLL mama_status mamaSource_getSymbology (const mamaSource *source*, mamaSymbology ∗ *symbology*)

Get the mamaSymbology associated with this source.

Parameters:

source The source object to check.

symbology The mamaSymbology associated with this source.

5.28.3.25 MAMAExpDLL mama_status mamaSource_getSubSourceManager (const mamaSource *source*, mamaSourceManager * *subSourceManager*)

A mamaSource can have sub-sources.

These sub sources are maintained by a mamaSourceManager within the mamaSource.

Parameters:

source The mamaSource for which the sub source manager is being obtained.

subSourceManager The location top store the address of the sub source manager object pointer.

5.29 sourceman.h File Reference

```
#include "mama/config.h"
#include "mama/status.h"
#include "mama/types.h"
#include "mama/quality.h"
#include "mama/log.h"
#include "mama/source.h"
```

Typedefs

- typedef mamaSource source
- typedef mamaSource void * closure

Functions

- typedef void (MAMACALLTYPE *mamaSourceManager_sourcesIterator-Cb)(mamaSourceManager sourceManager

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_-STATUS_OK.

- MAMAExpDLL mama_status mamaSourceManager_create (mamaSource-Manager *sourceManager)

  Create a mamaSourceManager object.

- MAMAExpDLL mama_status mamaSourceManager_destroy (mamaSource-Manager sourceManager)

  Destroy a mamaSourceManager object.

- MAMAExpDLL mama_status mamaSourceManager_createSource (mama-SourceManager sourceManager, const char *name, mamaSource *source)

  Create a new mamaSource and add it to the manager.

- MAMAExpDLL mama_status mamaSourceManager_findOrCreateSource (mamaSourceManager sourceManager, const char *name, mamaSource *source)

  Locates an existing mamaSource for the given name.

- MAMAExpDLL mama_status mamaSourceManager_findSource (mama-SourceManager sourceManager, const char *name, mamaSource *source)

Locates an existing mamaSource in the specified sourceManager with the specified string 'name' identifier.

- MAMAExpDLL mama_status mamaSourceManager_addSource (mamaSourceManager sourceManager, mamaSource source)

  Add an existing mamaSource to the specified mamaSourceManager.

- MAMAExpDLL mama_status mamaSourceManager_addSourceWithName (mamaSourceManager sourceManager, mamaSource source, const char *name)

  Add an existing mamaSource to the specified mamaSourceManager using the specified name as a unique identifier.

- MAMAExpDLL mama_status mamaSourceManager_iterateSources (mamaSourceManager sourceGroup, mamaSourceManager_sourcesIteratorCb callback, void *closure)

  Iterate over all the sources in this mamaSourceManager.

5.29.1 Typedef Documentation

5.29.1.1 typedef mamaSource source

5.29.1.2 typedef mamaSource void* closure

5.29.2 Function Documentation

5.29.2.1 typedef void (MAMACALLTYPE * *mamaSourceManager_sourcesIteratorCb*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

Parameters:

queue The mamaQueue for which the size limit has been exceeded. NULL if the queue is the default internal MAMA queue.

size The current number of events outstanding on the queue (if supported on the underlying middleware)

closure User supplied data set when the callback was registered. NULL in the case of the default MAMA queue as no closure can be specified when registering the data quality callbacks.

5.29.2.2 MAMAExpDLL mama_status mamaSourceManager_create (mamaSourceManager * *sourceManager*)

Create a mamaSourceManager object.

Parameters:

sourceManager The location of a mamaSourceManager to store the result.

5.29.2.3 MAMAExpDLL mama_status mamaSourceManager_destroy (mamaSourceManager *sourceManager*)

Destroy a mamaSourceManager object.

Parameters:

sourceManager The sourceManager object to destroy.

5.29.2.4 MAMAExpDLL mama_status mamaSourceManager_createSource (mamaSourceManager *sourceManager*, const char * *name*, mamaSource * *source*)

Create a new mamaSource and add it to the manager.

Parameters:

sourceManager The sourceManager to use for creating the mamaSource.

name The string identifier for the mamaSource.

source The address to which the new source will be * written.

Returns:

MAMA_STATUS_OK if execution is successful.

5.29.2.5 MAMAExpDLL mama_status mamaSourceManager_findOrCreateSource (mamaSourceManager *sourceManager*, const char * *name*, mamaSource * *source*)

Locates an existing mamaSource for the given name.

If none exists creates a new mamaSource and adds to the sourceManager.

Parameters:

sourceManager The sourceManager to use for locating the mamaSource.

name The string identifier for the mamaSource

source The location to which the address for the source will be written.

Returns:

MAMA_STATUS_OK if execution is successful.

5.29.2.6 MAMAExpDLL mama_status mamaSourceManager_findSource (mamaSourceManager *sourceManager*, const char * *name*, mamaSource * *source*)

Locates an existing mamaSource in the specified sourceManager with the specified string 'name' identifier.

The value of the source argument will be set to NULL if no source was located in the sourceManager provided.

Parameters:

sourceManager The mamaSourceManager to use to locate the specified mamaSource.

name The string identifier for the required mamaSource.

source The location to which the address for the source will be written. NULL if none is found.

Returns:

MAMA_STATUS_OK if creation is successful.

5.29.2.7 MAMAExpDLL mama_status mamaSourceManager_addSource (mamaSourceManager *sourceManager*, mamaSource *source*)

Add an existing mamaSource to the specified mamaSourceManager.

The id of the source will be used instead of the name to uniquely identify the source within the manager.

Parameters:

sourceManager The mamaSourceManager to which an existing mamaSource is being added.

source The mamaSource being added to the specified mamaSourceManager.

Returns:

MAMA_STATUS_OK if execution is successful.

5.29.2.8 MAMAExpDLL mama_status mamaSourceManager_addSourceWithName (mamaSourceManager *sourceManager*, mamaSource *source*, const char * *name*)

Add an existing mamaSource to the specified mamaSourceManager using the specified name as a unique identifier.

Parameters:

sourceManager The mamaSourceManager to which an existing mamaSource is being added.

name The string identifier for the mamaSource

source The mamaSource being added to the specified mamaSourceManager.

Returns:

MAMA_STATUS_OK if execution is successful.

5.29.2.9 MAMAExpDLL mama_status mamaSourceManager_iterateSources (mamaSourceManager *sourceGroup*, mamaSourceManager_sourcesIteratorCb *callback*, void * *closure*)

Iterate over all the sources in this mamaSourceManager.

Parameters:

sourceManager The mamaSourceManager to iterate over.

callback The callback function pointer to invoke for each source in the group.

closure User supplied arbitrary data. Passed back on each invocation of the callback function.

Returns:

MAMA_STATUS_OK if the function executes successfully.

5.30 stat.h File Reference

```
#include "mama/status.h"
```

Defines

- #define MAMA_STAT_NOT_LOCKABLE 0
- #define MAMA_STAT_LOCKABLE 1

Enumerations

- enum mamaStatType {

  MAMA_STAT_TYPE_INITIALS = 105, MAMA_STAT_TYPE_RECAPS = 106, MAMA_STAT_TYPE_NUM_MESSAGES = 107, MAMA_STAT_TYPE_FT_TAKEOVERS = 108,

  MAMA_STAT_TYPE_QUEUE_SIZE = 109, MAMA_STAT_TYPE_SUBSCRIPTIONS = 110, MAMA_STAT_TYPE_TIMEOUTS = 111, MAMA_STAT_TYPE_MSG_INDEX = 112,

  MAMA_STAT_TYPE_NAK_PACKETS_SENT = 113, MAMA_STAT_TYPE_NAKS_SENT = 114, MAMA_STAT_TYPE_MSGS_LOST = 115, MAMA_STAT_TYPE_NCFS_IGNORE = 116,

  MAMA_STAT_TYPE_NCFS_SHED = 117, MAMA_STAT_TYPE_NCFS_RX_DELAY = 118, MAMA_STAT_TYPE_NCFS_UNKNOWN = 119, MAMA_STAT_TYPE_DUPLICATE_MSGS = 120,

  MAMA_STAT_TYPE_UNRECOVERABLE_WINDOW_ADVANCE = 121, MAMA_STAT_TYPE_UNRECOVERABLE_TIMEOUT = 122, MAMA_STAT_TYPE_LBM_MSGS_RECEIVED_NO_TOPIC = 123, MAMA_STAT_TYPE_LBM_REQUESTS_RECEIVED = 124,

  MAMA_STAT_TYPE_WOMBAT_MSGS = 125, MAMA_STAT_TYPE_RV_MSGS = 126, MAMA_STAT_TYPE_FAST_MSGS = 127, MAMA_STAT_TYPE_UNKNOWN_MSGS = 128,

  MAMA_STAT_TYPE_PUBLISHER_SEND = 129, MAMA_STAT_TYPE_PUBLISHER_INBOX_SEND = 130, MAMA_STAT_TYPE_PUBLISHER_REPLY_SEND = 131, MAMA_STAT_TYPE_UNKNOWN = 999 }

Functions

- MAMAExpDLL mama_status mamaStat_create (mamaStat *stat, mamaStatsCollector collector, int lockable, const char *name, mama_fid_t type)

  Create a mamaStats object.

- MAMAExpDLL mama_status mamaStat_destroy (mamaStat stat)

  Destroy a mamaStats object.

- MAMAExpDLL mama_status mamaStat_increment (mamaStat stat)

  Increment the stats object counter.

- MAMAExpDLL mama_status mamaStat_decrement (mamaStat stat)

  Decrement the stats object counter.

- MAMAExpDLL mama_status mamaStat_reset (mamaStat stat)

  Reset the stats object counter.

- MAMAExpDLL mama_status mamaStat_add (mamaStat stat, int value)

  Adds the stats object counter with the value.

- MAMAExpDLL mama_status mamaStat_subtract (mamaStat stat, int value)

  Subtract the stats object counter with the value.

- MAMAExpDLL mama_status mamaStat_setIntervalValue (mamaStat stat, int value)

  set the Interval value

- MAMAExpDLL mama_fid_t mamaStat_getFid (mamaStat stat)

  Get the FID used when publishing the stat via the stats logger.

- MAMAExpDLL int mamaStat_getIntervalValue (mamaStat stat)

  Get the value of the stat for the current interval.

- MAMAExpDLL int mamaStat_getMaxValue (mamaStat stat)

  Get the maximum value of the stat.

- MAMAExpDLL int mamaStat_getTotalValue (mamaStat stat)

  Get the total value of the stat.

- MAMAExpDLL void mamaStat_getStats (mamaStat stat, mama_i32_t *intervalValue, mama_u32_t *maxValue, mama_u32_t *totalValue)

  Get the interval, maximum, and total values for the stat.

- MAMAExpDLL const char * mamaStat_getName (mamaStat stat)

  Get the name of the stat.

- MAMAExpDLL mama_status mamaStat_setLog (mamaStat stat, int log)

  Set whether or not this stat should be logged in the MAMA log.

- MAMAExpDLL int mamaStat_getLog (mamaStat stat)

  Returns whether or not the stat is currently being logged to the MAMA log.

- MAMAExpDLL mama_status mamaStat_setPublish (mamaStat stat, int publish)

  Set whether or not to publish this stat.

- MAMAExpDLL int mamaStat_getPublish (mamaStat stat)

  Returns whether or not the stat is currently being published.

- MAMAExpDLL mamaStatType mamaStatType_fromString (const char *statTypeString)

  Returns a mamaStatType from a string.

- MAMAExpDLL const char * mamaStatType_toString (mamaStatType statType)

  Returns a string representation of a mamaStatType.

5.30.1 Define Documentation

5.30.1.1 #define MAMA_STAT_NOT_LOCKABLE 0

5.30.1.2 #define MAMA_STAT_LOCKABLE 1

5.30.2 Enumeration Type Documentation

5.30.2.1 enum mamaStatType

Enumerator:

MAMA_STAT_TYPE_INITIALS

MAMA_STAT_TYPE_RECAPS

MAMA_STAT_TYPE_NUM_MESSAGES

MAMA_STAT_TYPE_FT_TAKEOVERS

MAMA_STAT_TYPE_QUEUE_SIZE

MAMA_STAT_TYPE_SUBSCRIPTIONS

MAMA_STAT_TYPE_TIMEOUTS

MAMA_STAT_TYPE_MSG_INDEX

MAMA_STAT_TYPE_NAK_PACKETS_SENT

MAMA_STAT_TYPE_NAKS_SENT

MAMA_STAT_TYPE_MSGS_LOST

MAMA_STAT_TYPE_NCFS_IGNORE

MAMA_STAT_TYPE_NCFS_SHED

MAMA_STAT_TYPE_NCFS_RX_DELAY

MAMA_STAT_TYPE_NCFS_UNKNOWN

MAMA_STAT_TYPE_DUPLICATE_MSGS

MAMA_STAT_TYPE_UNRECOVERABLE_WINDOW_ADVANCE

MAMA_STAT_TYPE_UNRECOVERABLE_TIMEOUT

MAMA_STAT_TYPE_LBM_MSGS_RECEIVED_NO_TOPIC

MAMA_STAT_TYPE_LBM_REQUESTS_RECEIVED

MAMA_STAT_TYPE_WOMBAT_MSGS

MAMA_STAT_TYPE_RV_MSGS

MAMA_STAT_TYPE_FAST_MSGS

MAMA_STAT_TYPE_UNKNOWN_MSGS

MAMA_STAT_TYPE_PUBLISHER_SEND

MAMA_STAT_TYPE_PUBLISHER_INBOX_SEND

MAMA_STAT_TYPE_PUBLISHER_REPLY_SEND

MAMA_STAT_TYPE_UNKNOWN

```
{
    MAMA_STAT_TYPE_INITIALS                      =  105,
    MAMA_STAT_TYPE_RECAPS                        =  106,
    MAMA_STAT_TYPE_NUM_MESSAGES                  =  107,
    MAMA_STAT_TYPE_FT_TAKEOVERS                  =  108,
    MAMA_STAT_TYPE_QUEUE_SIZE                    =  109,
    MAMA_STAT_TYPE_SUBSCRIPTIONS                 =  110,
    MAMA_STAT_TYPE_TIMEOUTS                      =  111,
    MAMA_STAT_TYPE_MSG_INDEX                     =  112,
    MAMA_STAT_TYPE_NAK_PACKETS_SENT              =  113,
    MAMA_STAT_TYPE_NAKS_SENT                     =  114,
    MAMA_STAT_TYPE_MSGS_LOST                     =  115,
    MAMA_STAT_TYPE_NCFS_IGNORE                   =  116,
    MAMA_STAT_TYPE_NCFS_SHED                     =  117,
    MAMA_STAT_TYPE_NCFS_RX_DELAY                 =  118,
    MAMA_STAT_TYPE_NCFS_UNKNOWN                  =  119,
    MAMA_STAT_TYPE_DUPLICATE_MSGS                =  120,
    MAMA_STAT_TYPE_UNRECOVERABLE_WINDOW_ADVANCE  =  121,
    MAMA_STAT_TYPE_UNRECOVERABLE_TIMEOUT         =  122,
    MAMA_STAT_TYPE_LBM_MSGS_RECEIVED_NO_TOPIC    =  123,
    MAMA_STAT_TYPE_LBM_REQUESTS_RECEIVED         =  124,
    MAMA_STAT_TYPE_WOMBAT_MSGS                   =  125,
    MAMA_STAT_TYPE_RV_MSGS                       =  126,
    MAMA_STAT_TYPE_FAST_MSGS                     =  127,
    MAMA_STAT_TYPE_UNKNOWN_MSGS                  =  128,
```

```
    MAMA_STAT_TYPE_PUBLISHER_SEND                =  129,
    MAMA_STAT_TYPE_PUBLISHER_INBOX_SEND          =  130,
    MAMA_STAT_TYPE_PUBLISHER_REPLY_SEND          =  131,
    MAMA_STAT_TYPE_UNKNOWN                       =  999
}mamaStatType;
```

5.30.3 Function Documentation

5.30.3.1 MAMAExpDLL mama_status mamaStat_create (mamaStat * *stat*, mamaStatsCollector *collector*, int *lockable*, const char * *name*, mama_fid_t *type*)

Create a mamaStats object.

Parameters:

stat The location of a mamaStats to store the result

5.30.3.2 MAMAExpDLL mama_status mamaStat_destroy (mamaStat *stat*)

Destroy a mamaStats object.

Parameters:

stat The stats object to destroy

5.30.3.3 MAMAExpDLL mama_status mamaStat_increment (mamaStat *stat*)

Increment the stats object counter.

Parameters:

stat The stats object to increment

5.30.3.4 MAMAExpDLL mama_status mamaStat_decrement (mamaStat *stat*)

Decrement the stats object counter.

Parameters:

stat The stats object to decrement

5.30.3.5 MAMAExpDLL mama_status mamaStat_reset (mamaStat *stat*)

Reset the stats object counter.

Parameters:

stat The stat object to reset

5.30.3.6 MAMAExpDLL mama_status mamaStat_add (mamaStat *stat*, int *value*)

Adds the stats object counter with the value.

Parameters:

stat The stats object to added

value The value to be added to be stat

5.30.3.7 MAMAExpDLL mama_status mamaStat_subtract (mamaStat *stat*, int *value*)

Subtract the stats object counter with the value.

Parameters:

stat The stats object to subtracted

value the value to be subtracted from stat

5.30.3.8 MAMAExpDLL mama_status mamaStat_setIntervalValue (mamaStat *stat*, int *value*)

set the Interval value

Parameters:

stat The stats object to be changed

value the value to be used

5.30.3.9 MAMAExpDLL mama_fid_t mamaStat_getFid (mamaStat *stat*)

Get the FID used when publishing the stat via the stats logger.

Parameters:

stat The stat object from which to get the FID

5.30.3.10 MAMAExpDLL int mamaStat_getIntervalValue (mamaStat *stat*)

Get the value of the stat for the current interval.

Parameters:

stat The stat object from which to get the invterval value

5.30.3.11 MAMAExpDLL int mamaStat_getMaxValue (mamaStat *stat*)

Get the maximum value of the stat.

Parameters:

stat The stat object from which to get the maximum value

5.30.3.12 MAMAExpDLL int mamaStat_getTotalValue (mamaStat *stat*)

Get the total value of the stat.

Parameters:

stat The stat object from which to get the total value

5.30.3.13 MAMAExpDLL void mamaStat_getStats (mamaStat *stat*, mama_i32_t * *intervalValue*, mama_u32_t * *maxValue*, mama_u32_t * *totalValue*)

Get the interval, maximum, and total values for the stat.

Parameters:

stat The stat object from which to get the values

intervalValue Address to which the interval value will be written

maxValue Address to which the maximum value will be written

totalValue Address to which the total value will be written

5.30.3.14 MAMAExpDLL const char* mamaStat_getName (mamaStat *stat*)

Get the name of the stat.

Parameters:

The stat object from which to get the name

5.30.3.15 MAMAExpDLL mama_status mamaStat_setLog (mamaStat *stat*, int *log*)

Set whether or not this stat should be logged in the MAMA log.

Parameters:

stat The stat object to set logging for

log Whether or not to log the stat

5.30.3.16 MAMAExpDLL int mamaStat_getLog (mamaStat *stat*)

Returns whether or not the stat is currently being logged to the MAMA log.

Parameters:

The stat object to get logging for

5.30.3.17 MAMAExpDLL mama_status mamaStat_setPublish (mamaStat *stat*, int *publish*)

Set whether or not to publish this stat.

stat The stat object to set publishing for publish Whether or not to publish the stat

5.30.3.18 MAMAExpDLL int mamaStat_getPublish (mamaStat *stat*)

Returns whether or not the stat is currently being published.

Parameters:

stat The stat object to check whether or not it is being published

5.30.3.19 MAMAExpDLL mamaStatType mamaStatType_fromString (const char * *statTypeString*)

Returns a mamaStatType from a string.

Parameters:

statTypeString The stat type as a string

5.30.3.20 MAMAExpDLL const char* mamaStatType_toString (mamaStatType *statType*)

Returns a string representation of a mamaStatType.

Parameters:

statType The statType to return as a string

5.31 statfields.h File Reference

```
#include "mama/mama.h"
```

Variables

- MAMAExpDLL const MamaReservedField MamaStatTime
- MAMAExpDLL const MamaReservedField MamaStatName
- MAMAExpDLL const MamaReservedField MamaStatType
- MAMAExpDLL const MamaReservedField MamaStatMiddleware
- MAMAExpDLL const MamaReservedField MamaStatInitials
- MAMAExpDLL const MamaReservedField MamaStatRecaps
- MAMAExpDLL const MamaReservedField MamaStatNumMessages
- MAMAExpDLL const MamaReservedField MamaStatFtTakeovers
- MAMAExpDLL const MamaReservedField MamaStatQueueSize
- MAMAExpDLL const MamaReservedField MamaStatNumSubscriptions
- MAMAExpDLL const MamaReservedField MamaStatTimeouts
- MAMAExpDLL const MamaReservedField MamaStatMsgIndex
- MAMAExpDLL const MamaReservedField MamaStatNakPacketsSent
- MAMAExpDLL const MamaReservedField MamaStatNaksSent
- MAMAExpDLL const MamaReservedField MamaStatMsgsLost
- MAMAExpDLL const MamaReservedField MamaStatNcfsIgnore
- MAMAExpDLL const MamaReservedField MamaStatNcfsShed
- MAMAExpDLL const MamaReservedField MamaStatNcfsRxDelay
- MAMAExpDLL const MamaReservedField MamaStatNcfsUnknown
- MAMAExpDLL const MamaReservedField MamaStatDuplicateDataMsgs
- MAMAExpDLL const MamaReservedField MamaStatUnrecoverableWindowAdvance
- MAMAExpDLL const MamaReservedField MamaStatUnrecoverableTimeout
- MAMAExpDLL const MamaReservedField MamaStatLbmMsgsReceivedNoTopic
- MAMAExpDLL const MamaReservedField MamaStatLbmRequestsReceived
- MAMAExpDLL const MamaReservedField MamaStatWombatMsgs
- MAMAExpDLL const MamaReservedField MamaStatRvMsgs
- MAMAExpDLL const MamaReservedField MamaStatFastMsgs
- MAMAExpDLL const MamaReservedField MamaStatUnknownMsgs
- MAMAExpDLL const MamaReservedField MamaStatPublisherSend
- MAMAExpDLL const MamaReservedField MamaStatPublisherInboxSend
- MAMAExpDLL const MamaReservedField MamaStatPublisherReplySend

5.31.1 Variable Documentation

5.31.1.1 MAMAExpDLL const MamaReservedField MamaStatTime

5.31.1.2 MAMAExpDLL const MamaReservedField MamaStatName

5.31.1.3 MAMAExpDLL const MamaReservedField MamaStatType

5.31.1.4 MAMAExpDLL const MamaReservedField MamaStatMiddleware

5.31.1.5 MAMAExpDLL const MamaReservedField MamaStatInitials

5.31.1.6 MAMAExpDLL const MamaReservedField MamaStatRecaps

5.31.1.7 MAMAExpDLL const MamaReservedField MamaStatNumMessages

5.31.1.8 MAMAExpDLL const MamaReservedField MamaStatFtTakeovers

5.31.1.9 MAMAExpDLL const MamaReservedField MamaStatQueueSize

5.31.1.10 MAMAExpDLL const MamaReservedField MamaStatNumSubscriptions

5.31.1.11 MAMAExpDLL const MamaReservedField MamaStatTimeouts

5.31.1.12 MAMAExpDLL const MamaReservedField MamaStatMsgIndex

5.31.1.13 MAMAExpDLL const MamaReservedField MamaStatNakPacketsSent

5.31.1.14 MAMAExpDLL const MamaReservedField MamaStatNaksSent

5.31.1.15 MAMAExpDLL const MamaReservedField MamaStatMsgsLost

5.31.1.16 MAMAExpDLL const MamaReservedField MamaStatNcfsIgnore

5.31.1.17 MAMAExpDLL const MamaReservedField MamaStatNcfsShed

5.31.1.18 MAMAExpDLL const MamaReservedField MamaStatNcfsRxDelay

5.31.1.19 MAMAExpDLL const MamaReservedField MamaStatNcfsUnknown

5.31.1.20 MAMAExpDLL const MamaReservedField MamaStatDuplicateDataMsgs

5.31.1.21 MAMAExpDLL const MamaReservedField MamaStatUnrecoverableWindowAdvance

5.31.1.22 MAMAExpDLL const MamaReservedField MamaStatUnrecoverableTimeout

5.31.1.23 MAMAExpDLL const MamaReservedField MamaStatLbmMsgsReceivedNoTopic

5.31.1.24 MAMAExpDLL const MamaReservedField

5.32 statscollector.h File Reference

```
#include "mama/status.h"
```

Enumerations

- enum mamaStatsCollectorType { MAMA_STATS_COLLECTOR_TYPE_QUEUE = 0, MAMA_STATS_COLLECTOR_TYPE_TRANSPORT = 1, MAMA_STATS_COLLECTOR_TYPE_USER = 2, MAMA_STATS_COLLECTOR_TYPE_GLOBAL = 3 }

Functions

- MAMAExpDLL mama_status mamaStatsCollector_create (mamaStatsCollector *statsCollector, mamaStatsCollectorType type, const char *name, const char *middleware)

  Create a mamaStatsCollector object.

- MAMAExpDLL mama_status mamaStatsCollector_destroy (mamaStatsCollector statsCollector)

  Destroy a mamaStatsCollector object.

- MAMAExpDLL mama_status mamaStatsCollector_addStat (mamaStatsCollector statsCollector, mamaStat stat)

  Register a stats object with the collector.

- MAMAExpDLL mama_status mamaStatsCollector_incrementStat (mamaStatsCollector statsCollector, mama_fid_t identifier)

  Increment the current interval value of the stat represented by FID identifier.

- MAMAExpDLL mama_status mamaStatsCollector_setName (mamaStatsCollector statsCollector, const char *name)

  Set the name of the stats collector object.

- MAMAExpDLL mama_status mamaStatsCollector_setPublish (mamaStatsCollector statsCollector, int publish)

  Set whether or not stats for this stats collector object should be published in stats messages.

- MAMAExpDLL int mamaStatsCollector_getPublish (mamaStatsCollector statsCollector)

  Get whether or not stats are being published for this stats collector.

- MAMAExpDLL mama_status mamaStatsCollector_setLog (mamaStatsCollector statsCollector, int log)

  Set whether or not stats for this stats collector object should be logged to the MAMA log.

- MAMAExpDLL int mamaStatsCollector_getLog (mamaStatsCollector statsCollector)

  Get whether or not stats are being logged for this stats collector.

- MAMAExpDLL const char * mamaStatsCollectorType_stringForType (mamaStatsCollectorType type)

  Return a string representation of a mamaStatsCollectorType.

5.32.1 Enumeration Type Documentation

5.32.1.1 enum mamaStatsCollectorType

Enumerator:

MAMA_STATS_COLLECTOR_TYPE_QUEUE

MAMA_STATS_COLLECTOR_TYPE_TRANSPORT

MAMA_STATS_COLLECTOR_TYPE_USER

MAMA_STATS_COLLECTOR_TYPE_GLOBAL

```
{
    MAMA_STATS_COLLECTOR_TYPE_QUEUE     = 0,
    MAMA_STATS_COLLECTOR_TYPE_TRANSPORT = 1,
    MAMA_STATS_COLLECTOR_TYPE_USER      = 2,
    MAMA_STATS_COLLECTOR_TYPE_GLOBAL    = 3
} mamaStatsCollectorType;
```

5.32.2 Function Documentation

5.32.2.1 MAMAExpDLL mama_status mamaStatsCollector_create (mamaStatsCollector * *statsCollector*, mamaStatsCollectorType *type*, const char * *name*, const char * *middleware*)

Create a mamaStatsCollector object.

Parameters:

statsCollector The stats collector object to create

5.32.2.2 MAMAExpDLL mama_status mamaStatsCollector_destroy (mamaStatsCollector *statsCollector*)

Destroy a mamaStatsCollector object.

Parameters:

statsCollector The stats collector object to destroy

5.32.2.3 MAMAExpDLL mama_status mamaStatsCollector_addStat (mamaStatsCollector *statsCollector*, mamaStat *stat*)

Register a stats object with the collector.

Parameters:

statscollector The stats collector object to register with

stat The stat object to register

5.32.2.4 MAMAExpDLL mama_status mamaStatsCollector_incrementStat (mamaStatsCollector *statsCollector*, mama_fid_t *identifier*)

Increment the current interval value of the stat represented by FID identifier.

Parameters:

statscollector The stats collector object for which to increment the stat

identifier FID of the stat to increment

5.32.2.5 MAMAExpDLL mama_status mamaStatsCollector_setName (mamaStatsCollector *statsCollector*, const char * *name*)

Set the name of the stats collector object.

By default, this will be the same as the associated queue, transport, or MAMA Application.

Parameters:

statscollector The stats collector object for which to set the name

name The name to set for the stats collector

5.32.2.6 MAMAExpDLL mama_status mamaStatsCollector_setPublish (mamaStatsCollector *statsCollector*, int *publish*)

Set whether or not stats for this stats collector object should be published in stats messages.

Parameters:

statscollector The stats collector object to set publishing for

publish Whether or not to publish stats for this stats collector

5.32.2.7 MAMAExpDLL int mamaStatsCollector_getPublish (mamaStatsCollector *statsCollector*)

Get whether or not stats are being published for this stats collector.

Parameters:

statscollector The stats collector object to get publishing for

5.32.2.8 MAMAExpDLL mama_status mamaStatsCollector_setLog (mamaStatsCollector *statsCollector*, int *log*)

Set whether or not stats for this stats collector object should be logged to the MAMA log.

Parameters:

statscollector The stats collector object to set logging for

log Whether or not to log stats for this stats collector

5.32.2.9 MAMAExpDLL int mamaStatsCollector_getLog (mamaStatsCollector *statsCollector*)

Get whether or not stats are being logged for this stats collector.

Parameters:

statscollector The stats collector object to get logging for.

5.32.2.10 MAMAExpDLL const char* mamaStatsCollectorType_stringForType (mamaStatsCollectorType *type*)

Return a string representation of a mamaStatsCollectorType.

Parameters:

type The stats collector type to return as a string

5.33 statslogger.h File Reference

```
#include <mama/status.h>
#include <mama/types.h>
```

Defines

- #define SL_TOPIC_USAGE_LOG "SL_TOPIC_USAGE_LOG"
- #define STATS_TOPIC "STATS_TOPIC"

Typedefs

- typedef enum mamaSlEvent_ mamaSlEvent

  The event types handled by the stats logger.

- typedef enum mamaSlSubFailReason_ mamaSlSubFailReason

  The reasons for a subscription to fail.

Enumerations

- enum mamaSlEvent_ {

  MAMA_SL_EVENT_LOGIN = 0, MAMA_SL_EVENT_LOGOUT = 1, MAMA_SL_EVENT_SUB_CREATE = 2, MAMA_SL_EVENT_SUB_FAIL = 3,

  MAMA_SL_EVENT_SUB_DESTROY = 4, MAMA_SL_EVENT_SUB_-ENT_PASS = 5 }

  The event types handled by the stats logger.

- enum mamaSlSubFailReason_ { MAMA_SL_SUB_FAIL_REGEX = 0, MAMA_SL_SUB_FAIL_CODE = 1, MAMA_SL_SUB_FAIL_TIMEOUT = 2, MAMA_SL_SUB_FAIL_NOT_FOUND = 3 }

  The reasons for a subscription to fail.

Functions

- MAMAExpDLL const char * mamaStatsLogger_eventToString (mamaSlEvent event)

  Convert a mamaSlEvent value to a string.

- MAMAExpDLL const char * mamaStatsLogger_subFailReasonToString (mamaSlSubFailReason subFailReason)

  Convert a mamaSlSubFailReason value to a string.

- MAMAExpDLL mama_status mamaStatsLogger_allocate (mamaStatsLogger *logger)

  Allocate memory for a new stats logger.

- MAMAExpDLL mama_status mamaStatsLogger_create (mamaStatsLogger logger, mamaQueue queue, mamaTransport tport)

  Create and activate mamaStatsLogger object.

- MAMAExpDLL mama_status mamaStatsLogger_createForStats (mamaStatsLogger logger, mamaQueue queue, mamaTransport tport, const char *topic)
- MAMAExpDLL mama_status mamaStatsLogger_destroy (mamaStatsLogger logger)

  Destroy a mamaStatsLogger object, freeing any memory.

- MAMAExpDLL mama_status mamaStatsLogger_setReportInterval (mamaStatsLogger logger, mama_f64_t interval)

  Set the reporting interval.

- MAMAExpDLL mama_status mamaStatsLogger_setReportSize (mamaStatsLogger logger, mama_size_t numEvents)

  Set the maximum number of events to cache before reporting.

- MAMAExpDLL mama_status mamaStatsLogger_setUserName (mamaStatsLogger logger, const char *userName)

  Set the userName.

- MAMAExpDLL mama_status mamaStatsLogger_getUserName (const mamaStatsLogger logger, const char **userName)

  Get the userName.

- MAMAExpDLL mama_status mamaStatsLogger_setHostName (mamaStatsLogger logger, const char *hostName)

  Set the hostName.

- MAMAExpDLL mama_status mamaStatsLogger_getHostName (const mamaStatsLogger logger, const char **hostName)

  Get the hostName.

- MAMAExpDLL mama_status mamaStatsLogger_setApplicationName (mamaStatsLogger logger, const char *appName)

  Set the applicationName.

- MAMAExpDLL mama_status mamaStatsLogger_getApplicationName (const mamaStatsLogger logger, const char **appName)

  Get the applicationName.

- MAMAExpDLL mama_status mamaStatsLogger_setApplicationClass (mamaStatsLogger logger, const char *applClass)

  Set the applicationClass.

- MAMAExpDLL mama_status mamaStatsLogger_getApplicationClass (const mamaStatsLogger logger, const char **appClass)

  Get the applicationClass.

- MAMAExpDLL mama_status mamaStatsLogger_setIpAddress (mamaStatsLogger logger, const char *ipAddress)

  Set the ipAddress.

- MAMAExpDLL mama_status mamaStatsLogger_getIpAddress (const mamaStatsLogger logger, const char **ipAddress)

  Get the ipAddress.

- MAMAExpDLL mama_status mamaStatsLogger_logLogin (mamaStatsLogger logger)

  Log a login event.

- MAMAExpDLL mama_status mamaStatsLogger_logLogout (mamaStatsLogger logger)

  Log a logout event.

- MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionEntitlePass (mamaStatsLogger logger, mamaSubscription subscription, int32_t entitleCode)

  Log a subscription entitlement check pass event.

- MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionFail (mamaStatsLogger logger, mamaSubscription subscription, mamaSlSubFailReason reason)

  Log a subscription fail event.

- MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionDestroy (mamaStatsLogger logger, mamaSubscription subscription)

Log a subscription destroy event.

- MAMAExpDLL mama_status mamaStatsLogger_addStatMsg (mamaStatsLogger *logger, mamaMsg msg)
- MAMAExpDLL mama_status mamaStatsLogger_sendReport (mamaStatsLogger *logger)
- MAMAExpDLL mama_status mamaStatsLogger_getLogMsgStats (mamaStatsLogger logger, mama_bool_t *logMsgStats)
- MAMAExpDLL mama_status mamaStatsLogger_setLogMsgStats (mamaStatsLogger logger, mama_bool_t logMsgStats)
- MAMAExpDLL mama_status mamaStatsLogger_incMsgCount (mamaStatsLogger statsLogger)

5.33.1 Define Documentation

5.33.1.1 #define SL_TOPIC_USAGE_LOG "SL_TOPIC_USAGE_LOG"

5.33.1.2 #define STATS_TOPIC "STATS_TOPIC"

5.33.2 Typedef Documentation

5.33.2.1 typedef enum mamaSlEvent_ mamaSlEvent

The event types handled by the stats logger.

5.33.2.2 typedef enum mamaSlSubFailReason_ mamaSlSubFailReason

The reasons for a subscription to fail.

5.33.3 Enumeration Type Documentation

5.33.3.1 enum mamaSlEvent_

The event types handled by the stats logger.

Enumerator:

MAMA_SL_EVENT_LOGIN

MAMA_SL_EVENT_LOGOUT

MAMA_SL_EVENT_SUB_CREATE

MAMA_SL_EVENT_SUB_FAIL

MAMA_SL_EVENT_SUB_DESTROY

MAMA_SL_EVENT_SUB_ENT_PASS

```
{
    MAMA_SL_EVENT_LOGIN         = 0,
    MAMA_SL_EVENT_LOGOUT        = 1,
    MAMA_SL_EVENT_SUB_CREATE    = 2,
    MAMA_SL_EVENT_SUB_FAIL      = 3,
    MAMA_SL_EVENT_SUB_DESTROY   = 4,
    MAMA_SL_EVENT_SUB_ENT_PASS  = 5
} mamaSlEvent;
```

5.33.3.2 enum mamaSlSubFailReason_

The reasons for a subscription to fail.

Enumerator:

MAMA_SL_SUB_FAIL_REGEX

MAMA_SL_SUB_FAIL_CODE

MAMA_SL_SUB_FAIL_TIMEOUT

MAMA_SL_SUB_FAIL_NOT_FOUND

```
{
    MAMA_SL_SUB_FAIL_REGEX      = 0,
    MAMA_SL_SUB_FAIL_CODE       = 1,
    MAMA_SL_SUB_FAIL_TIMEOUT    = 2,
    MAMA_SL_SUB_FAIL_NOT_FOUND  = 3
} mamaSlSubFailReason;
```

5.33.4 Function Documentation

5.33.4.1 MAMAExpDLL const char* mamaStatsLogger_eventToString (mamaSlEvent *event*)

Convert a mamaSlEvent value to a string.

Do no attempt to free the string result.

Parameters:

event The mamaSlEvent to convert.

5.33.4.2 MAMAExpDLL const char* mamaStatsLogger_-subFailReasonToString (mamaSlSubFailReason *subFailReason*)

Convert a mamaSlSubFailReason value to a string.

Do no attempt to free the string result.

Parameters:

subFailReason The mamaSlSubFailReason to convert.

5.33.4.3 MAMAExpDLL mama_status mamaStatsLogger_allocate (mamaStatsLogger * *logger*)

Allocate memory for a new stats logger.

The logger is not actually created until a call the create() functions is made. Memory must be freed using the `mamaStatsLogger_Deallocate()` function.

Parameters:

logger Where the address of the new logger will be written

5.33.4.4 MAMAExpDLL mama_status mamaStatsLogger_create (mamaStatsLogger *logger*, mamaQueue *queue*, mamaTransport *tport*)

Create and activate mamaStatsLogger object.

Parameters:

logger The location of a mamaStatsLogger

queue The queue for the Stats Logger to use for publishing

tport the transport

5.33.4.5 MAMAExpDLL mama_status mamaStatsLogger_createForStats (mamaStatsLogger *logger*, mamaQueue *queue*, mamaTransport *tport*, const char * *topic*)

5.33.4.6 MAMAExpDLL mama_status mamaStatsLogger_destroy (mamaStatsLogger *logger*)

Destroy a mamaStatsLogger object, freeing any memory.

Parameters:

logger The location of a mamaStatsLogger

5.33.4.7 MAMAExpDLL mama_status mamaStatsLogger_setReportInterval (mamaStatsLogger *logger*, mama_f64_t *interval*)

Set the reporting interval.

Parameters:

logger The location of a mamaStatsLogger

interval The report interval in seconds

5.33.4.8 MAMAExpDLL mama_status mamaStatsLogger_setReportSize (mamaStatsLogger *logger*, mama_size_t *numEvents*)

Set the maximum number of events to cache before reporting.

Parameters:

logger The location of a mamaStatsLogger

numEvents The number of events to report after

5.33.4.9 MAMAExpDLL mama_status mamaStatsLogger_setUserName (mamaStatsLogger *logger*, const char * *userName*)

Set the userName.

Parameters:

logger The logger object to update.

userName The user name for the logger

5.33.4.10 MAMAExpDLL mama_status mamaStatsLogger_getUserName (const mamaStatsLogger *logger*, const char ** *userName*)

Get the userName.

Parameters:

logger The logger object to check.

userName Location of the result for the description of the logger.

5.33.4.11 MAMAExpDLL mama_status mamaStatsLogger_setHostName (mamaStatsLogger *logger*, const char * *hostName*)

Set the hostName.

Parameters:

logger The logger object to update.

hostName The host name for the logger

5.33.4.12 MAMAExpDLL mama_status mamaStatsLogger_getHostName (const mamaStatsLogger *logger*, const char ** *hostName*)

Get the hostName.

Parameters:

logger The logger object to check.

hostName Location of the result for the description of the logger.

5.33.4.13 MAMAExpDLL mama_status mamaStatsLogger_setApplicationName (mamaStatsLogger *logger*, const char * *appName*)

Set the applicationName.

Parameters:

logger The logger object to update.

appName The applicationName for the logger

5.33.4.14 MAMAExpDLL mama_status mamaStatsLogger_getApplicationName (const mamaStatsLogger *logger*, const char ** *appName*)

Get the applicationName.

Parameters:

logger The logger object to check.

appName Location of the result for the applicationName

5.33.4.15 MAMAExpDLL mama_status mamaStatsLogger_setApplicationClass (mamaStatsLogger *logger*, const char * *applClass*)

Set the applicationClass.

Parameters:

logger The logger object to update.

applClass The applicationName for the logger

5.33.4.16 MAMAExpDLL mama_status mamaStatsLogger_getApplicationClass (const mamaStatsLogger *logger*, const char ** *appClass*)

Get the applicationClass.

Parameters:

logger The logger object to check.

appClass Location of the result for the applicationClass

5.33.4.17 MAMAExpDLL mama_status mamaStatsLogger_setIpAddress (mamaStatsLogger *logger*, const char * *ipAddress*)

Set the ipAddress.

Parameters:

logger The logger object to update.

ipAddress The ipAddress for the logger

5.33.4.18 MAMAExpDLL mama_status mamaStatsLogger_getIpAddress (const mamaStatsLogger *logger*, const char ** *ipAddress*)

Get the ipAddress.

Parameters:

logger The logger object to check.

ipAddress Location of the result for the ipAddress of the logger.

5.33.4.19 MAMAExpDLL mama_status mamaStatsLogger_logLogin (mamaStatsLogger *logger*)

Log a login event.

Parameters:

logger The logger object to use.

5.33.4.20 MAMAExpDLL mama_status mamaStatsLogger_logLogout (mamaStatsLogger *logger*)

Log a logout event.

Parameters:

logger The logger object to use.

5.33.4.21 MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionEntitlePass (mamaStatsLogger *logger*, mamaSubscription *subscription*, int32_t *entitleCode*)

Log a subscription entitlement check pass event.

Parameters:

logger The logger object to use.

subscription the subscription

entitleCode the entitlement code

5.33.4.22 MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionFail (mamaStatsLogger *logger*, mamaSubscription *subscription*, mamaSlSubFailReason *reason*)

Log a subscription fail event.

Parameters:

logger The logger object to use.

subscription the subscription

reason the reason

5.33.4.23 MAMAExpDLL mama_status mamaStatsLogger_logSubscriptionDestroy (mamaStatsLogger *logger*, mamaSubscription *subscription*)

Log a subscription destroy event.

Parameters:

logger The logger object to use.

subscription the subscription

5.33.4.24 MAMAExpDLL mama_status mamaStatsLogger_addStatMsg (mamaStatsLogger * *logger*, mamaMsg *msg*)

5.33.4.25 MAMAExpDLL mama_status mamaStatsLogger_sendReport (mamaStatsLogger * *logger*)

5.33.4.26 MAMAExpDLL mama_status mamaStatsLogger_getLogMsgStats (mamaStatsLogger *logger*, mama_bool_t * *logMsgStats*)

5.33.4.27 MAMAExpDLL mama_status mamaStatsLogger_setLogMsgStats (mamaStatsLogger *logger*, mama_bool_t *logMsgStats*)

5.33.4.28 MAMAExpDLL mama_status mamaStatsLogger_incMsgCount (mamaStatsLogger *statsLogger*)

5.34 statsloggerfields.h File Reference

```
#include "mama/mama.h"
```

Variables

- MAMAExpDLL const MamaReservedField MamaSlIntervalStartTime
- MAMAExpDLL const MamaReservedField MamaSlIntervalEndTime
- MAMAExpDLL const MamaReservedField MamaSlEvents
- MAMAExpDLL const MamaReservedField MamaSlEventTime
- MAMAExpDLL const MamaReservedField MamaSlEventType
- MAMAExpDLL const MamaReservedField MamaSlEventReason
- MAMAExpDLL const MamaReservedField MamaSlTotalMsgCount
- MAMAExpDLL const MamaReservedField MamaSlIntervalMsgCount
- MAMAExpDLL const MamaReservedField MamaStatEvents

5.34.1 Variable Documentation

5.34.1.1 MAMAExpDLL const MamaReservedField MamaSlIntervalStartTime

5.34.1.2 MAMAExpDLL const MamaReservedField MamaSlIntervalEndTime

5.34.1.3 MAMAExpDLL const MamaReservedField MamaSlEvents

5.34.1.4 MAMAExpDLL const MamaReservedField MamaSlEventTime

5.34.1.5 MAMAExpDLL const MamaReservedField MamaSlEventType

5.34.1.6 MAMAExpDLL const MamaReservedField MamaSlEventReason

5.34.1.7 MAMAExpDLL const MamaReservedField MamaSlTotalMsgCount

5.34.1.8 MAMAExpDLL const MamaReservedField MamaSlIntervalMsgCount

5.34.1.9 MAMAExpDLL const MamaReservedField MamaStatEvents

5.35 status.h File Reference

Defines

- #define MAMA_STATUS_BASE 9000
- #define MAMA_STATUS_MAX 1000
- #define NULLARG_STATUS_CHECK(x)
- #define NULLARG_STATUS_CHECK_STR(x)
- #define NOMEM_STATUS_CHECK(x)

Enumerations

- enum mama_status {

  MAMA_STATUS_OK = 0, MAMA_STATUS_NOMEM = 1, MAMA_STATUS_PLATFORM = 2, MAMA_STATUS_SYSTEM_ERROR = 3,

  MAMA_STATUS_INVALID_ARG = 4, MAMA_STATUS_NULL_ARG = 5, MAMA_STATUS_NOT_FOUND = 6, MAMA_STATUS_TIMER_FAILURE = 7,

  MAMA_STATUS_IP_NOT_FOUND = 8, MAMA_STATUS_TIMEOUT = 9, MAMA_STATUS_NOT_ENTITLED = 10, MAMA_STATUS_PROPERTY_TOO_LONG = 11,

  MAMA_STATUS_MD_NOT_OPENED = 12, MAMA_STATUS_PUB_SUB_NOT_OPENED = 13, MAMA_STATUS_ENTITLEMENTS_NOT_ENABLED = 14, MAMA_STATUS_BAD_TRANSPORT_TYPE = 15,

  MAMA_STATUS_UNSUPPORTED_IO_TYPE = 16, MAMA_STATUS_TOO_MANY_DISPATCHERS = 17, MAMA_STATUS_NOT_IMPLEMENTED = 18, MAMA_STATUS_WRONG_FIELD_TYPE = 19,

  MAMA_STATUS_BAD_SYMBOL = 20, MAMA_STATUS_IO_ERROR = 21, MAMA_STATUS_NOT_INSTALLED = 22, MAMA_STATUS_CONFLATE_ERROR = 23,

  MAMA_STATUS_QUEUE_FULL = 24, MAMA_STATUS_QUEUE_END = 25, MAMA_STATUS_NO_BRIDGE_IMPL = 26, MAMA_STATUS_INVALID_QUEUE = 27,

  MAMA_STATUS_NOT_MODIFIABLE = 28, MAMA_STATUS_DELETE = 29, MAMA_STATUS_NOT_PERMISSIONED = 4001, MAMA_STATUS_SUBSCRIPTION_INVALID_STATE = 5001,

  MAMA_STATUS_QUEUE_OPEN_OBJECTS = 5002, MAMA_STATUS_SUBSCRIPTION_INVALID_TYPE = 5003, MAMA_STATUS_SUBSCRIPTION_GAP = 5004, MAMA_STATUS_NOT_INITIALISED = 5005,

  MAMA_STATUS_NO_SUBSCRIBERS = 5006, MAMA_STATUS_EXPIRED = 5007, MAMA_STATUS_BANDWIDTH_EXCEEDED = 5008 }

Functions

- MAMAExpDLL const char * mamaStatus_stringForStatus (mama_status status)

5.35.1 Define Documentation

5.35.1.1 #define MAMA_STATUS_BASE 9000

5.35.1.2 #define MAMA_STATUS_MAX 1000

5.35.1.3 #define NULLARG_STATUS_CHECK(x)

Value:

```
do { \
        if (!(x)) return MAMA_STATUS_NULL_ARG; \
        } while(0);
```

5.35.1.4 #define NULLARG_STATUS_CHECK_STR(x)

Value:

```
do { \
        if (!(x) || (strlen((x))==0) ) return MAMA_STATUS_NULL_ARG; \
    } while(0);
```

5.35.1.5 #define NOMEM_STATUS_CHECK(x)

Value:

```
do { \
        if ((x==NULL))  \
        {    \
            mama_log (MAMA_LOG_LEVEL_SEVERE, "Could not allocate memory");   \
            return MAMA_STATUS_NOMEM;      \
         } \
        } while(0);
```

5.35.2 Enumeration Type Documentation

5.35.2.1 enum mama_status

Enumerator:

MAMA_STATUS_OK

MAMA_STATUS_NOMEM
MAMA_STATUS_PLATFORM
MAMA_STATUS_SYSTEM_ERROR
MAMA_STATUS_INVALID_ARG
MAMA_STATUS_NULL_ARG
MAMA_STATUS_NOT_FOUND
MAMA_STATUS_TIMER_FAILURE
MAMA_STATUS_IP_NOT_FOUND
MAMA_STATUS_TIMEOUT
MAMA_STATUS_NOT_ENTITLED
MAMA_STATUS_PROPERTY_TOO_LONG
MAMA_STATUS_MD_NOT_OPENED
MAMA_STATUS_PUB_SUB_NOT_OPENED
MAMA_STATUS_ENTITLEMENTS_NOT_ENABLED
MAMA_STATUS_BAD_TRANSPORT_TYPE
MAMA_STATUS_UNSUPPORTED_IO_TYPE
MAMA_STATUS_TOO_MANY_DISPATCHERS
MAMA_STATUS_NOT_IMPLEMENTED
MAMA_STATUS_WRONG_FIELD_TYPE
MAMA_STATUS_BAD_SYMBOL
MAMA_STATUS_IO_ERROR
MAMA_STATUS_NOT_INSTALLED
MAMA_STATUS_CONFLATE_ERROR
MAMA_STATUS_QUEUE_FULL
MAMA_STATUS_QUEUE_END
MAMA_STATUS_NO_BRIDGE_IMPL
MAMA_STATUS_INVALID_QUEUE
MAMA_STATUS_NOT_MODIFIABLE
MAMA_STATUS_DELETE
MAMA_STATUS_NOT_PERMISSIONED
MAMA_STATUS_SUBSCRIPTION_INVALID_STATE
MAMA_STATUS_QUEUE_OPEN_OBJECTS
MAMA_STATUS_SUBSCRIPTION_INVALID_TYPE
MAMA_STATUS_SUBSCRIPTION_GAP
MAMA_STATUS_NOT_INITIALISED

MAMA_STATUS_NO_SUBSCRIBERS

MAMA_STATUS_EXPIRED

MAMA_STATUS_BANDWIDTH_EXCEEDED

```
{
    /* Status OK */
    MAMA_STATUS_OK                          = 0,
    /* Out of memory */
    MAMA_STATUS_NOMEM                       = 1,
    /* Messaging platform specific error */
    MAMA_STATUS_PLATFORM                    = 2,
    /* General system error */
    MAMA_STATUS_SYSTEM_ERROR                = 3,
    /* Invalid argument */
    MAMA_STATUS_INVALID_ARG                 = 4,
    /* Null argument */
    MAMA_STATUS_NULL_ARG                    = 5,
    /* Not found */
    MAMA_STATUS_NOT_FOUND                   = 6,
    /* Timer failure */
    MAMA_STATUS_TIMER_FAILURE               = 7,
    /* IP address not found */
    MAMA_STATUS_IP_NOT_FOUND                = 8,
    /* Timeout e.g. when subscribing to a symbol */
    MAMA_STATUS_TIMEOUT                     = 9,
    /* Not entitled to the symbol being subscribed to */
    MAMA_STATUS_NOT_ENTITLED                = 10,
    /* Property too long */
    MAMA_STATUS_PROPERTY_TOO_LONG           = 11,
    /* MD Not opened */
    MAMA_STATUS_MD_NOT_OPENED               = 12,
    /* Publish/subscribe not opened */
    MAMA_STATUS_PUB_SUB_NOT_OPENED          = 13,
    /* Entitlements not enabled */
    MAMA_STATUS_ENTITLEMENTS_NOT_ENABLED    = 14,
    /* Bad transport type */
    MAMA_STATUS_BAD_TRANSPORT_TYPE          = 15,
    /* Using unsupported I/O type */
    MAMA_STATUS_UNSUPPORTED_IO_TYPE         = 16,
    /* Too many dispatchers */
    MAMA_STATUS_TOO_MANY_DISPATCHERS        = 17,
    /* Not implemented */
    MAMA_STATUS_NOT_IMPLEMENTED             = 18,
    /* Wrong field type */
    MAMA_STATUS_WRONG_FIELD_TYPE            = 19,
    /* Invalid symbol */
    MAMA_STATUS_BAD_SYMBOL                  = 20,
    /* I/O error */
    MAMA_STATUS_IO_ERROR                    = 21,
    /* Not installed */
    MAMA_STATUS_NOT_INSTALLED               = 22,
    /* Conflation error */
    MAMA_STATUS_CONFLATE_ERROR              = 23,
    /* Event dispatch queue full */
    MAMA_STATUS_QUEUE_FULL                  = 24,
    /* End of event queue reached */
```

```
    MAMA_STATUS_QUEUE_END                  = 25,
    /* No bridge */
    MAMA_STATUS_NO_BRIDGE_IMPL             = 26,
    /* Invalid queue */
    MAMA_STATUS_INVALID_QUEUE              = 27,
     /* Not modifiable  */
    MAMA_STATUS_NOT_MODIFIABLE             = 28,
     /* Message Type DELETE  */
    MAMA_STATUS_DELETE                     = 29,
        /* Not permissioned for the subject */
    MAMA_STATUS_NOT_PERMISSIONED                        = 4001,
    /* Subscription is in an invalid state. */
    MAMA_STATUS_SUBSCRIPTION_INVALID_STATE  = 5001,
    /* Queue has open objects. */
    MAMA_STATUS_QUEUE_OPEN_OBJECTS          = 5002,
    /* The function isn't supported for this type of subscription. */
    MAMA_STATUS_SUBSCRIPTION_INVALID_TYPE   = 5003,
    /* The underlying transport saw a gap. */
    MAMA_STATUS_SUBSCRIPTION_GAP            = 5004,

    /* A resource has not been initialised. */
    MAMA_STATUS_NOT_INITIALISED             = 5005,

    /* The transport line is reported to be down. */
    MAMA_STATUS_NO_SUBSCRIBERS              = 5006,

    /* The symbol has expired. */
    MAMA_STATUS_EXPIRED                     = 5007,

    /* The application's bandwidth limit has been exceeded. */
    MAMA_STATUS_BANDWIDTH_EXCEEDED          = 5008

#ifdef WITH_ENTITLEMENTS
    /* Out of memory */
    ,MAMA_ENTITLE_STATUS_NOMEM      = MAMA_STATUS_BASE + OEA_STATUS_NOMEM, /* 9001 */
    /* Invalid parameter */
    MAMA_ENTITLE_STATUS_BAD_PARAM  = MAMA_STATUS_BASE + OEA_STATUS_BAD_PARAM, /* 9002 */
    /* The XML returned from entitlement server was invalid */
    MAMA_ENTITLE_STATUS_BAD_DATA   = MAMA_STATUS_BASE + OEA_STATUS_BAD_DATA, /* 9003 */
    /* Invalid URL */
    MAMA_ENTITLE_STATUS_URL_ERROR  = MAMA_STATUS_BASE + OEA_STATUS_HTTP_URL_ERROR, /* 9004
    /* Unable to determine OS ID of account process is running under.*/
    MAMA_ENTITLE_STATUS_OS_LOGIN_ID_UNAVAILABLE = MAMA_STATUS_BASE + OEA_STATUS_OS_LOGIN_ID
    /* When an attempt is made to get entitlements after a successful attempt
     * has already been made */
    MAMA_ENTITLE_STATUS_ALREADY_ENTITLED        = MAMA_STATUS_BASE + OEA_STATUS_ALREADY_ENT
    /* A user has exceeded concurrent access limit */
    MAMA_ENTITLE_STATUS_CAC_LIMIT_EXCEEDED      = MAMA_STATUS_BASE + OEA_STATUS_CAC_LIMIT_E
    /* Failed to create OEP listener that processes inbound messages from
     * site server.  Required for concurrent access control and/or
     * dynamic entitlements update. */
    MAMA_ENTITLE_STATUS_OEP_LISTENER_CREATION_FAILURE = MAMA_STATUS_BASE + OEA_STATUS_OEP_L
    /* No such host */
    MAMA_ENTITLE_HTTP_ERRHOST      = MAMA_STATUS_BASE + OEA_HTTP_ERRHOST, /* 9010 */
    /* Can't create socket */
    MAMA_ENTITLE_HTTP_ERRSOCK      = MAMA_STATUS_BASE + OEA_HTTP_ERRSOCK, /* 9011 */
    /* Can't connect to host */
```

```
    MAMA_ENTITLE_HTTP_ERRCONN      = MAMA_STATUS_BASE + OEA_HTTP_ERRCONN, /* 9012 */
    /* Write error on socket while writing header */
    MAMA_ENTITLE_HTTP_ERRWRHD      = MAMA_STATUS_BASE + OEA_HTTP_ERRWRHD, /* 9013 */
    /* Write error on socket while writing data */
    MAMA_ENTITLE_HTTP_ERRWRDT      = MAMA_STATUS_BASE + OEA_HTTP_ERRWRDT, /* 9014 */
    /* Read error on socket while reading result */
    MAMA_ENTITLE_HTTP_ERRRDHD      = MAMA_STATUS_BASE + OEA_HTTP_ERRRDHD, /* 9015 */
    /* Invalid answer from data server */
    MAMA_ENTITLE_HTTP_ERRPAHD      = MAMA_STATUS_BASE + OEA_HTTP_ERRPAHD, /* 9016 */
    /* Null data pointer */
    MAMA_ENTITLE_HTTP_ERRNULL      = MAMA_STATUS_BASE + OEA_HTTP_ERRNULL, /* 9017 */
    /* No/Bad length in header */
    MAMA_ENTITLE_HTTP_ERRNOLG      = MAMA_STATUS_BASE + OEA_HTTP_ERRNOLG, /* 9018 */
    /* Can't allocate memory */
    MAMA_ENTITLE_HTTP_ERRMEM       = MAMA_STATUS_BASE + OEA_HTTP_ERRMEM, /* 9019 */
    /* Read error while reading data */
    MAMA_ENTITLE_HTTP_ERRRDDT      = MAMA_STATUS_BASE + OEA_HTTP_ERRRDDT, /* 9020 */
    /* Invalid url - must start with 'http:// */
    MAMA_ENTITLE_HTTP_ERRURLH      = MAMA_STATUS_BASE + OEA_HTTP_ERRURLH, /* 9021 */
    /* Invalid port in url */
    MAMA_ENTITLE_HTTP_ERRURLP      = MAMA_STATUS_BASE + OEA_HTTP_ERRURLP, /* 9022 */
    /* Invalid QUERY HTTP RESULT 400 */
    MAMA_ENTITLE_HTTP_BAD_QUERY    = MAMA_STATUS_BASE + OEA_HTTP_BAD_QUERY, /* 9023 */
    /* FORBIDDEN HTTP RESULT 403 */
    MAMA_ENTITLE_HTTP_FORBIDDEN    = MAMA_STATUS_BASE + OEA_HTTP_FORBIDDEN, /* 9024 */
    /* Request Timeout HTTP RESULT 403 */
    MAMA_ENTITLE_HTTP_TIMEOUT      = MAMA_STATUS_BASE + OEA_HTTP_TIMEOUT, /* 9025 */
    /* Server Error HTTP RESULT 500 */
    MAMA_ENTITLE_HTTP_SERVER_ERR   = MAMA_STATUS_BASE + OEA_HTTP_SERVER_ERR, /* 9026 */
    /* Not Implemented HTTP RESULT 501 */
    MAMA_ENTITLE_HTTP_NO_IMPL      = MAMA_STATUS_BASE + OEA_HTTP_NO_IMPL, /* 9027 */
    /* Overloaded HTTP RESULT 503 */
    MAMA_ENTITLE_HTTP_OVERLOAD     = MAMA_STATUS_BASE + OEA_HTTP_OVERLOAD, /* 9028 */
    /* No servers specified */
    MAMA_ENTITLE_NO_USER           = MAMA_ENTITLE_HTTP_OVERLOAD + 1, /* 9029 */
    MAMA_ENTITLE_NO_SERVERS_SPECIFIED =  MAMA_ENTITLE_NO_USER + 1, /* 9030 */
    MAMA_ENTITLE_SITE_NOT_FOUND    = MAMA_STATUS_BASE + OEA_STATUS_SITE_NOT_FOUND /* 9032 */
#endif
} mama_status;
```

5.35.3 Function Documentation

5.35.3.1 MAMAExpDLL const char* mamaStatus_stringForStatus (mama_status *status*)

5.36 subscmsgtype.h File Reference

Defines

- #define MAMA_SUBSCROOT "_MD"
- #define MAMA_SUBSCMANROOT "_MDSM"
- #define MAMA_DATADICTROOT "_MDDD"
- #define MAMA_WORLDVIEWROOT "_MDWV"
- #define MAMA_WORLDVIEWREQUEST "_MDWV.WORLDVIEW_-REQUEST"
- #define MAMA_WORLDVIEWUPDATE "_MDWV.WORLDVIEW_-UPDATE"
- #define MAMA_TIMEOUT (10.0)
- #define MAMA_RETRIES (2)
- #define MAMA_REFRESHINTERVALMIN (5.0)
- #define MAMA_REFRESHINTERVALRAND (55.0)

Enumerations

- enum mamaSubscMsgType {

  MAMA_SUBSC_SUBSCRIBE = 0, MAMA_SUBSC_SNAPSHOT = 1, MAMA_SUBSC_UNSUBSCRIBE = 2, MAMA_SUBSC_RESUBSCRIBE = 3,

  MAMA_SUBSC_REFRESH = 4, MAMA_SUBSC_DDICT_SNAPSHOT = 5, MAMA_SUBSC_WORLDVIEW_RQST = 6, MAMA_SUBSC_DQ_-SUBSCRIBER = 7,

  MAMA_SUBSC_DQ_PUBLISHER = 8, MAMA_SUBSC_DQ_NETWORK = 9, MAMA_SUBSC_DQ_UNKNOWN = 10, MAMA_SUBSC_FEATURE_-SET_RQST = 11,

  MAMA_SUBSC_DQ_GROUP_SUBSCRIBER = 12 }

Functions

- MAMAExpDLL const char * MamaSubscMsgTypeStr (mamaSubscMsgType msgType)

5.36.1 Define Documentation

5.36.1.1 #define MAMA_SUBSCROOT "_MD"

5.36.1.2 #define MAMA_SUBSCMANROOT "_MDSM"

5.36.1.3 #define MAMA_DATADICTROOT "_MDDD"

5.36.1.4 #define MAMA_WORLDVIEWROOT "_MDWV"

5.36.1.5 #define MAMA_WORLDVIEWREQUEST "_-MDWV.WORLDVIEW_REQUEST"

5.36.1.6 #define MAMA_WORLDVIEWUPDATE "_MDWV.WORLDVIEW_-UPDATE"

5.36.1.7 #define MAMA_TIMEOUT (10.0)

5.36.1.8 #define MAMA_RETRIES (2)

5.36.1.9 #define MAMA_REFRESHINTERVALMIN (5.0)

5.36.1.10 #define MAMA_REFRESHINTERVALRAND (55.0)

5.36.2 Enumeration Type Documentation

5.36.2.1 enum mamaSubscMsgType

Enumerator:

MAMA_SUBSC_SUBSCRIBE

MAMA_SUBSC_SNAPSHOT

MAMA_SUBSC_UNSUBSCRIBE

MAMA_SUBSC_RESUBSCRIBE

MAMA_SUBSC_REFRESH

MAMA_SUBSC_DDICT_SNAPSHOT

MAMA_SUBSC_WORLDVIEW_RQST

MAMA_SUBSC_DQ_SUBSCRIBER

MAMA_SUBSC_DQ_PUBLISHER

MAMA_SUBSC_DQ_NETWORK

MAMA_SUBSC_DQ_UNKNOWN

MAMA_SUBSC_FEATURE_SET_RQST

MAMA_SUBSC_DQ_GROUP_SUBSCRIBER

```
{
    MAMA_SUBSC_SUBSCRIBE            = 0,
    MAMA_SUBSC_SNAPSHOT             = 1,
    MAMA_SUBSC_UNSUBSCRIBE          = 2,
    MAMA_SUBSC_RESUBSCRIBE          = 3,
    MAMA_SUBSC_REFRESH              = 4,
    MAMA_SUBSC_DDICT_SNAPSHOT       = 5,
    MAMA_SUBSC_WORLDVIEW_RQST       = 6,
    MAMA_SUBSC_DQ_SUBSCRIBER        = 7,  /* Subscriber knows it lost data itself  */
    MAMA_SUBSC_DQ_PUBLISHER         = 8,  /* Subscriber knows publisher lost data  */
    MAMA_SUBSC_DQ_NETWORK           = 9,  /* Subscriber knows network lost data    */
    MAMA_SUBSC_DQ_UNKNOWN           = 10, /* Subscriber lost data but not know why */
    MAMA_SUBSC_FEATURE_SET_RQST     = 11,
    MAMA_SUBSC_DQ_GROUP_SUBSCRIBER  = 12  /* Subscriber knows it lost data itself  */
} mamaSubscMsgType;
```

5.36.3 Function Documentation

5.36.3.1 MAMAExpDLL const char* MamaSubscMsgTypeStr (mamaSubscMsgType *msgType*)

5.37 subscription.h File Reference

```
#include "mama/marketdata.h"
#include "mama/msg.h"
#include "mama/quality.h"
#include "mama/servicelevel.h"
#include "mama/status.h"
#include "mama/subscriptiontype.h"
#include "mama/transport.h"
#include "mama/types.h"
```

Data Structures

- struct mamaMsgCallbacks_

  A convenience structure for passing the callbacks to the subscription factory methods.

- struct mamaWildCardMsgCallbacks_

  A convenience structure for passing the callbacks to wild card subscription factory methods.

Typedefs

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionCreateCB (mamaSubscription subscription, void ∗closure)

  Function invoked when subscription creation is complete, and before any calls to `wombat_subscriptionOnMsgCB`.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionDestroyCB (mamaSubscription subscription, void ∗closure)

  Function invoked when a subscription has been completely destroyed, the client can have confidence that no further messages will be placed on the queue for this subscription.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionErrorCB (mamaSubscription subscription, mama_status status, void ∗platformError, const char ∗subject, void ∗closure)

  Invoked if an error occurs during prior to subscription creation or if the subscription receives a message for an unentitled subject.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionGapCB (mamaSubscription subscription, void ∗closure)

  Function invoked when a sequence number gap is detected.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionOnMsgCB (mamaSubscription subscription, mamaMsg msg, void ∗closure, void ∗itemClosure)

  Invoked when a message arrives.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionQualityCB (mamaSubscription subscription, mamaQuality quality, const char ∗symbol, short cause, const void ∗platformInfo, void ∗closure)

  Invoked to indicate a data quality event.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionRecapCB (mamaSubscription subscription, void ∗closure)

  Function invoked when a recap is requested upon detecting a sequence number gap.

- typedef void(MAMACALLTYPE ∗) wombat_subscriptionWildCardOnMsgCB (mamaSubscription subscription, mamaMsg msg, const char ∗topic, void ∗closure, void ∗itemClosure)

  Invoked when a message arrives for a wild card subscrption.

- typedef mamaMsgCallbacks_ mamaMsgCallbacks

  A convenience structure for passing the callbacks to the subscription factory methods.

- typedef mamaWildCardMsgCallbacks_ mamaWildCardMsgCallbacks

  A convenience structure for passing the callbacks to wild card subscription factory methods.

Enumerations

- enum preInitialScheme { PRE_INITIAL_SCHEME_ON_GAP, PRE_INITIAL_SCHEME_ON_INITIAL }

  MAMA can cache messages which arrive prior to an initial value.

- enum mamaSubscriptionState {

  MAMA_SUBSCRIPTION_UNKNOWN = 0, MAMA_SUBSCRIPTION_ALLOCATED = 1, MAMA_SUBSCRIPTION_SETUP = 2, MAMA_SUBSCRIPTION_ACTIVATING = 3,

  MAMA_SUBSCRIPTION_ACTIVATED = 4, MAMA_SUBSCRIPTION_DEACTIVATING = 5, MAMA_SUBSCRIPTION_DEACTIVATED = 6, MAMA_SUBSCRIPTION_DESTROYING = 7,

MAMA_SUBSCRIPTION_DESTROYED = 8, MAMA_SUBSCRIPTION_-DEALLOCATING = 9, MAMA_SUBSCRIPTION_DEALLOCATED = 10, MAMA_SUBSCRIPTION_REACTIVATING = 11 }

The enumeration describes the state of the subsubscription at any time.

- enum dqStartegyScheme { DQ_SCHEME_DELIVER_ALL, DQ_SCHEME_-INGORE_DUPS }
- enum dqftStrategyScheme { DQ_FT_DO_NOT_WAIT_FOR_RECAP, DQ_-FT_WAIT_FOR_RECAP }

Functions

- MAMAExpDLL mama_status mamaSubscription_activate (mamaSubscription subscription)

  Activate a subscription that has been set up by mamaSubscription_setup().

- MAMAExpDLL mama_status mamaSubscription_allocate (mamaSubscription *result)

  Allocate memory for a new subscription.

- MAMAExpDLL int mamaSubscription_checkDebugLevel (mamaSubscription subscription, MamaLogLevel level)

  Return whether the debug level for this subscription equals or exceeds some level.

- MAMAExpDLL mama_status mamaSubscription_create (mamaSubscription subscription, mamaQueue queue, const mamaMsgCallbacks *callbacks, mama-Source source, const char *symbol, void *closure)

  Create and activate subscription.

- MAMAExpDLL mama_status mamaSubscription_createBasic (mama-Subscription subscription, mamaTransport transport, mamaQueue queue, const mamaMsgCallbacks *callbacks, const char *symbol, void *closure)

  Create a basic subscription without marketdata semantics.

- MAMAExpDLL mama_status mamaSubscription_createBasicWildCard (mamaSubscription subscription, mamaTransport transport, mamaQueue queue, const mamaWildCardMsgCallbacks *callbacks, const char *source, const char *symbol, void *closure)

  Crate and activate a wildcard subscription that may be actually activated later.

- MAMAExpDLL mama_status mamaSubscription_createSnapshot (mama-Subscription subscription, mamaQueue queue, const mamaMsgCallbacks *callbacks, mamaSource source, const char *symbol, void *closure)

Create a snapshot subscription for initial value only (no updates).

- MAMAExpDLL mama_status mamaSubscription_deactivate (mamaSubscription subscription)

  Deactivate a subscription.

- MAMAExpDLL mama_status mamaSubscription_deallocate (mamaSubscription subscription)

  Free the memory for a mamaSubscription which was allocated via a call to `mamaSubscription_allocate()` *This function will call* `mamaSubscription_destroy()` *if the subscription has not already been destroyed.*

- MAMAExpDLL mama_status mamaSubscription_destroy (mamaSubscription subscription)

  Destroy the subscription.

- MAMAExpDLL mama_status mamaSubscription_destroyEx (mamaSubscription subscription)

  This function will destroy the subscription and can be called from any thread.

- MAMAExpDLL mama_status mamaSubscription_getAppDataType (mamaSubscription subscription, uint8_t *appDataType)

  Retrieve the application-specific data type.

- MAMAExpDLL mama_status mamaSubscription_getClosure (mamaSubscription subscription, void **closure)

  Return the closure.

- MAMAExpDLL MamaLogLevel mamaSubscription_getDebugLevel (mamaSubscription subscription)

  Return the debug level for this subscription.

- MAMAExpDLL mama_status mamaSubscription_getItemClosure (mamaSubscription subscription, void **closure)

  Return the item closure for the current message for the subscription.

- MAMAExpDLL mama_status mamaSubscription_getMsgQualifierFilter (mamaSubscription subscription, int *ignoreDefinitelyDuplicate, int *ignorePossiblyDuplicate, int *ignoreDefinitelyDelayed, int *ignorePossiblyDelayed, int *ignoreOutOfSequence)

  Get the filters that discard message according to the message qualifier.

- MAMAExpDLL mama_status mamaSubscription_getPlatformError (mamaSubscription subsc, void **error)

Return the middleware-specific platform error.

- MAMAExpDLL mama_status mamaSubscription_getPreIntitialCacheSize (mamaSubscription subscription, int *result)

  Return the initial value cache size.

- MAMAExpDLL mama_status mamaSubscription_getQueue (mamaSubscription subscription, mamaQueue *queue)

  Return the `mamaQueue` *for this subscription.*

- MAMAExpDLL mama_status mamaSubscription_getReceivedInitial (mamaSubscription subscription, int *receivedInitial)

  Returns a value of 1 or 0 indicating whether the subscription has received an initial.

- MAMAExpDLL mama_status mamaSubscription_getRecoverGaps (mamaSubscription subscription, int *doesRecover)

  Whether the specified subscription will attempt to recover from sequence number gaps.

- MAMAExpDLL mama_status mamaSubscription_getRequiresInitial (mamaSubscription subscription, int *requiresInitial)

  Returns a value of 1 or 0 indicating whether this subscription is interested in initial values.

- MAMAExpDLL mama_status mamaSubscription_getRetries (mamaSubscription subscription, int *retries)

  Retrieve the retries.

- MAMAExpDLL mama_status mamaSubscription_getServiceLevel (mamaSubscription subscription, mamaServiceLevel *serviceLevel)

  Get the service level for the specified subscription.

- MAMAExpDLL mama_status mamaSubscription_getServiceLevelOpt (mamaSubscription subscription, long *serviceLevel)

  Get the service level options for the specified subscription.

- MAMAExpDLL mama_status mamaSubscription_getSource (mamaSubscription subscription, const char **source)

  Return the source for the specified subscription.

- MAMAExpDLL mama_status mamaSubscription_getState (mamaSubscription subscription, mamaSubscriptionState *state)

This function will return the current state of the subscription, this function should be used in preference to the mamaSubscription_isActive or mamaSubscription_isValid functions.

- MAMAExpDLL mama_status mamaSubscription_getSubscriptionType (mamaSubscription subscription, mamaSubscriptionType *type)

  Returns the underlying `mamaSubscriptionType` *for the specified subscription.*

- MAMAExpDLL mama_status mamaSubscription_getSubscSymbol (mamaSubscription subscription, const char **symbol)

  Return the symbol as subscribed to (i.e., as mapped by the mapping function, if any, for the mamaTransport).

- MAMAExpDLL mama_status mamaSubscription_getSymbol (mamaSubscription subscription, const char **symbol)

  Return the symbol as provided by the user (before any symbol mapping).

- MAMAExpDLL mama_status mamaSubscription_getTimeout (mamaSubscription subscription, double *timeout)

  Retrieve the timeout.

- MAMAExpDLL mama_status mamaSubscription_getTransport (mamaSubscription subscription, mamaTransport *transport)

  Return the `mamaTransport` *for this subscription.*

- MAMAExpDLL mamaMsgCallbacks * mamaSubscription_getUserCallbacks (mamaSubscription subscription)

  Returns a pointer to the `mamaMsgCallbacks` *structure.*

- MAMAExpDLL int mamaSubscription_isActive (mamaSubscription subscription)

  Returns whether the subscription is active, note that this function has been deprecated, use mamaSubscription_getState instead.

- MAMAExpDLL int mamaSubscription_isValid (mamaSubscription subscription)

  Return whether the subscription is valid, note that this function has been deprecated, use mamaSubscription_getState instead.

- MAMAExpDLL mama_status mamaSubscription_muteCurrentTopic (mamaSubscription subscription)

  Mute the current topic for wildcard subscriptions.

- MAMAExpDLL mama_status mamaSubscription_setAppDataType (mamaSubscription subscription, uint8_t appDataType)

  Set the application-specific data type.

- MAMAExpDLL mama_status mamaSubscription_setDebugLevel (mamaSubscription subscription, MamaLogLevel level)

  Set the debug level for this subscription.

- MAMAExpDLL mama_status mamaSubscription_setGroupSizeHint (mamaSubscription subscription, int groupSizeHint)

  A hint as to the expected size of size of groups when making group subscriptions.

- MAMAExpDLL mama_status mamaSubscription_setItemClosure (mamaSubscription subscription, void *closure)

  Set the item closure for group subscriptions.

- MAMAExpDLL mama_status mamaSubscription_setMsgQualifierFilter (mamaSubscription subscription, int ignoreDefinitelyDuplicate, int ignorePossiblyDuplicate, int ignoreDefinitelyDelayed, int ignorePossiblyDelayed, int ignoreOutOfSequence)

  Set a filter to discard messages.

- MAMAExpDLL mama_status mamaSubscription_setPreIntitialCacheSize (mamaSubscription subscription, int cacheSize)

  Set the number of messages to cache for each symbol before the initial value arrives.

- MAMAExpDLL mama_status mamaSubscription_setRecoverGaps (mamaSubscription subscription, int doesRecover)

  Whether a subscription should attempt to recover from sequence number gaps.

- MAMAExpDLL mama_status mamaSubscription_setRequiresInitial (mamaSubscription subscription, int requiresInitial)

  Whether an initial value is required for the specified subscription.

- MAMAExpDLL mama_status mamaSubscription_setServiceLevel (mamaSubscription subscription, mamaServiceLevel serviceLevel, long serviceLevelOpt)

  Set the service level.

- MAMAExpDLL mama_status mamaSubscription_setRetries (mamaSubscription subscription, int retries)

  Set the number of retries when requesting recaps.

- MAMAExpDLL mama_status mamaSubscription_setSubscriptionType (mamaSubscription subscription, mamaSubscriptionType type)

  Set the subscription type for the subscription being created.

- MAMAExpDLL mama_status mamaSubscription_setSymbol (mamaSubscription subscription, const char *symbol)

  Set the symbol (e.g.

- MAMAExpDLL mama_status mamaSubscription_setTimeout (mamaSubscription subscription, double timeout)

  Set the timeout for this subscription.

- MAMAExpDLL mama_status mamaSubscription_setup (mamaSubscription subscription, mamaQueue queue, const mamaMsgCallbacks *callbacks, mamaSource source, const char *symbol, void *closure)

  Set the parameters for a subscription that may be actually activated later.

- MAMAExpDLL mama_status mamaSubscription_setup2 (mamaSubscription subscription, mamaTransport transport, mamaQueue queue, const mamaMsgCallbacks *callbacks, const char *sourceName, const char *symbol, void *closure)

  Set the parameters for a subscription that may be actually activated later.

- MAMAExpDLL mama_status mamaSubscription_setupBasicWildCard (mamaSubscription subscription, mamaTransport transport, mamaQueue queue, const mamaWildCardMsgCallbacks *callbacks, const char *source, const char *symbol, void *closure)

  Set the parameters for a wildcard subscription that may be actually activated later.

- MAMAExpDLL const char * mamaSubscription_stringForState (mamaSubscriptionState state)

  This function will return the string representation of the subscription state.

5.37.1 Typedef Documentation

5.37.1.1 typedef void(MAMACALLTYPE *) wombat_subscriptionCreateCB(mamaSubscription subscription, void *closure)

Function invoked when subscription creation is complete, and before any calls to `wombat_subscriptionOnMsgCB`.

Parameters:

subscription The subscription.

closure The closure passed to the mamaSubscription_create function.

5.37.1.2 typedef void(MAMACALLTYPE ∗) wombat_subscriptionDestroyCB(mamaSubscription subscription, void ∗closure)

Function invoked when a subscription has been completely destroyed, the client can have confidence that no further messages will be placed on the queue for this subscription.

Parameters:

subscription the `MamaSubscription`.

closure The closure passed to the mamaSubscription_create function.

5.37.1.3 typedef void(MAMACALLTYPE ∗) wombat_subscriptionErrorCB(mamaSubscription subscription, mama_status status, void ∗platformError, const char ∗subject, void ∗closure)

Invoked if an error occurs during prior to subscription creation or if the subscription receives a message for an unentitled subject.

If the status is `MAMA_MSG_STATUS_NOT_ENTITTLED` the subject parameter is the specific unentitled subject. If the subscription subject contains wildcards, the subscription may still receive messages for other entitled subjects. Note wildcard subscriptions are not supported on all platforms.

Parameters:

subscription The subscription.

status The error code.

platformError Third party, platform specific messaging error.

subject The subject for NOT_ENTITLED

closure The closure passed to the mamaSubscription_create function.

5.37.1.4 typedef void(MAMACALLTYPE ∗) wombat_subscriptionGapCB(mamaSubscription subscription, void ∗closure)

Function invoked when a sequence number gap is detected.

At this point the topic is considered stale and the subscription will not receive further messages until the feed handler satisfies a recap request.

Parameters:

subscription the `MamaSubscription`.

closure The closure passed to the mamaSubscription_create function.

5.37.1.5 typedef void(MAMACALLTYPE ∗) wombat_subscriptionOnMsgCB(mamaSubscription subscription, mamaMsg msg, void ∗closure, void ∗itemClosure)

Invoked when a message arrives.

Parameters:

subscription the `MamaSubscription`.

msg The mamaMsg.

closure The closure passed to the mamaSubscription_create function.

itemClosure The item closure for the subscription can be set with mamaSubscription_setItemClosure.

5.37.1.6 typedef void(MAMACALLTYPE ∗) wombat_subscriptionQualityCB(mamaSubscription subscription, mamaQuality quality, const char ∗symbol, short cause, const void ∗platformInfo, void ∗closure)

Invoked to indicate a data quality event.

Parameters:

subscription the `MamaSubscription`.

quality The new quality.

symbol The symbol.

cause The cause of the data quality event.

platformInfo Info associated with the data quality event.

closure The closure passed to the mamaSubscription_create function.

The cause and platformInfo are supplied only by some middlewares. The information provided by platformInfo is middleware specific. The following middlewares are supported:

tibrv: provides the char∗ version of the tibrv advisory message.

5.37.1.7 typedef void(MAMACALLTYPE *) wombat_subscriptionRecapCB(mamaSubscription subscription, void *closure)

Function invoked when a recap is requested upon detecting a sequence number gap.

Parameters:

subscription the `MamaSubscription`.

closure The closure passed to the mamaSubscription_create function.

5.37.1.8 typedef void(MAMACALLTYPE *) wombat_subscriptionWildCardOnMsgCB(mamaSubscription subscription, mamaMsg msg, const char *topic, void *closure, void *itemClosure)

Invoked when a message arrives for a wild card subscrption.

Parameters:

subscription the `MamaSubscription`.

msg The mamaMsg.

topic The symbol.

closure The closure passed to the mamaSubscription_create function.

itemClosure Symbol specific closure.

5.37.1.9 typedef struct mamaMsgCallbacks_ mamaMsgCallbacks

A convenience structure for passing the callbacks to the subscription factory methods.

5.37.1.10 typedef struct mamaWildCardMsgCallbacks_ mamaWildCardMsgCallbacks

A convenience structure for passing the callbacks to wild card subscription factory methods.

5.37.2 Enumeration Type Documentation

5.37.2.1 enum preInitialScheme

MAMA can cache messages which arrive prior to an initial value.

There are two options in how and when these message are used.

PRE_INITIAL_SCHEME_ON_GAP (default) MAMA waits until the message after the initialvalue and only tries to use the cache if a gap is detected. This reduces the need to reap the subscription.

PRE_INITIAL_SCHEME_ON_INITIAL MAMA checks the pre-initial cache to see if there are any messages which sequentially follow the initial value and pass them to the application.

Enumerator:

PRE_INITIAL_SCHEME_ON_GAP

PRE_INITIAL_SCHEME_ON_INITIAL

```
{
    PRE_INITIAL_SCHEME_ON_GAP,
    PRE_INITIAL_SCHEME_ON_INITIAL

} preInitialScheme;
```

5.37.2.2 enum mamaSubscriptionState

The enumeration describes the state of the subsubscription at any time.

The state can be obtained by calling the mamaSubscription_getState function. A string representation of this state can be obtained by calling the mamaSubscription_string-ForState function.

Enumerator:

MAMA_SUBSCRIPTION_UNKNOWN

MAMA_SUBSCRIPTION_ALLOCATED

MAMA_SUBSCRIPTION_SETUP

MAMA_SUBSCRIPTION_ACTIVATING

MAMA_SUBSCRIPTION_ACTIVATED

MAMA_SUBSCRIPTION_DEACTIVATING

MAMA_SUBSCRIPTION_DEACTIVATED

MAMA_SUBSCRIPTION_DESTROYING

MAMA_SUBSCRIPTION_DESTROYED

MAMA_SUBSCRIPTION_DEALLOCATING

MAMA_SUBSCRIPTION_DEALLOCATED

MAMA_SUBSCRIPTION_REACTIVATING

```
{
    /* The state of the subscription is unknown. */
    MAMA_SUBSCRIPTION_UNKNOWN       = 0,

    /* The subscription has been allocated in memory. */
    MAMA_SUBSCRIPTION_ALLOCATED     = 1,

    /* Initial setup work has been done, mamaSubscription_activate must still be called. Note that this
     * for market data subscriptions.
     */
    MAMA_SUBSCRIPTION_SETUP         = 2,

    /* The subscription is now on the throttle queue waiting to be fully activated. */
    MAMA_SUBSCRIPTION_ACTIVATING    = 3,

    /* The subscription is now fully activated and is processing messages. */
    MAMA_SUBSCRIPTION_ACTIVATED     = 4,

    /* The subscription is being de-activated, it will not be fully deactivated until the middleware re
    MAMA_SUBSCRIPTION_DEACTIVATING  = 5,

    /* The subscription has been de-activated, messages are no longer being processed. */
    MAMA_SUBSCRIPTION_DEACTIVATED   = 6,

    /* The subscription is being destroyed, but waiting on deactivation to complete*/
    MAMA_SUBSCRIPTION_DESTROYING    = 7,

    /* The subscription has been fully destroyed. */
    MAMA_SUBSCRIPTION_DESTROYED     = 8,

     /* The subscription is in the process of being de-allocated, but waiting on deactivation to comple
     MAMA_SUBSCRIPTION_DEALLOCATING  = 9,

     /* The subscription has been de-allocated. This state is only temporary and exists until such poir
      * memory is freed. It is provided so that a log entry will be written out.
      */
     MAMA_SUBSCRIPTION_DEALLOCATED   = 10,

     /* The subscription is being re-activated, it will not be fully reactivated until deactivation com
     MAMA_SUBSCRIPTION_REACTIVATING  = 11

} mamaSubscriptionState;
```

5.37.2.3 enum dqStartegyScheme

Enumerator:

DQ_SCHEME_DELIVER_ALL

DQ_SCHEME_INGORE_DUPS

```
{
    DQ_SCHEME_DELIVER_ALL,
    DQ_SCHEME_INGORE_DUPS
} dqStartegyScheme;
```

5.37.2.4 enum dqftStrategyScheme

Enumerator:

DQ_FT_DO_NOT_WAIT_FOR_RECAP

DQ_FT_WAIT_FOR_RECAP

```
{
    DQ_FT_DO_NOT_WAIT_FOR_RECAP,
    DQ_FT_WAIT_FOR_RECAP
}dqftStrategyScheme;
```

5.37.3 Function Documentation

5.37.3.1 MAMAExpDLL mama_status mamaSubscription_activate (mamaSubscription *subscription*)

Activate a subscription that has been set up by mamaSubscription_setup().

Subscription creation actually occurs on the throttle queue. An individual subscription cannot be assumed to be fully created until its onCreate() callback has been successfully invoked. The subscription rate can be governed via the `mamaTransport_setOutboundThrottle ()` function. Any subscription properties should be set prior to calling this function.

Parameters:

← ***subscription*** The subscription.

Returns:

Mama status code can be one of: MAMA_STATUS_SUBSCRIPTION_INVALID_TYPE - the type of subscription does not support activation. This will be returned if activate is called for a basic subscription. MAMA_STATUS_SUBSCRIPTION_INVALID_STATE - the subscription cannot be activated as it is in the wrong state. MAMA_STATUS_OK.

5.37.3.2 MAMAExpDLL mama_status mamaSubscription_allocate (mamaSubscription * *result*)

Allocate memory for a new subscription.

The subscription is not actually created until a call to one of the create() functions is made. Memory must be freed using the `mamaSubscription_deallocate()` function.

Parameters:

→ ***result*** Where the address of the new subscription will be written.

Returns:

mama_status value can be one of MAMA_STATUS_NOMEM MAMA_-STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.3 MAMAExpDLL int mamaSubscription_checkDebugLevel (mamaSubscription *subscription*, MamaLogLevel *level*)

Return whether the debug level for this subscription equals or exceeds some level.

Parameters:

← ***subscription*** The subscription.

← ***level*** The debug level to check.

Returns:

whether the level equals or exceeds the set level for this subscription.

5.37.3.4 MAMAExpDLL mama_status mamaSubscription_create (mamaSubscription *subscription*, mamaQueue *queue*, const mamaMsgCallbacks ∗ *callbacks*, mamaSource *source*, const char ∗ *symbol*, void ∗ *closure*)

Create and activate subscription.

This is effectively a pair of calls to mamaSubscription_setup() and mama-Subscription_activate().

Parameters:

← ***subscription*** The subscription.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***source*** The mamaSource identifying the publisher for this symbol.

← ***symbol*** The symbol name.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NOMEM MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.5 MAMAExpDLL mama_status mamaSubscription_createBasic (mamaSubscription *subscription*, mamaTransport *transport*, mamaQueue *queue*, const mamaMsgCallbacks ∗ *callbacks*, const char ∗ *symbol*, void ∗ *closure*)

Create a basic subscription without marketdata semantics.

Parameters:

← ***subscription*** The subscription.

← ***transport*** The transport to use. Must be a basic transport.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***symbol*** The symbol on which message are being published.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NULL_ARG MAMA_STATUS_OK

5.37.3.6 MAMAExpDLL mama_status mamaSubscription_createBasicWild-Card (mamaSubscription *subscription*, mamaTransport *transport*, mamaQueue *queue*, const mamaWildCardMsgCallbacks ∗ *callbacks*, const char ∗ *source*, const char ∗ *symbol*, void ∗ *closure*)

Crate and activate a wildcard subscription that may be actually activated later.

Activate the subscription using mamaSubscription_activate().

The topic must be a valid wildcard topic for the underlying middleware.

Applications may set and retireve per-topic closures using mamaSubscription_setItem-Closure() however this method is only guaranteed to set the correct closure if called

while in the msg callback. It will be the topicClosure argument for subsequent callback invocations for the current topic. The topicClosure argument to the callback is NULL prior to an application setting the closure for the topic.

NOTE: For WMW a source with a NULL symbol parameter creates a "transport" subscription that receives all messages on the transport and bypasses the naming service. A publishing transport can be assigned a name with the publish_name property.

Parameters:

← ***subscription*** The subscription.

← ***transport*** The transport to use.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***source*** The source name of the feed handler to provide the subscription.

← ***symbol*** The symbol name.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NULL_ARG MAMA_STATUS_OK

5.37.3.7 MAMAExpDLL mama_status mamaSubscription_createSnapshot (mamaSubscription *subscription*, mamaQueue *queue*, const mamaMsgCallbacks ∗ *callbacks*, mamaSource *source*, const char ∗ *symbol*, void ∗ *closure*)

Create a snapshot subscription for initial value only (no updates).

This function is equivalent to mamaSubscription_create () with svcLevel set to MAMA_SERVICE_LEVEL_SNAPSHOT and default arguments for type, svcLevel-Opt, requiresInitial, retries, timeout.

Parameters:

← ***subscription*** The subscription.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***source*** The mamaSource identifying the publisher of data for the specified symbol.

← ***symbol*** The symbol name.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NULL_ARG MAMA_STATUS_OK

5.37.3.8 MAMAExpDLL mama_status mamaSubscription_deactivate (mamaSubscription *subscription*)

Deactivate a subscription.

The subscription can be reactivated if desired using mamaSubscription_activate(). Note that the subscription will not be fully deactivated until the onDestroy callback is received.

Parameters:

← ***subscription*** The subscription.

Returns:

mama status code can be one of: MAMA_STATUS_SUBSCRIPTION_-INVALID_TYPE - the type of subscription does not support deactivation. This will be returned if deactivate is called for a basic subscription. MAMA_STATUS_-SUBSCRIPTION_INVALID_STATE - the subscription cannot be deactivated as it is in the wrong state. MAMA_STATUS_OK.

5.37.3.9 MAMAExpDLL mama_status mamaSubscription_deallocate (mamaSubscription *subscription*)

Free the memory for a mamaSubscription which was allocated via a call to `mamaSubscription_allocate()` This function will call `mamaSubscription_destroy()` if the subscription has not already been destroyed.

Parameters:

← ***subscription*** The subscription to be deallocated.

Returns:

mama status code can be one of: MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATS_NULL_ARG MAMA_STATUS_OK.

5.37.3.10 MAMAExpDLL mama_status mamaSubscription_destroy (mamaSubscription *subscription*)

Destroy the subscription.

This function must be called to destroy resources associated with the subscription. It sends an unsubscribe to the feed handler for subscription management. This function does not free the memory associated with the subscription. mamaSubscription_create() can be called again after this function has been called. This function must be called from the same thread dispatching on the associated event queue unless both the default queue and dispatch queue are not actively dispatching.

Parameters:

← ***subscription*** The subscription.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_-STATUS_SUBSCRIPTION_INVALID_STATE MAMA_STATUS_OK

5.37.3.11 MAMAExpDLL mama_status mamaSubscription_destroyEx (mamaSubscription *subscription*)

This function will destroy the subscription and can be called from any thread.

Note that the subscription will not be fully destroyed until the onDestroy callback is received. To destroy from the dispatching thread the mamaSubscription_destroy function should be used in preference.

Parameters:

← ***subscription*** The subscription.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_-STATUS_OK

5.37.3.12 MAMAExpDLL mama_status mamaSubscription_getAppDataType (mamaSubscription *subscription*, uint8_t * *appDataType*)

Retrieve the application-specific data type.

Parameters:

← ***subscription*** The subscription.

→ ***appDataType*** A pointer to a uint8_t to hold the value.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.13 MAMAExpDLL mama_status mamaSubscription_getClosure (mamaSubscription *subscription*, void ∗∗ *closure*)

Return the closure.

Parameters:

← ***subscription*** The subscription.

→ ***closure*** holds the result.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.14 MAMAExpDLL MamaLogLevel mamaSubscription_getDebugLevel (mamaSubscription *subscription*)

Return the debug level for this subscription.

Parameters:

← ***subscription*** The subscription.

Returns:

the debug level for this subscription.

5.37.3.15 MAMAExpDLL mama_status mamaSubscription_getItemClosure (mamaSubscription *subscription*, void ∗∗ *closure*)

Return the item closure for the current message for the subscription.

See `setItemClosure` for more detail.

Parameters:

← ***subscription*** The subscription.

→ ***closure*** The address to where the closure should be written.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.16 MAMAExpDLL mama_status mamaSubscription_getMsgQualifierFilter (mamaSubscription *subscription*, int ∗ *ignoreDefinitelyDuplicate*, int ∗ *ignorePossiblyDuplicate*, int ∗ *ignoreDefinitelyDelayed*, int ∗ *ignorePossiblyDelayed*, int ∗ *ignoreOutOfSequence*)

Get the filters that discard message according to the message qualifier.

Parameters:

← ***subscription*** The subscription from which the filer is being obtained.

→ ***ignoreDefinitelyDuplicate*** If true callbacks will not be invoked for messages where MamaMsg::getIsDefinitelyDuplicate returns true.

→ ***ignorePossiblyDuplicate*** If true callbacks will not be invoked for messages where MamaMsg::getIsPossiblyDuplicate returns true.

→ ***ignoreDefinitelyDelayed*** If true callbacks will not be invoked for messages where MamaMsg::getIsDefinitelyDelayed returns true.

→ ***ignorePossiblyDelayed*** If true callbacks will not be invoked for messages where MamaMsg::getIsPossiblyDelayed returns true.

→ ***ignoreOutOfSequence*** If true callbacks will not be invoked for messages where MamaMsg::getIsOutOfSequence returns true.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.17 MAMAExpDLL mama_status mamaSubscription_getPlatformError (mamaSubscription *subsc*, void ∗∗ *error*)

Return the middleware-specific platform error.

When a mamaSubscription_ method return MAMA_STATUS_PLATFORM the error will be the result from the underlying platform.

Parameters:

← ***subsc*** The subscription.

→ ***error*** a pointer to a void pointer to hold the result.

Returns:

mama_status return code can be one of: MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.18 MAMAExpDLL mama_status mamaSubscription_getPreIntitialCacheSize (mamaSubscription *subscription*, int * *result*)

Return the initial value cache size.

Parameters:

← ***subscription*** The subscription.

→ ***result*** A pointer to an int for the result.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.19 MAMAExpDLL mama_status mamaSubscription_getQueue (mamaSubscription *subscription*, mamaQueue * *queue*)

Return the `mamaQueue` for this subscription.

Parameters:

← ***subscription*** The subscription.

→ ***queue*** A pointer to hold the queue.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.20 MAMAExpDLL mama_status mamaSubscription_getReceivedInitial (mamaSubscription *subscription*, int * *receivedInitial*)

Returns a value of 1 or 0 indicating whether the subscription has received an initial.

Parameters:

← ***subscription*** The subscription

→ ***receivedInitial*** Will be 1 if an initial has been recieved else 0.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.21 MAMAExpDLL mama_status mamaSubscription_getRecoverGaps (mamaSubscription *subscription*, int ∗ *doesRecover*)

Whether the specified subscription will attempt to recover from sequence number gaps.

Parameters:

subscription[in] The subscription

doesRecover[out] 0 - does not recover, 1 - does attempt to recover

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.22 MAMAExpDLL mama_status mamaSubscription_getRequiresInitial (mamaSubscription *subscription*, int ∗ *requiresInitial*)

Returns a value of 1 or 0 indicating whether this subscription is interested in initial values.

Parameters:

← ***subscription*** The subscription

→ ***requiresInitial*** 1 if an initial is required else 0.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.23 MAMAExpDLL mama_status mamaSubscription_getRetries (mamaSubscription *subscription*, int * *retries*)

Retrieve the retries.

Parameters:

← ***subscription*** The subscription.

→ ***retries*** A pointer to a int to hold the value.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.24 MAMAExpDLL mama_status mamaSubscription_getServiceLevel (mamaSubscription *subscription*, mamaServiceLevel * *serviceLevel*)

Get the service level for the specified subscription.

Parameters:

← ***subscription*** The subscription for which the service level is being obtained.

→ ***serviceLevel*** Address to where the service level will be written

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.25 MAMAExpDLL mama_status mamaSubscription_getServiceLevelOpt (mamaSubscription *subscription*, long * *serviceLevel*)

Get the service level options for the specified subscription.

Parameters:

← ***subscription*** The subscription for which the service level is being obtained.

→ ***serviceLevel*** Address to where the service level options will be written

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.26 MAMAExpDLL mama_status mamaSubscription_getSource (mamaSubscription *subscription*, const char ∗∗ *source*)

Return the source for the specified subscription.

Note that this function just returns a pointer to the source inside the subscription and does not allocate any memory.

Parameters:

← ***subscription*** The subscription.

→ ***source*** holds the result, (do not free).

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.27 MAMAExpDLL mama_status mamaSubscription_getState (mamaSubscription *subscription*, mamaSubscriptionState ∗ *state*)

This function will return the current state of the subscription, this function should be used in preference to the mamaSubscription_isActive or mamaSubscription_isValid functions.

This function is thread-safe.

Parameters:

← ***subscription*** The subscription to return the state for.

→ ***state*** To return the state, this can be one of the mamaSubscriptionState enumeration values.

Returns:

mama_status return code can be one of: MAMA_STATUS_OK MAMA_-STATUS_NULL_ARG

5.37.3.28 MAMAExpDLL mama_status mamaSubscription_getSubscriptionType (mamaSubscription *subscription*, mamaSubscriptionType ∗ *type*)

Returns the underlying `mamaSubscriptionType` for the specified subscription.

Parameters:

← ***subscription*** The subscription for which the type will be returned.

→ ***type*** Address to which the type will be written.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.29 MAMAExpDLL mama_status mamaSubscription_getSubscSymbol (mamaSubscription *subscription*, const char ∗∗ *symbol*)

Return the symbol as subscribed to (i.e., as mapped by the mapping function, if any, for the mamaTransport).

Note that this function just returns a pointer to the symbol inside the subscription and does not allocate any memory.

Parameters:

← ***subscription*** The subscription.

→ ***symbol*** holds the result, do not free.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.30 MAMAExpDLL mama_status mamaSubscription_getSymbol (mamaSubscription *subscription*, const char ∗∗ *symbol*)

Return the symbol as provided by the user (before any symbol mapping).

Note that this function just returns a pointer to the symbol inside the subscription and does not allocate any memory.

Parameters:

← ***subscription*** The subscription.

→ ***symbol*** holds the result, (do not free).

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.31 MAMAExpDLL mama_status mamaSubscription_getTimeout (mamaSubscription *subscription*, double * *timeout*)

Retrieve the timeout.

Parameters:

← ***subscription*** The subscription.

→ ***timeout*** A pointer to a double to hold the value.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.32 MAMAExpDLL mama_status mamaSubscription_getTransport (mamaSubscription *subscription*, mamaTransport * *transport*)

Return the `mamaTransport` for this subscription.

Parameters:

← ***subscription*** The subscription.

→ ***transport*** A pointer to hold the transport.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.33 MAMAExpDLL mamaMsgCallbacks* mamaSubscription_getUserCallbacks (mamaSubscription *subscription*)

Returns a pointer to the `mamaMsgCallbacks` structure.

Parameters:

← ***subscription*** The `mamaSubscription`

Returns:

pointer to the callbacks container structure MAMA_STATUS_OK

5.37.3.34 MAMAExpDLL int mamaSubscription_isActive (mamaSubscription *subscription*)

Returns whether the subscription is active, note that this function has been deprecated, use mamaSubscription_getState instead.

Parameters:

← ***subscription*** The subscription.

Returns:

whether the subscription is active.

5.37.3.35 MAMAExpDLL int mamaSubscription_isValid (mamaSubscription *subscription*)

Return whether the subscription is valid, note that this function has been deprecated, use mamaSubscription_getState instead.

Parameters:

← ***subscription*** The subscription

Returns:

whether the subscription is valid.

5.37.3.36 MAMAExpDLL mama_status mamaSubscription_muteCurrentTopic (mamaSubscription *subscription*)

Mute the current topic for wildcard subscriptions.

Applications should only invoke this method from the message callback. When invoked for a WMW transport subsription, the subscription will cease receiving calbacks on the current topic.

This method only works for WMW "transport" subscriptions which are Wild card subscriptions created with a source of a named publisher, and a NULL topic.

Parameters:

← ***subscription*** The subscription

Returns:

mama_status return code can be one of: MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.37 MAMAExpDLL mama_status mamaSubscription_setAppDataType (mamaSubscription *subscription*, uint8_t *appDataType*)

Set the application-specific data type.

Parameters:

← ***subscription*** The subscription.

← ***appDataType*** The application-specific data type.

Returns:

mama_status return code can be one of:

MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.38 MAMAExpDLL mama_status mamaSubscription_setDebugLevel (mamaSubscription *subscription*, MamaLogLevel *level*)

Set the debug level for this subscription.

Parameters:

← ***subscription*** The subscription.

← ***level*** The new debug level.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.39 MAMAExpDLL mama_status mamaSubscription_setGroupSizeHint (mamaSubscription *subscription*, int *groupSizeHint*)

A hint as to the expected size of size of groups when making group subscriptions.

Parameters:

← ***subscription*** The subscription

← ***groupSizeHint*** The size of the groups

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.40 MAMAExpDLL mama_status mamaSubscription_setItemClosure (mamaSubscription *subscription*, void * *closure*)

Set the item closure for group subscriptions.

Group subscriptions receive updates for multiple symbols. This method allows calls to set a per-symbol closure which will be passed as the fourth argument to subsequent calls to the onMsg callback. This method may only be called during the onMsg callback.

This method also sets per-topic closures for wildcard subscriptions.

Setting the item closure for a non-group subscription provides a second closure.

Parameters:

← ***subscription*** The subscription.

← ***closure*** The new item closure.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.41 MAMAExpDLL mama_status mamaSubscription_setMsgQualifierFilter (mamaSubscription *subscription*, int *ignoreDefinitelyDuplicate*, int *ignorePossiblyDuplicate*, int *ignoreDefinitelyDelayed*, int *ignorePossiblyDelayed*, int *ignoreOutOfSequence*)

Set a filter to discard messages.

Parameters:

← ***subscription*** The subscription on which the filter is being set.

← ***ignoreDefinitelyDuplicate*** If true callbacks will not be invoked for messages where MamaMsg::getIsDefinitelyDuplicate returns true.

← ***ignorePossiblyDuplicate*** If true callbacks will not be invoked for messages where MamaMsg::getIsPossiblyDuplicate returns true.

← ***ignoreDefinitelyDelayed*** If true callbacks will not be invoked for messages where MamaMsg::getIsDefinitelyDelayed returns true.

← ***ignorePossiblyDelayed*** If true callbacks will not be invoked for messages where MamaMsg::getIsPossiblyDelayed returns true.

← ***ignoreOutOfSequence*** If true callbacks will not be invoked for messages where MamaMsg::getIsOutOfSequence returns true.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.42 MAMAExpDLL mama_status mamaSubscription_setPreIntitialCacheSize (mamaSubscription *subscription*, int *cacheSize*)

Set the number of messages to cache for each symbol before the initial value arrives.

This allows the subscription to recover when the initial value arrives late (after a subsequent trade or quote already arrived).

For group subscription, a separate cache is used for each group member.

The default is 10.

Parameters:

← ***subscription*** The subscription.

← ***cacheSize*** The size of the cache.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.43 MAMAExpDLL mama_status mamaSubscription_setRecoverGaps (mamaSubscription *subscription*, int *doesRecover*)

Whether a subscription should attempt to recover from sequence number gaps.

Parameters:

← ***subscription*** The subscription

→ ***doesRecover*** 0 indicates not to recover. 1 The subscription will attempt to recover via a recap request.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.44 MAMAExpDLL mama_status mamaSubscription_setRequiresInitial (mamaSubscription *subscription*, int *requiresInitial*)

Whether an initial value is required for the specified subscription.

This only applies to market data subscriptions and not to basic subscriptions. Default value of 1 indicating that initial values are required.

Parameters:

← ***subscription*** The subscription

← ***requiresInitial*** [1|0] Whether to request an initial value or not.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.45 MAMAExpDLL mama_status mamaSubscription_setServiceLevel (mamaSubscription *subscription*, mamaServiceLevel *serviceLevel*, long *serviceLevelOpt*)

Set the service level.

This method must be invoked before createXXX ().

Parameters:

← ***subscription*** The subscription for which the service level is being set.

← ***serviceLevel*** The service level of the subscription (real time, snapshot, etc.). The default is MAMA_SERVICE_LEVEL_REAL_TIME.

← ***serviceLevelOpt*** An optional argument for certain service levels. (Defaults to 0 - for future use)

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.46 MAMAExpDLL mama_status mamaSubscription_setRetries (mamaSubscription *subscription*, int *retries*)

Set the number of retries when requesting recaps.

Parameters:

← ***subscription*** The subscription.

← ***retries*** The number of retries.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.47 MAMAExpDLL mama_status mamaSubscription_setSubscriptionType (mamaSubscription *subscription*, mamaSubscriptionType *type*)

Set the subscription type for the subscription being created.

If not called the subscription type defaults to MAMA_SUBSC_TYPE_NORMAL. See `mamaSubscriptionType` enum for valid values.

Parameters:

← ***subscription*** The subscription for which the type is being set.

← ***type*** The type of the subscription being created.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.48 MAMAExpDLL mama_status mamaSubscription_setSymbol (mamaSubscription *subscription*, const char * *symbol*)

Set the symbol (e.g.

to change symbol mapping).

Parameters:

← ***subscription*** The subscription.

← ***symbol.***

Returns:

mama_status return code can be one of: MAMA_STATUS_NOMEM MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.49 MAMAExpDLL mama_status mamaSubscription_setTimeout (mamaSubscription *subscription*, double *timeout*)

Set the timeout for this subscription.

The timeout is used for requesting recaps.

Parameters:

← ***subscription*** The subscription.

← ***timeout*** The timeout in seconds.

Returns:

mama_status return code can be one of: MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.50 MAMAExpDLL mama_status mamaSubscription_setup (mamaSubscription *subscription*, mamaQueue *queue*, const mamaMsgCallbacks * *callbacks*, mamaSource *source*, const char * *symbol*, void * *closure*)

Set the parameters for a subscription that may be actually activated later.

Activate the subscription using mamaSubscription_activate().

Parameters:

← ***subscription*** The subscription.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***source*** The MAMA source to use.

← ***symbol*** The symbol name.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NOMEM MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.51 MAMAExpDLL mama_status mamaSubscription_setup2 (mamaSubscription *subscription*, mamaTransport *transport*, mamaQueue *queue*, const mamaMsgCallbacks * *callbacks*, const char * *sourceName*, const char * *symbol*, void * *closure*)

Set the parameters for a subscription that may be actually activated later.

Activate the subscription using mamaSubscription_activate().

Parameters:

← ***subscription*** The subscription.

← ***transport*** The MAMA transport.

← ***queue*** The mama queue.

← ***callbacks*** The mamaMsgCallbacks structure containing the callback functions.

← ***sourceName*** The source name..

← ***symbol*** The symbol name.

← ***closure*** The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NOMEM MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.52 MAMAExpDLL mama_status mamaSubscription_setupBasicWild-Card (mamaSubscription *subscription*, mamaTransport *transport*, mamaQueue *queue*, const mamaWildCardMsgCallbacks * *callbacks*, const char * *source*, const char * *symbol*, void * *closure*)

Set the parameters for a wildcard subscription that may be actually activated later.

Activate the subscription using mamaSubscription_activate().

The topic must be a valid wildcard topic for the underlying middleware.

Applications may set and retireve per-topic closures using mamaSubscription_setItem-Closure() however this method is only guaranteed to set the correct closure if called while in the msg callback. It will be the topicClosure argument for subsequent callback invocations for the current topic. The topicClosure argument to the callback is NULL prior to an application setting the closure for the topic.

Parameters:

subscription The subscription.

transport The transport to use.

queue The mama queue.

callbacks The mamaMsgCallbacks structure containing the callback functions.

source The source name of the feed handler to provide the subscription.

symbol The symbol name.

closure The closure will be passed to subsequent callback invocations for this subscription.

Returns:

mama_status return code can be one of: MAMA_STATUS_INVALID_ARG MAMA_STATUS_INVALID_QUEUE MAMA_STATUS_SUBSCRIPTION_-INVALID_STATE MAMA_STATUS_NO_BRIDGE_IMPL MAMA_STATUS_-NOMEM MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.37.3.53 MAMAExpDLL const char* mamaSubscription_stringForState (mamaSubscriptionState *state*)

This function will return the string representation of the subscription state.

The subscription state can be obtained by calling the mamaSubscription_getState function.

Parameters:

← ***state*** The state to obtain the string representation for.

Returns:

A string representation of the state code or "State not recognised" if it is not recognised.

5.38 subscriptiontype.h File Reference

```
#include "wombat/port.h"
```

Enumerations

- enum mamaSubscriptionType {

  MAMA_SUBSC_TYPE_NORMAL = 0, MAMA_SUBSC_TYPE_GROUP = 1, MAMA_SUBSC_TYPE_BOOK = 2, MAMA_SUBSC_TYPE_BASIC = 3,

  MAMA_SUBSC_TYPE_DICTIONARY = 4, MAMA_SUBSC_TYPE_-SYMBOL_LIST = 5, MAMA_SUBSC_TYPE_SYMBOL_LIST_NORMAL = 6, MAMA_SUBSC_TYPE_SYMBOL_LIST_GROUP = 7,

  MAMA_SUBSC_TYPE_SYMBOL_LIST_BOOK = 8 }

Functions

- MAMAExpDLL const char * MamaSubscTypeStr (mamaSubscriptionType subtype)

5.38.1 Enumeration Type Documentation

5.38.1.1 enum mamaSubscriptionType

Enumerator:

MAMA_SUBSC_TYPE_NORMAL

MAMA_SUBSC_TYPE_GROUP

MAMA_SUBSC_TYPE_BOOK

MAMA_SUBSC_TYPE_BASIC

MAMA_SUBSC_TYPE_DICTIONARY

MAMA_SUBSC_TYPE_SYMBOL_LIST

MAMA_SUBSC_TYPE_SYMBOL_LIST_NORMAL

MAMA_SUBSC_TYPE_SYMBOL_LIST_GROUP

MAMA_SUBSC_TYPE_SYMBOL_LIST_BOOK

```
{
    MAMA_SUBSC_TYPE_NORMAL                  = 0,
    MAMA_SUBSC_TYPE_GROUP                   = 1,
    MAMA_SUBSC_TYPE_BOOK                    = 2,
    MAMA_SUBSC_TYPE_BASIC                   = 3, /* No Market data */
    MAMA_SUBSC_TYPE_DICTIONARY              = 4,
```

```
    MAMA_SUBSC_TYPE_SYMBOL_LIST              = 5,
    MAMA_SUBSC_TYPE_SYMBOL_LIST_NORMAL       = 6,
    MAMA_SUBSC_TYPE_SYMBOL_LIST_GROUP        = 7,
    MAMA_SUBSC_TYPE_SYMBOL_LIST_BOOK         = 8
} mamaSubscriptionType;
```

5.38.2 Function Documentation

5.38.2.1 MAMAExpDLL const char* MamaSubscTypeStr (mamaSubscriptionType *subtype*)

5.39 symbollist.h File Reference

```
#include "mama/mama.h"
#include <mama/config.h>
#include "mama/symbollisttypes.h"
```

Typedefs

- typedef void ∗ closure
- typedef void ∗ closure
- typedef mamaSymbolListMember member
- typedef mamaSymbolListMember void ∗ closure
- typedef void ∗ closure

Functions

- typedef mama_status (MAMACALLTYPE ∗addSymbolCbType)(mamaSymbolListMember symbol

  Prototype for add symbol callback.

- typedef void (MAMACALLTYPE ∗mamaSymbolListIterateCompleteFunc)(mamaSymbolList symbolList

  The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

- MAMAExpDLL mama_status mamaSymbolList_addMember (mamaSymbolList symbolList, mamaSymbolListMember member)

  Add a symbol member to the symbol list, this will cause the add callback to be invoked if it has been installed.

- MAMAExpDLL mama_status mamaSymbolList_allocate (mamaSymbolList ∗symbolList)

  Allocate and initialize memory for a new symbolList, mamaSymbolList_deallocate should be called on the returned symbol list.

- MAMAExpDLL mamaSymbolListMember mamaSymbolList_allocateMember (mamaSymbolList symbolList)

  Allocate a symbolListMember.

- MAMAExpDLL mama_status mamaSymbolList_clear (mamaSymbolList symbolList, int membersToo)

Clear the symbol list.

- MAMAExpDLL mama_status mamaSymbolList_deallocate (mamaSymbolList symbolList)

  Free a symbol list created by mamaSymbolList_allocate.

- MAMAExpDLL mama_status mamaSymbolList_deallocateMember (mamaSymbolList symbolList, mamaSymbolListMember member)

  Deallocate a symbolListMember.

- MAMAExpDLL mama_status mamaSymbolList_deallocateWithMembers (mamaSymbolList symbolList)

  Free the memory for the symbolList and all members.

- MAMAExpDLL mama_status mamaSymbolList_findMember (const mamaSymbolList symbolList, const char *symbol, const char *source, mamaTransport transport, mamaSymbolListMember *member)

  Find a symbol member in the symbol list.

- MAMAExpDLL mama_status mamaSymbolList_getClosure (const mamaSymbolList symbolList, void **closure)

  Get the closure associated with the this symbolList.

- MAMAExpDLL mama_status mamaSymbolList_getSize (const mamaSymbolList symbolList, unsigned long *size)

  Get the size of the symbolList.

- MAMAExpDLL mama_status mamaSymbolList_iterate (mamaSymbolList symbolList, mamaSymbolListIterateMemberFunc memberFunc, mamaSymbolListIterateCompleteFunc completeFunc, void *iterateClosure)

  Iterate over the symbol list.

- MAMAExpDLL mama_status mamaSymbolList_removeMember (mamaSymbolList symbolList, const char *symbol, const char *source, mamaTransport transport, mamaSymbolListMember *member)

  Remove a symbol member from the symbol list.

- MAMAExpDLL mama_status mamaSymbolList_removeMemberByRef (mamaSymbolList symbolList, mamaSymbolListMember member)

  Remove a symbol member from the symbol list.

- MAMAExpDLL mama_status mamaSymbolList_setAddSymbolHandler (mamaSymbolList symbolList, addSymbolCbType addCb)

Registers the user defined add symbol callback with the symbolList.

- MAMAExpDLL mama_status mamaSymbolList_setClosure (mamaSymbolList symbolList, void *closure)

  Set the closure associated with the this symbolList.

- MAMAExpDLL mama_status mamaSymbolList_setRemoveSymbolHandler (mamaSymbolList symbolList, removeSymbolCbType removeCb)

  Registers the user defined remove symbol callback with the symbolList.

5.39.1 Typedef Documentation

5.39.1.1 typedef void* closure

5.39.1.2 typedef void* closure

5.39.1.3 typedef mamaSymbolListMember member

5.39.1.4 typedef mamaSymbolListMember void* closure

5.39.1.5 typedef void* closure

5.39.2 Function Documentation

5.39.2.1 typedef mama_status (MAMACALLTYPE * *removeSymbolCbType*)

Prototype for add symbol callback.

The registered add symbol callback is executed for each symbol added to the symbol list.

Parameters:

symbol The symbol to be added to the list

userData User-data to be associated with symbol

5.39.2.2 typedef void (MAMACALLTYPE * *mamaSymbolListIterateCompleteFunc*)

The callback invoked if an error occurs calling mama_startBackground() or when mama_startBackground() exits normally in which case status will be MAMA_STATUS_OK.

Parameters:

queue The mamaQueue for which the size limit has been exceeded. NULL if the queue is the default internal MAMA queue.

size The current number of events outstanding on the queue (if supported on the underlying middleware)

closure User supplied data set when the callback was registered. NULL in the case of the default MAMA queue as no closure can be specified when registering the data quality callbacks.

5.39.2.3 MAMAExpDLL mama_status mamaSymbolList_addMember (mamaSymbolList *symbolList*, mamaSymbolListMember *member*)

Add a symbol member to the symbol list, this will cause the add callback to be invoked if it has been installed.

Parameters:

← ***symbolList*** The symbolList.

← ***member*** The symbol member.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.4 MAMAExpDLL mama_status mamaSymbolList_allocate (mamaSymbolList ∗ *symbolList*)

Allocate and initialize memory for a new symbolList, mamaSymbolList_deallocate should be called on the returned symbol list.

Parameters:

→ ***symbolList*** The address of the allocated symbolList.

Returns:

mama_status return code can be one of MAMA_STATUS_NOMEM MAMA_-STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.5 MAMAExpDLL mamaSymbolListMember mamaSymbolList_allocateMember (mamaSymbolList *symbolList*)

Allocate a symbolListMember.

Parameters:

← ***symbolList*** The symbolList the member to be allocated in.

Returns:

Pointer to the new member.

5.39.2.6 MAMAExpDLL mama_status mamaSymbolList_clear (mamaSymbolList *symbolList*, int *membersToo*)

Clear the symbol list.

Parameters:

← ***symbolList*** The symbolList.

← ***membersToo*** Whether to also clear all members of the list.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.7 MAMAExpDLL mama_status mamaSymbolList_deallocate (mamaSymbolList *symbolList*)

Free a symbol list created by mamaSymbolList_allocate.

Parameters:

← ***symbolList*** The symbolList to be deallocated.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.8 MAMAExpDLL mama_status mamaSymbolList_deallocateMember (mamaSymbolList *symbolList*, mamaSymbolListMember *member*)

Deallocate a symbolListMember.

Parameters:

← ***symbolList*** The symbolList the member belongs to.

← ***member*** The member of the list to deallocate.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.9 MAMAExpDLL mama_status mamaSymbolList_deallocateWithMembers (mamaSymbolList *symbolList*)

Free the memory for the symbolList and all members.

Parameters:

symbolList The symbolList to be deallocated.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.10 MAMAExpDLL mama_status mamaSymbolList_findMember (const mamaSymbolList *symbolList*, const char * *symbol*, const char * *source*, mamaTransport *transport*, mamaSymbolListMember * *member*)

Find a symbol member in the symbol list.

Parameters:

← ***symbolList*** The symbolList.

← ***symbol*** The name of the symbol to find.

← ***source*** The source of the symbol to find.

← ***transport*** The tport of the symbol to find.

→ ***member*** The return symbol member (set to NULL if not found).

Returns:

mama_status return code can be one of MAMA_STATUS_INVALID_ARG - the member can't be found. MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.11 MAMAExpDLL mama_status mamaSymbolList_getClosure (const mamaSymbolList *symbolList*, void ∗∗ *closure*)

Get the closure associated with the this symbolList.

Parameters:

← ***symbolList*** The symbolList.

→ ***closure*** The closure

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.12 MAMAExpDLL mama_status mamaSymbolList_getSize (const mamaSymbolList *symbolList*, unsigned long ∗ *size*)

Get the size of the symbolList.

Parameters:

← ***symbolList*** The symbolList.

← ***size*** The size

Returns:

mama_status value can be one of: MAMA_STATUS_NULL_ARG MAMA_-STATUS_OK

5.39.2.13 MAMAExpDLL mama_status mamaSymbolList_iterate (mamaSymbolList *symbolList*, mamaSymbolListIterateMemberFunc *memberFunc*, mamaSymbolListIterateCompleteFunc *completeFunc*, void ∗ *iterateClosure*)

Iterate over the symbol list.

The "handler" function will be invoked for each

Parameters:

← ***symbolList*** The symbolList.

← ***memberFunc*** The function invoked for each symbol list member.

← ***completeFunc*** The function invoked upon completion.

← ***iterateClosure*** The closure passed to each callback

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.14 MAMAExpDLL mama_status mamaSymbolList_removeMember (mamaSymbolList *symbolList*, const char ∗ *symbol*, const char ∗ *source*, mamaTransport *transport*, mamaSymbolListMember ∗ *member*)

Remove a symbol member from the symbol list.

Parameters:

← ***symbolList*** The symbolList.

← ***symbol*** The name of the symbol to be removed.

← ***source*** The source of the symbol to be removed.

← ***transport*** The tport of the symbol to be removed.

→ ***member*** The return symbol member (set to NULL if not found). Note that this will be freed by the symbol list and should be used for reference only.

Returns:

mama_status return code can be one of MAMA_STATUS_INVALID_ARG - the member can't be found. MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.15 MAMAExpDLL mama_status mamaSymbolList_removeMemberByRef (mamaSymbolList *symbolList*, mamaSymbolListMember *member*)

Remove a symbol member from the symbol list.

Parameters:

symbolList The symbolList.

member The symbol to be removed.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.16 MAMAExpDLL mama_status mamaSymbolList_setAddSymbolHandler (mamaSymbolList *symbolList*, addSymbolCbType *addCb*)

Registers the user defined add symbol callback with the symbolList.

The registered callback will get called each time a symbol is added to the symbol list.

Parameters:

← ***symbolList*** The symbolList.

← ***addCb*** Pointer to the user defined callback. Must conform to function prototype `addSymbolCbType`.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.17 MAMAExpDLL mama_status mamaSymbolList_setClosure (mamaSymbolList *symbolList*, void ∗ *closure*)

Set the closure associated with the this symbolList.

Parameters:

← ***symbolList*** The symbolList.

← ***closure*** The closure

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.39.2.18 MAMAExpDLL mama_status mamaSymbolList_setRemoveSymbolHandler (mamaSymbolList *symbolList*, removeSymbolCbType *removeCb*)

Registers the user defined remove symbol callback with the symbolList.

The registered callback will get called each time a symbol is deleted from the symbol list.

Parameters:

← ***symbolList*** The symbolList.

← ***removeCb*** Pointer to the user defined callback. Must conform to function prototype `removeSymbolCbType`.

Returns:

mama_status return code can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.40 symbollistmember.h File Reference

```
#include "mama/mama.h"
#include "mama/symbollisttypes.h"
```

Functions

- MAMAExpDLL mama_status mamaSymbolListMember_allocate (mamaSymbolList symbolList, mamaSymbolListMember *member)

  This function will allocate a new symbol list member.

- MAMAExpDLL mama_status mamaSymbolListMember_deallocate (mamaSymbolListMember member)

  Frees a symbol list member previously created by a called to mamaSymbolListMember_allocate.

- MAMAExpDLL mama_status mamaSymbolListMember_getClosure (const mamaSymbolListMember member, void **closure)

  This function returns the closure assocated with the symbol list member.

- int mamaSymbolListMember_getSize (void)
- MAMAExpDLL mama_status mamaSymbolListMember_getSource (const mamaSymbolListMember member, const char **source)

  This function returns the source assocated with the symbol list member.

- MAMAExpDLL mama_status mamaSymbolListMember_getSymbol (const mamaSymbolListMember member, const char **symbol)

  This function returns the symbol assocated with the symbol list member.

- MAMAExpDLL mama_status mamaSymbolListMember_getSymbolList (const mamaSymbolListMember member, mamaSymbolList *symbolList)

  This function returns the symbol list assocated with the symbol list member.

- MAMAExpDLL mama_status mamaSymbolListMember_getTransport (const mamaSymbolListMember member, mamaTransport *transport)

  This function returns the transport assocated with the symbol list member.

- MAMAExpDLL mama_status mamaSymbolListMember_setClosure (mamaSymbolListMember member, void *closure)
- MAMAExpDLL mama_status mamaSymbolListMember_setSource (mamaSymbolListMember member, const char *source)
- MAMAExpDLL mama_status mamaSymbolListMember_setSymbol (mamaSymbolListMember member, const char *symbol)

- MAMAExpDLL mama_status mamaSymbolListMember_setSymbolList (mamaSymbolListMember member, mamaSymbolList symbolList)
- MAMAExpDLL mama_status mamaSymbolListMember_setTransport (mamaSymbolListMember member, mamaTransport transport)

5.40.1 Function Documentation

5.40.1.1 MAMAExpDLL mama_status mamaSymbolListMember_allocate (mamaSymbolList *symbolList*, mamaSymbolListMember ∗ *member*)

This function will allocate a new symbol list member.

The returned member should be freed using mamaSymbolListMember_deallocate.

Parameters:

← ***symbolList*** The parent symbol list that the member will become part of.

→ ***member*** To return the new member.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.40.1.2 MAMAExpDLL mama_status mamaSymbolListMember_deallocate (mamaSymbolListMember *member*)

Frees a symbol list member previously created by a called to mamaSymbolListMember_allocate.

Parameters:

← ***member*** The member to free.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.40.1.3 MAMAExpDLL mama_status mamaSymbolListMember_getClosure (const mamaSymbolListMember *member*, void ∗∗ *closure*)

This function returns the closure assocated with the symbol list member.

Parameters:

← ***member*** The member to free.

→ ***closure*** To return the closure.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_-STATUS_OK

5.40.1.4 int mamaSymbolListMember_getSize (void)

5.40.1.5 MAMAExpDLL mama_status mamaSymbolListMember_getSource (const mamaSymbolListMember *member*, const char ∗∗ *source*)

This function returns the source assocated with the symbol list member.

Parameters:

← ***member*** The member to free.

→ ***source*** To return the source.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_-STATUS_OK

5.40.1.6 MAMAExpDLL mama_status mamaSymbolListMember_getSymbol (const mamaSymbolListMember *member*, const char ∗∗ *symbol*)

This function returns the symbol assocated with the symbol list member.

Parameters:

← ***member*** The member to free.

→ ***symbol*** To return the symbol.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_-STATUS_OK

5.40.1.7 MAMAExpDLL mama_status mamaSymbolListMember_getSymbolList (const mamaSymbolListMember *member*, mamaSymbolList * *symbolList*)

This function returns the symbol list assocated with the symbol list member.

Parameters:

← ***member*** The member to free.

→ ***symbolList*** To return the symbol list.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.40.1.8 MAMAExpDLL mama_status mamaSymbolListMember_getTransport (const mamaSymbolListMember *member*, mamaTransport * *transport*)

This function returns the transport assocated with the symbol list member.

Parameters:

← ***member*** The member to free.

→ ***transport*** To return the transport.

Returns:

mama_status value can be one of MAMA_STATUS_NULL_ARG MAMA_STATUS_OK

5.40.1.9 MAMAExpDLL mama_status mamaSymbolListMember_setClosure (mamaSymbolListMember *member*, void * *closure*)

5.40.1.10 MAMAExpDLL mama_status mamaSymbolListMember_setSource (mamaSymbolListMember *member*, const char * *source*)

5.40.1.11 MAMAExpDLL mama_status mamaSymbolListMember_setSymbol (mamaSymbolListMember *member*, const char * *symbol*)

5.40.1.12 MAMAExpDLL mama_status mamaSymbolListMember_setSymbolList (mamaSymbolListMember *member*, mamaSymbolList *symbolList*)

5.40.1.13 MAMAExpDLL mama_status mamaSymbolListMember_setTransport (mamaSymbolListMember *member*, mamaTransport *transport*)

5.41 symbollisttypes.h File Reference

Typedefs

- typedef mamaSymbolList_ ∗ mamaSymbolList
- typedef mamaSymbolListMember_ ∗ mamaSymbolListMember

5.41.1 Typedef Documentation

5.41.1.1 typedef struct mamaSymbolList_∗ mamaSymbolList

5.41.1.2 typedef struct mamaSymbolListMember_∗ mamaSymbolListMember

5.42 symbolmap.h File Reference

```
#include <mama/types.h>
```

Typedefs

- typedef int(MAMACALLTYPE ∗) mamaSymbolMapFunc (void ∗closure, char ∗result, const char ∗symbol, size_t maxLen)

  Prototype for symbol-mapping function.

5.42.1 Typedef Documentation

5.42.1.1 typedef int(MAMACALLTYPE ∗) mamaSymbolMapFunc(void ∗closure, char ∗result, const char ∗symbol, size_t maxLen)

Prototype for symbol-mapping function.

Parameters:

closure Caller supplied closure.

result The resulting symbol after mapping.

symbol The symbol to be mapped.

maxLen The maximum length of the mapped symbol (i.e., "result"), including trailing NULL.

Returns:

Whether a mapping for the symbol was found.

5.43 symbolmapfile.h File Reference

```
#include "mama/mama.h"
```

Typedefs

- typedef void ∗ mamaSymbolMapFile

Functions

- MAMAExpDLL void mamaSymbolMapFile_addMap (mamaSymbolMapFile fileMap, const char ∗fromSymbol, const char ∗toSymbol)

  Add additional individual symbology mapping.

- MAMAExpDLL mama_status mamaSymbolMapFile_create (mamaSymbolMapFile ∗fileMap)

  Create a file-based symbol mapper.

- MAMAExpDLL mama_status mamaSymbolMapFile_destroy (mamaSymbolMapFile fileMap)

  Destroy the file-based symbol mapper.

- MAMAExpDLL mama_status mamaSymbolMapFile_load (mamaSymbolMapFile fileMap, const char ∗fileName)

  (Re)Load the map for the file-based symbol mapper.

- MAMAExpDLL int mamaSymbolMapFile_map (mamaSymbolMapFile fileMap, char ∗result, const char ∗symbol, size_t maxLen)

  Perform a mapping from the file-based symbol mapper.

- MAMAExpDLL int mamaSymbolMapFile_revMap (mamaSymbolMapFile fileMap, char ∗result, const char ∗symbol, size_t maxLen)

  Perform a reverse mapping from the file-based symbol mapper.

5.43.1 Typedef Documentation

5.43.1.1 typedef void* mamaSymbolMapFile

5.43.2 Function Documentation

5.43.2.1 MAMAExpDLL void mamaSymbolMapFile_addMap (mamaSymbolMapFile *fileMap*, const char * *fromSymbol*, const char * *toSymbol*)

Add additional individual symbology mapping.

Parameters:

fileMap Handle to the file-map to destroy.

fromSymbol The from symbol.

toSymbol The to symbol.

5.43.2.2 MAMAExpDLL mama_status mamaSymbolMapFile_create (mamaSymbolMapFile * *fileMap*)

Create a file-based symbol mapper.

Use mamaSymbolMapFile_load() to open the file. If the file exists, its contents should be a two columns of data, with the columns separated by white space. A matching symbol of the left column is mapped to the symbol in the right column. when performing the mapping, if the symbol does not match anything in the file (or the file cannot be found), then the original symbol is used (i.e., no mapping).

Parameters:

fileMap The initialized file-map handle.

Returns:

mama_status return value can be one of MAMA_STATUS_NOMEM MAMA_-STATUS_OK

5.43.2.3 MAMAExpDLL mama_status mamaSymbolMapFile_destroy (mamaSymbolMapFile *fileMap*)

Destroy the file-based symbol mapper.

Parameters:

fileMap Handle to the file-map to destroy.

Returns:

mama_status return value can be one of MAMA_STATUS_INVALID_ARG MAMA_STATUS_OK

5.43.2.4 MAMAExpDLL mama_status mamaSymbolMapFile_load (mamaSymbolMapFile *fileMap*, const char * *fileName*)

(Re)Load the map for the file-based symbol mapper.

Parameters:

fileMap Handle to the file-map to destroy.

fileName The name of the file containing symbols to map.

Returns:

mama_status return value can be one of MAMA_STATUS_INVALID_ARG MAMA_STATUS_OK

5.43.2.5 MAMAExpDLL int mamaSymbolMapFile_map (mamaSymbolMapFile *fileMap*, char * *result*, const char * *symbol*, size_t *maxLen*)

Perform a mapping from the file-based symbol mapper.

Parameters:

fileMap Handle to the file-map.

result Returned "toSymbol" (feed side).

symbol "fromSymbol" (client side).

maxLen Max buffer length of result.

Returns:

1 if a symbol could be mapped else 0.

5.43.2.6 MAMAExpDLL int mamaSymbolMapFile_revMap (mamaSymbolMapFile *fileMap*, char * *result*, const char * *symbol*, size_t *maxLen*)

Perform a reverse mapping from the file-based symbol mapper.

Parameters:

fileMap Handle to the file-map.

result Returned "fromSymbol" (client side).

symbol "toSymbol" (feed side).

maxLen Max buffer length of result.

5.44 timer.h File Reference

```
#include <mama/status.h>
#include <mama/types.h>
#include <mama/queue.h>
```

Typedefs

- typedef void(MAMACALLTYPE ∗) mamaTimerCb (mamaTimer timer, void ∗closure)

  Prototype for callback invoked by timer.

Functions

- MAMAExpDLL mama_status mamaTimer_create (mamaTimer ∗result, mamaQueue queue, mamaTimerCb action, mama_f64_t interval, void ∗closure)

  Create a repeating timer.

- MAMAExpDLL mama_status mamaTimer_create2 (mamaTimer ∗result, mamaQueue queue, mamaTimerCb action, mamaTimerCb onTimerDestroyed, mama_f64_t interval, void ∗closure)

  Create a repeating timer.

- MAMAExpDLL mama_status mamaTimer_allocate (mamaTimer ∗result, mamaQueue queue)

  Allocate a repeating timer.

- MAMAExpDLL mama_status mamaTimer_allocate2 (mamaTimer ∗result, mamaQueue queue, mamaTimerCb onTimerDestroyed)

  Allocate a repeating timer.

- MAMAExpDLL mama_status mamaTimer_start (mamaTimer result, mamaTimerCb action, mama_f64_t interval, void ∗closure)

  Start a repeating timer created using allocate.

- MAMAExpDLL mama_status mamaTimer_destroy (mamaTimer timer)

  Destroy the timer.

- MAMAExpDLL mama_status mamaTimer_reset (mamaTimer timer)

  Reset the timer to the beginning of the interval.

- MAMAExpDLL mama_status mamaTimer_setInterval (mamaTimer timer, mama_f64_t interval)

  Set the timer to use a different interval (and reset to the beginning of that interval).

- MAMAExpDLL mama_status mamaTimer_getInterval (const mamaTimer timer, mama_f64_t *interval)

  Get the current timer interval.

- MAMAExpDLL mama_status mamaTimer_getQueue (const mamaTimer timer, mamaQueue *queue)

  Return the `mamaQueue` *for this timer.*

5.44.1 Typedef Documentation

5.44.1.1 typedef void(MAMACALLTYPE *) mamaTimerCb(mamaTimer timer, void *closure)

Prototype for callback invoked by timer.

Parameters:

timer The timer handle.

closure Caller supplied closure.

5.44.2 Function Documentation

5.44.2.1 MAMAExpDLL mama_status mamaTimer_create (mamaTimer * *result*, mamaQueue *queue*, mamaTimerCb *action*, mama_f64_t *interval*, void * *closure*)

Create a repeating timer.

Since the mamaTimer relies on the timer mechanism of the underlying middleware, the resolution of the timer is also dependent on the middleware. Consult your middleware documentation for details.

The callback is invoked repeatedly at the specified interval until the timer is destroyed.

Parameters:

result A pointer to the timer handle.

queue The queue from which the timer event will be dispatched.

action The callback to be invoked after the interval.

closure The closure that is passed to the callback.

interval,: The interval in seconds.

5.44.2.2 MAMAExpDLL mama_status mamaTimer_create2 (mamaTimer * *result*, mamaQueue *queue*, mamaTimerCb *action*, mamaTimerCb *onTimerDestroyed*, mama_f64_t *interval*, void * *closure*)

Create a repeating timer.

Since the mamaTimer relies on the timer mechanism of the underlying middleware, the resolution of the timer is also dependent on the middleware. Consult your middleware documentation for details.

The callback is invoked repeatedly at the specified interval until the timer is destroyed.

Parameters:

result A pointer to the timer handle.

queue The queue from which the timer event will be dispatched.

action The callback to be invoked after the interval.

onTimerDestroyed This callback will be invoked whenever the timer is destroyed, can be NULL.

closure The closure that is passed to the callback.

interval,: The interval in seconds.

5.44.2.3 MAMAExpDLL mama_status mamaTimer_allocate (mamaTimer * *result*, mamaQueue *queue*)

Allocate a repeating timer.

Parameters:

result A pointer to the timer handle.

queue The queue from which the timer event will be dispatched.

5.44.2.4 MAMAExpDLL mama_status mamaTimer_allocate2 (mamaTimer * *result*, mamaQueue *queue*, mamaTimerCb *onTimerDestroyed*)

Allocate a repeating timer.

Parameters:

result A pointer to the timer handle.

queue The queue from which the timer event will be dispatched.

onTimerDestroyed Callback will be invoked whenever the timer has been completely destroyed.

5.44.2.5 MAMAExpDLL mama_status mamaTimer_start (mamaTimer *result*, mamaTimerCb *action*, mama_f64_t *interval*, void * *closure*)

Start a repeating timer created using allocate.

The callback is invoked repeatedly at the specified interval until the timer is destroyed.

Parameters:

result The timer handle returned from allocate.

queue The queue from which the timer event will be dispatched.

action The callback to be invoked after the interval.

closure The closure that is passed to the callback.

interval,: The interval in seconds.

5.44.2.6 MAMAExpDLL mama_status mamaTimer_destroy (mamaTimer *timer*)

Destroy the timer.

This function must be called from the same thread dispatching on the associated event queue unless both the default queue and dispatch queue are not actively dispatching. Note that this function is asynchronous and is only guaranteed to have finished whenever the onTimerDestroyed function passed to the mamaTimer_create2 has been called.

Parameters:

timer The mamaTimer to be destroyed.

5.44.2.7 MAMAExpDLL mama_status mamaTimer_reset (mamaTimer *timer*)

Reset the timer to the beginning of the interval.

Parameters:

timer The mamaTimer to be reset.

5.44.2.8 MAMAExpDLL mama_status mamaTimer_setInterval (mamaTimer *timer*, mama_f64_t *interval*)

Set the timer to use a different interval (and reset to the beginning of that interval).

Parameters:

timer The mamaTimer to change the interval.

interval The new interval for the timer.

5.44.2.9 MAMAExpDLL mama_status mamaTimer_getInterval (const mamaTimer *timer*, mama_f64_t * *interval*)

Get the current timer interval.

Parameters:

timer The mamaTimer.

interval Address of the location where the interval will be written.

5.44.2.10 MAMAExpDLL mama_status mamaTimer_getQueue (const mamaTimer *timer*, mamaQueue * *queue*)

Return the `mamaQueue` for this timer.

Parameters:

timer The timer.

queue A pointer to hold the queue.

5.45 timezone.h File Reference

```
#include <mama/config.h>
#include <mama/status.h>
#include <mama/types.h>
#include "wombat/port.h"
#include <time.h>
```

Functions

- MAMAExpDLL mamaTimeZone mamaTimeZone_local (void)

  Return a pointer to a mamaTimeZone corresponding to the local time zone.

- MAMAExpDLL mamaTimeZone mamaTimeZone_utc (void)

  Return a pointer to a mamaTimeZone corresponding to the UTC time zone.

- MAMAExpDLL mamaTimeZone mamaTimeZone_usEastern (void)

  Return a pointer to a mamaTimeZone corresponding to the US Eastern time zone.

- MAMAExpDLL mama_status mamaTimeZone_create (mamaTimeZone *timeZone)

  Create an instance of a mamaTimeZone.

- MAMAExpDLL mama_status mamaTimeZone_createFromTz (mamaTimeZone *timeZone, const char *tzId)

  Create an instance of a mamaTimeZone with a standard TZ identifier (NULL identifier is equivalent to the local time zone).

- MAMAExpDLL mama_status mamaTimeZone_createCopy (mamaTimeZone *timeZone, const mamaTimeZone timeZoneCopy)

  Create an instance of a mamaTimeZone by copying from another instance.

- MAMAExpDLL mama_status mamaTimeZone_destroy (mamaTimeZone timeZone)

  Destroy an instance of a mamaTimeZone.

- MAMAExpDLL mama_status mamaTimeZone_copy (mamaTimeZone timeZone, const mamaTimeZone timeZoneCopy)

  Copy an instance of a mamaTimeZone (to a previously-created instance).

- MAMAExpDLL mama_status mamaTimeZone_set (mamaTimeZone timeZone, const char *tzId)

  Set an existing instance of a mamaTimeZone to a new time zone.

- MAMAExpDLL mama_status mamaTimeZone_clear (mamaTimeZone timeZone)

  Clear the time zone (but not destroyed).

- MAMAExpDLL mama_status mamaTimeZone_getTz (const mamaTimeZone timeZone, const char **result)

  Get the time zone ID string.

- MAMAExpDLL mama_status mamaTimeZone_getOffset (const mamaTimeZone timeZone, mama_i32_t *result)

  Get the offset from UTC (in seconds).

- MAMAExpDLL mama_status mamaTimeZone_check (mamaTimeZone timeZone)

  Check (recalculate) the UTC offset in case it has changed due to daylight savings adjustments.

- MAMAExpDLL void mamaTimeZone_setScanningInterval (mama_f64_t seconds)

  Set the interval of the thread updating each MamaTimeZone instance offset.

5.45.1 Function Documentation

5.45.1.1 MAMAExpDLL mamaTimeZone mamaTimeZone_local (void)

Return a pointer to a mamaTimeZone corresponding to the local time zone.

5.45.1.2 MAMAExpDLL mamaTimeZone mamaTimeZone_utc (void)

Return a pointer to a mamaTimeZone corresponding to the UTC time zone.

5.45.1.3 MAMAExpDLL mamaTimeZone mamaTimeZone_usEastern (void)

Return a pointer to a mamaTimeZone corresponding to the US Eastern time zone.

5.45.1.4 MAMAExpDLL mama_status mamaTimeZone_create (mamaTimeZone * *timeZone*)

Create an instance of a mamaTimeZone.

5.45.1.5 MAMAExpDLL mama_status mamaTimeZone_createFromTz (mamaTimeZone * *timeZone*, const char * *tzId*)

Create an instance of a mamaTimeZone with a standard TZ identifier (NULL identifier is equivalent to the local time zone).

5.45.1.6 MAMAExpDLL mama_status mamaTimeZone_createCopy (mamaTimeZone * *timeZone*, const mamaTimeZone *timeZoneCopy*)

Create an instance of a mamaTimeZone by copying from another instance.

5.45.1.7 MAMAExpDLL mama_status mamaTimeZone_destroy (mamaTimeZone *timeZone*)

Destroy an instance of a mamaTimeZone.

5.45.1.8 MAMAExpDLL mama_status mamaTimeZone_copy (mamaTimeZone *timeZone*, const mamaTimeZone *timeZoneCopy*)

Copy an instance of a mamaTimeZone (to a previously-created instance).

5.45.1.9 MAMAExpDLL mama_status mamaTimeZone_set (mamaTimeZone *timeZone*, const char * *tzId*)

Set an existing instance of a mamaTimeZone to a new time zone.

5.45.1.10 MAMAExpDLL mama_status mamaTimeZone_clear (mamaTimeZone *timeZone*)

Clear the time zone (but not destroyed).

5.45.1.11 MAMAExpDLL mama_status mamaTimeZone_getTz (const mamaTimeZone *timeZone*, const char ** *result*)

Get the time zone ID string.

5.45.1.12 MAMAExpDLL mama_status mamaTimeZone_getOffset (const mamaTimeZone *timeZone*, mama_i32_t * *result*)

Get the offset from UTC (in seconds).

Can be positive or negative, depending upon the direction.

5.45.1.13 MAMAExpDLL mama_status mamaTimeZone_check (mamaTimeZone *timeZone*)

Check (recalculate) the UTC offset in case it has changed due to daylight savings adjustments.

5.45.1.14 MAMAExpDLL void mamaTimeZone_setScanningInterval (mama_f64_t *seconds*)

Set the interval of the thread updating each MamaTimeZone instance offset.

5.46 transport.h File Reference

```
#include "mama/types.h"
#include "mama/status.h"
#include "mama/subscription.h"
#include "mama/msg.h"
#include "mama/symbolmap.h"
#include "mama/quality.h"
```

Typedefs

- typedef void(MAMACALLTYPE ∗) mamaTransportCB (mamaTransport tport, mamaTransportEvent, short cause, const void ∗platformInfo, void ∗closure)

  Invoked when transport disconnects, reconnects, or has a data quality event.

- typedef void(MAMACALLTYPE ∗) mamaTransportTopicCB (mamaTransport tport, mamaTransportTopicEvent event, const char ∗topic, const void ∗platformInfo, void ∗closure)

  Invoked when topic is subscribed ot unsubcribed on that transport.

- typedef void(∗) mamaTransportLbInitialCB (int numTransports, int ∗transportIndex)
- typedef void(∗) mamaTransportLbCB (int curTransportIndex, int numTransports, const char ∗source, const char ∗symbol, int ∗nextTransportIndex)

Enumerations

- enum mamaTransportEvent {

  MAMA_TRANSPORT_CONNECT, MAMA_TRANSPORT_CONNECT_FAILED, MAMA_TRANSPORT_RECONNECT, MAMA_TRANSPORT_DISCONNECT,

  MAMA_TRANSPORT_ACCEPT, MAMA_TRANSPORT_ACCEPT_RECONNECT, MAMA_TRANSPORT_PUBLISHER_DISCONNECT, MAMA_TRANSPORT_QUALITY,

  MAMA_TRANSPORT_NAMING_SERVICE_CONNECT, MAMA_TRANSPORT_NAMING_SERVICE_DISCONNECT, MAMA_TRANSPORT_WRITE_QUEUE_HIGH_WATER_MARK, MAMA_TRANSPORT_WRITE_QUEUE_LOW_WATER_MARK }

  MAMA passes this enumeration as a parameter to the mamaTransportCB if a client installs one.

- enum mamaTransportTopicEvent { MAMA_TRANSPORT_TOPIC_SUBSCRIBED, MAMA_TRANSPORT_TOPIC_UNSUBSCRIBED }

  MAMA passes this enumeration as a parameter to the mamaTransportTopicCB if a client installs one.

- enum tportLbScheme { TPORT_LB_SCHEME_STATIC, TPORT_LB_SCHEME_ROUND_ROBIN, TPORT_LB_SCHEME_API, TPORT_LB_SCHEME_LIBRARY }

  Enum to represent the different load balancing schemes available.

- enum mamaThrottleInstance { MAMA_THROTTLE_DEFAULT = 0, MAMA_THROTTLE_INITIAL = 1, MAMA_THROTTLE_RECAP = 2 }

  Enum to determine to which throttle a call applies.

Functions

- MAMAExpDLL const char * mamaTransportEvent_toString (mamaTransportEvent event)

  Return a text description of the transport event.

- MAMAExpDLL mama_status mamaTransport_setTransportCallback (mamaTransport transport, mamaTransportCB callback, void *closure)

  Set the transport callback.

- MAMAExpDLL mama_status mamaTransport_setWriteQueueWatermarks (mamaTransport transport, uint32_t highWater, uint32_t lowWater)

  Set the transport write queue high and low water mark values.

- MAMAExpDLL mama_status mamaTransport_setTransportTopicCallback (mamaTransport transport, mamaTransportTopicCB callback, void *closure)

  Set the transport topic callback.

- MAMAExpDLL mama_status mamaTransport_allocate (mamaTransport *result)

  Allocate a transport structure.

- MAMAExpDLL mama_status mamaTransport_create (mamaTransport transport, const char *name, mamaBridge bridgeImpl)

  Create a previously allocated transport.

- MAMAExpDLL mama_status mamaTransport_setName (mamaTransport transport, const char *name)

  Set the transport name.

- MAMAExpDLL mama_status mamaTransport_getName (mamaTransport transport, const char **result)

  Get the transport name.

- MAMAExpDLL const char * mamaTransport_getMiddleware (mamaTransport transport)
- MAMAExpDLL mama_status mamaTransport_getOutboundThrottle (mamaTransport transport, mamaThrottleInstance instance, double *result)

  Get the outbound throttle rate.

- MAMAExpDLL void mamaTransport_disableRefresh (mamaTransport transport, uint8_t disable)

  Disable Refreshing.

- MAMAExpDLL mama_status mamaTransport_setOutboundThrottle (mamaTransport transport, mamaThrottleInstance instance, double outboundThrottle)

  Set the throttle rate.

- MAMAExpDLL void mamaTransport_setSymbolMapFunc (mamaTransport transport, mamaSymbolMapFunc mapFunc, void *closure)

  Set the symbol mapping function for a mamaTransport.

- MAMAExpDLL mamaSymbolMapFunc mamaTransport_getSymbolMapFunc (mamaTransport transport)

  Return the symbol mapping function for a mamaTransport.

- MAMAExpDLL void * mamaTransport_getSymbolMapFuncClosure (mamaTransport transport)

  Return the symbol mapping function closure for a mamaTransport.

- MAMAExpDLL mama_status mamaTransport_setDescription (mamaTransport transport, const char *description)

  Set the description for this transport.

- MAMAExpDLL mama_status mamaTransport_getDescription (mamaTransport transport, const char **description)

  Get the description attribute for the specified transport.

- MAMAExpDLL mama_status mamaTransport_setInvokeQualityForAllSubscs (mamaTransport transport, int invokeQualityForAllSubscs)

Set whether to invoke the quality callback for all subscriptions whenever a data quality event occurs (the default), or whether to invoke the quality callback only when data subsequently arrives for a subscription.

- MAMAExpDLL mama_status mamaTransport_getInvokeQualityForAllSubscs (const mamaTransport transport, int *invokeQualityForAllSubscs)

  Get whether the transport has been set to invoke the quality callback for all subscriptions whenever a data quality event occurs.

- MAMAExpDLL mama_status mamaTransport_getQuality (const mamaTransport transport, mamaQuality *quality)

  Get the quality of data for the transport.

- MAMAExpDLL mama_status mamaTransport_destroy (mamaTransport transport)

  Destroy the transport.

- MAMAExpDLL mama_status mamaTransport_findConnection (mamaTransport transport, mamaConnection *result, const char *ipAddress, uint16_t port)

  Find a connection with specified IP Address and Port.

- MAMAExpDLL mama_status mamaTransport_getAllConnections (mamaTransport transport, mamaConnection **result, uint32_t *len)

  Return a list of all clients connected to this transport.

- MAMAExpDLL mama_status mamaTransport_getAllConnectionsForTopic (mamaTransport transport, const char *topic, mamaConnection **result, uint32_t *len)

  Return a list of all clients connected to this transport for a topic.

- MAMAExpDLL mama_status mamaTransport_freeAllConnections (mamaTransport transport, mamaConnection *result, uint32_t len)

  Free resources allocated by mamaTransport_getAllConnections().

- MAMAExpDLL mama_status mamaTransport_getAllServerConnections (mamaTransport transport, mamaServerConnection **result, uint32_t *len)

  Return a list of all servers connected to this transport.

- MAMAExpDLL mama_status mamaTransport_freeAllServerConnections (mamaTransport transport, mamaServerConnection *result, uint32_t len)

  Free resources allocated by mamaTransport_getAllServerConnections().

- MAMAExpDLL mama_status mamaTransport_setLbInitialCallback (mamaTransport transport, mamaTransportLbInitialCB callback)

  Set the callback which will decide which member of the load balancing group to use initially.

- MAMAExpDLL mama_status mamaTransport_setLbCallback (mamaTransport transport, mamaTransportLbCB callback)

  Set the callback which will allocate the next member of the load balancing group to use.

- MAMAExpDLL mama_status mamaTransport_requestConflation (mamaTransport transport)

  Request that publishers conflate incoming messages.

- MAMAExpDLL mama_status mamaTransport_requestEndConflation (mamaTransport transport)

  Request that publishers stop conflating incoming messages for this transport.

- MAMAExpDLL mama_status mamaTransport_setDeactivateSubscriptionOnError (mamaTransport transport, int deactivate)
- MAMAExpDLL int mamaTransport_getDeactivateSubscriptionOnError (mamaTransport transport)
- MAMAExpDLL mama_status mamaTransport_getNativeTransport (mamaTransport transport, int index, void **nativeTport)

  Return a pointer tothe underlying native transport.

- MAMAExpDLL mama_status mamaTransport_getNativeTransportNamingCtx (mamaTransport transport, int index, void **nativeTportNamingCtx)

  Return a pointer to the underlying native naming context the transport is using.

- MAMAExpDLL mamaStatsCollector mamaTransport_getStatsCollector (mamaTransport transport)
- MAMAExpDLL mama_status mamaTransport_getClosure (mamaTransport transport, void **closure)

  Return the closure.

- MAMAExpDLL mama_status mamaTransport_setClosure (mamaTransport transport, void *closure)

  Set a closure.

5.46.1 Typedef Documentation

5.46.1.1 typedef void(MAMACALLTYPE ∗) mamaTransportCB(mamaTransport tport, mamaTransportEvent, short cause, const void ∗platformInfo, void ∗closure)

Invoked when transport disconnects, reconnects, or has a data quality event.

Parameters:

tport The transport associated with the transport event

mamaTransportEvent The transport event

cause The cause of the transport event

platformInfo Info associated with the transport event

closure The closure argument to pass to the callback whenever it is invoked.

The cause and platformInfo are supplied only by some middlewares. The information provided by platformInfo is middleware specific. The following middlewares are supported:

tibrv: provides the char∗ version of the tibrv advisory message. wmw: provides a pointer to a mamaConnection struct for the event

5.46.1.2 typedef void(MAMACALLTYPE ∗) mamaTransportTopicCB(mamaTransport tport, mamaTransportTopicEvent event, const char ∗topic, const void ∗platformInfo, void ∗closure)

Invoked when topic is subscribed ot unsubcribed on that transport.

Parameters:

tport The transport associated with the transport topic event

mamaTransportTopicEvent The transport topic event

topic The topic being subscribed or unsubscribed to

platformInfo Info associated with the transport topicevent

closure The closure argument to pass to the callback whenever it is invoked.

The platformInfo is supplied only by some middlewares. The information provided by platformInfo is middleware specific. The following middlewares are supported:

wmw: provides a pointer to a mamaConnection struct for the event

5.46.1.3 typedef void(∗) mamaTransportLbInitialCB(int numTransports, int ∗transportIndex)

5.46.1.4 typedef void(∗) mamaTransportLbCB(int curTransportIndex, int numTransports, const char ∗source, const char ∗symbol, int ∗nextTransportIndex)

5.46.2 Enumeration Type Documentation

5.46.2.1 enum mamaTransportEvent

MAMA passes this enumeration as a parameter to the mamaTransportCB if a client installs one.

The values passed depend on the underlying middleware:

All Middleware: MAMA_TRANSPORT_QUALITY is passed when an the underlying transport detects an event that may affect data quality. Depending on the event, MAMA may also mark subscriptions as STALE or MAYBE_STALE.

TIBRV: TIBRV invokes the callback with MAMA_TRANSPORT_DISCONNECT when the transport disconnects from the RV daemon and MAMA_TRANSPORT_-RECONNECT when it reestablishes a connection to the daemon.

Wombat Middleware: For Wombat Middleware MAMA distinguishes between connections established from a transport to another transport and connections established by the transport from another transport. In general MAMA applications establish connection from a MAMA transport to a feed handler (publisher); however, they may also accept connections from other MAMA clients. Applications subscribe to topics and publisher messages over all connections. When a transport successfully connects to another transport (i.e. a feed handler) the mamaTransport-Callback receives MAMA_TRANSPORT_CONNECT if the connection succeeds immediately otherwise it receives a MAMA_TRANSPORT_RECONNECT if it subsequently succeeds. Wombat Middleware transports may be configured to retry failed connections. When a connection that a transport establishes to another transport (the publisher by convention) and the connection is lost the mamaTranpsortCallback receives the MAMA_TRANSPORT_PUBLISHER_DISCONNECT event. When a transport accepts a connection from a remote transport (the subscriber by convention) it receives either a MAMA_TRANSPORT_ACCEPT or MAMA_TRANSPORT_-ACCEPT_RECONNECT event indicating that the remote client is connecting or reconnecting. When the client disconnects MAMA passes the MAMA_TRANSPORT_-DISCONNECT value.

Enumerator:

MAMA_TRANSPORT_CONNECT

MAMA_TRANSPORT_CONNECT_FAILED

MAMA_TRANSPORT_RECONNECT

MAMA_TRANSPORT_DISCONNECT

MAMA_TRANSPORT_ACCEPT

MAMA_TRANSPORT_ACCEPT_RECONNECT

MAMA_TRANSPORT_PUBLISHER_DISCONNECT

MAMA_TRANSPORT_QUALITY

MAMA_TRANSPORT_NAMING_SERVICE_CONNECT

MAMA_TRANSPORT_NAMING_SERVICE_DISCONNECT

MAMA_TRANSPORT_WRITE_QUEUE_HIGH_WATER_MARK

MAMA_TRANSPORT_WRITE_QUEUE_LOW_WATER_MARK

```
{
    MAMA_TRANSPORT_CONNECT,
    MAMA_TRANSPORT_CONNECT_FAILED,
    MAMA_TRANSPORT_RECONNECT,
    MAMA_TRANSPORT_DISCONNECT,
    MAMA_TRANSPORT_ACCEPT,
    MAMA_TRANSPORT_ACCEPT_RECONNECT,
    MAMA_TRANSPORT_PUBLISHER_DISCONNECT,
    MAMA_TRANSPORT_QUALITY,
    MAMA_TRANSPORT_NAMING_SERVICE_CONNECT,
    MAMA_TRANSPORT_NAMING_SERVICE_DISCONNECT,
    MAMA_TRANSPORT_WRITE_QUEUE_HIGH_WATER_MARK,
    MAMA_TRANSPORT_WRITE_QUEUE_LOW_WATER_MARK
} mamaTransportEvent;
```

5.46.2.2 enum mamaTransportTopicEvent

MAMA passes this enumeration as a parameter to the mamaTransportTopicCB if a client installs one.

Wombat Middleware: For connection based transports an event is invoked when a client makes a subscription to a topic on a transport or unsubscribes from a topic on a transport.

Enumerator:

MAMA_TRANSPORT_TOPIC_SUBSCRIBED

MAMA_TRANSPORT_TOPIC_UNSUBSCRIBED

```
{
    MAMA_TRANSPORT_TOPIC_SUBSCRIBED,
    MAMA_TRANSPORT_TOPIC_UNSUBSCRIBED
} mamaTransportTopicEvent;
```

5.46.2.3 enum tportLbScheme

Enum to represent the different load balancing schemes available.

TPORT_LB_SCHEME_STATIC - keep the same lb sub tport for each subscription. TPORT_LB_SCHEME_ROUND_ROBIN - round robin between the members of the lb group for each subscription TPORT_LB_SCHEME_API - let the user defined load balancing callback(s) decide. TPORT_LB_SCHEME_LIBRARY - use the load balancing in a external library

Enumerator:

TPORT_LB_SCHEME_STATIC

TPORT_LB_SCHEME_ROUND_ROBIN

TPORT_LB_SCHEME_API

TPORT_LB_SCHEME_LIBRARY

```
{
    TPORT_LB_SCHEME_STATIC,
    TPORT_LB_SCHEME_ROUND_ROBIN,
    TPORT_LB_SCHEME_API,
    TPORT_LB_SCHEME_LIBRARY
} tportLbScheme;
```

5.46.2.4 enum mamaThrottleInstance

Enum to determine to which throttle a call applies.

Currently, the the default throttle, used by the publisher, and the initial value request throttle are the same. Mama sends recap requests on a separate throttle.

Enumerator:

MAMA_THROTTLE_DEFAULT

MAMA_THROTTLE_INITIAL

MAMA_THROTTLE_RECAP

```
{
    MAMA_THROTTLE_DEFAULT = 0,
    MAMA_THROTTLE_INITIAL = 1,
    MAMA_THROTTLE_RECAP   = 2
} mamaThrottleInstance;
```

5.46.3 Function Documentation

5.46.3.1 MAMAExpDLL const char* mamaTransportEvent_toString (mamaTransportEvent *event*)

Return a text description of the transport event.

5.46.3.2 MAMAExpDLL mama_status mamaTransport_setTransportCallback (mamaTransport *transport*, mamaTransportCB *callback*, void * *closure*)

Set the transport callback.

It receives advisories when a transport disconnects or reconnects. Passing NULL removes the callback.

5.46.3.3 MAMAExpDLL mama_status mamaTransport_setWriteQueueWatermarks (mamaTransport *transport*, uint32_t *highWater*, uint32_t *lowWater*)

Set the transport write queue high and low water mark values.

The MAMA_TRANSPORT_WRITE_QUEUE_HIGH_WATER_MARK and MAMA_TRANSPORT_WRITE_QUEUE_HIGH_WATER_MARK events will be delivered though the transport callback when the respective number of items are outstanding on a clients write queue.

5.46.3.4 MAMAExpDLL mama_status mamaTransport_setTransportTopicCallback (mamaTransport *transport*, mamaTransportTopicCB *callback*, void * *closure*)

Set the transport topic callback.

It receives advisories when a client subscribes or unsubscribes to a topic on the transport

5.46.3.5 MAMAExpDLL mama_status mamaTransport_allocate (mamaTransport * *result*)

Allocate a transport structure.

Do not free this memory, use mamaTransport_destroy() instead. The transport is not created until mamaTransport_create() is called. Any transport properties should be set after calling allocate() and prior to calling create()

Parameters:

result The address to where the transport will be written.

5.46.3.6 MAMAExpDLL mama_status mamaTransport_create (mamaTransport *transport*, const char * *name*, mamaBridge *bridgeImpl*)

Create a previously allocated transport.

Platform specific parameters are read from the properties file *mama.properties*. The properties file is located in the WOMBAT_PATH directory. The parameters are dependent on the underlying messaging transport.

TIBRV: transports support the following: mama.tibrv.transport.name.service, mama.tibrv.transport.name.network, and mama.tibrv.transport.name.daemon. These correspond to the parameters for tibrvTransport_Create().

LBM: See the example mama.properties supplied with the release.

Parameters:

transport The previously allocated transport.

name The name of the transport in the mama.properties file.

bridgeImpl The middleware for which the transport is being created.

5.46.3.7 MAMAExpDLL mama_status mamaTransport_setName (mamaTransport *transport*, const char * *name*)

Set the transport name.

This can be used to set the name without calling mamaTransport_create()

Parameters:

transport The transport from which the name is being obtained.

name The name of the transport in the mama.properties file.

5.46.3.8 MAMAExpDLL mama_status mamaTransport_getName (mamaTransport *transport*, const char ** *result*)

Get the transport name.

This the name that was passed to the mamaTransport_create() or mamaTransport_setName() function.

Parameters:

transport The transport from which the name is being obtained.

result Pointer to the transport name string.

5.46.3.9 MAMAExpDLL const char* mamaTransport_getMiddleware (mamaTransport *transport*)

5.46.3.10 MAMAExpDLL mama_status mamaTransport_getOutbound-Throttle (mamaTransport *transport*, mamaThrottleInstance *instance*, double * *result*)

Get the outbound throttle rate.

This the rate at which the transport sends outbound messages to the feed handlers. It is also the rate at which new subscriptions are created. Its purpose is to avoid flooding the network with requests.

Parameters:

transport the transport.

instance the instance

result Pointer to the resulting value in messages/second.

5.46.3.11 MAMAExpDLL void mamaTransport_disableRefresh (mamaTransport *transport*, uint8_t *disable*)

Disable Refreshing.

Parameters:

transport the transport instance

disable t/f.

5.46.3.12 MAMAExpDLL mama_status mamaTransport_setOutbound-Throttle (mamaTransport *transport*, mamaThrottleInstance *instance*, double *outboundThrottle*)

Set the throttle rate.

Parameters:

transport the transport.

instance the instance

outboundThrottle the rate in messages/second.

5.46.3.13 MAMAExpDLL void mamaTransport_setSymbolMapFunc (mamaTransport *transport*, mamaSymbolMapFunc *mapFunc*, void * *closure*)

Set the symbol mapping function for a mamaTransport.

Parameters:

transport The transport.

mapFunc The symbol mapping function.

closure The closure argument to pass to mappingFunc whenever it is invoked.

5.46.3.14 MAMAExpDLL mamaSymbolMapFunc mamaTransport_getSymbolMapFunc (mamaTransport *transport*)

Return the symbol mapping function for a mamaTransport.

Parameters:

transport The transport.

Returns:

The symbol mapping function (could be NULL).

5.46.3.15 MAMAExpDLL void* mamaTransport_getSymbolMapFuncClosure (mamaTransport *transport*)

Return the symbol mapping function closure for a mamaTransport.

Parameters:

transport The transport.

Returns:

The symbol mapping function closure (could be NULL).

5.46.3.16 MAMAExpDLL mama_status mamaTransport_setDescription (mamaTransport *transport*, const char * *description*)

Set the description for this transport.

If the underlying middleware supports the concept of a description the values will be passed on. Otherwise MAMA will maintain the description internally.

Parameters:

transport The transport.

description The description for the transport. This will be copied.

5.46.3.17 MAMAExpDLL mama_status mamaTransport_getDescription (mamaTransport *transport*, const char ** *description*)

Get the description attribute for the specified transport.

Note that a copy of the description is not returned. Do not change or free this memory.

Parameters:

transport The transport

description The address to where the description will be written.

5.46.3.18 MAMAExpDLL mama_status mamaTransport_setInvokeQualityForAllSubscs (mamaTransport *transport*, int *invokeQualityForAllSubscs*)

Set whether to invoke the quality callback for all subscriptions whenever a data quality event occurs (the default), or whether to invoke the quality callback only when data subsequently arrives for a subscription.

If set to true, an onQuality callback will be invoked for a subscription whenever a data quality event occurs on the transport, even in between updates for that description. If set to false, the onQuality callback will not be called when the data quality event occurs on the transport. However, it will still be invoked when an update arrives for the subscription.

Parameters:

transport The transport

invokeQualityForAllSubscs Whether to invoke quality callback for all subscriptions

5.46.3.19 MAMAExpDLL mama_status mamaTransport_getInvokeQualityForAllSubscs (const mamaTransport *transport*, int * *invokeQualityForAllSubscs*)

Get whether the transport has been set to invoke the quality callback for all subscriptions whenever a data quality event occurs.

Parameters:

transport The transport

invokeQualityForAllSubscs Whether transport has been set to invoke quality callback for all subscriptions

5.46.3.20 MAMAExpDLL mama_status mamaTransport_getQuality (const mamaTransport *transport*, mamaQuality * *quality*)

Get the quality of data for the transport.

Currently only implemented for the Tibco RV middleware. Other middlewares always return STATUS_OK

Parameters:

transport The transport

quality The quality of data for the transport

5.46.3.21 MAMAExpDLL mama_status mamaTransport_destroy (mamaTransport *transport*)

Destroy the transport.

Parameters:

transport the transport.

5.46.3.22 MAMAExpDLL mama_status mamaTransport_findConnection (mamaTransport *transport*, mamaConnection * *result*, const char * *ipAddress*, uint16_t *port*)

Find a connection with specified IP Address and Port.

If the port is 0, the call returns the first connection with the specified IP Address. If a connection is not found the method returns MAMA_STATUS_NOT_FOUND and *result == NULL. The caller is responsible for calling mamaConnection_free().

For middleware that does not provide access to per-connection information (non WMW middleware), the method returns MAMA_STATUS_NOT_IMPL.

5.46.3.23 MAMAExpDLL mama_status mamaTransport_getAllConnections (mamaTransport *transport*, mamaConnection ∗∗ *result*, uint32_t ∗ *len*)

Return a list of all clients connected to this transport.

The result and len must be passed to mamaTransport_freeAllConnections() to free resources allocated by this function.

5.46.3.24 MAMAExpDLL mama_status mamaTransport_getAllConnections-ForTopic (mamaTransport *transport*, const char ∗ *topic*, mamaConnection ∗∗ *result*, uint32_t ∗ *len*)

Return a list of all clients connected to this transport for a topic.

The result and len must be passed to mamaTransport_freeAllConnections() to free resources allocated by this function.

5.46.3.25 MAMAExpDLL mama_status mamaTransport_freeAllConnections (mamaTransport *transport*, mamaConnection ∗ *result*, uint32_t *len*)

Free resources allocated by mamaTransport_getAllConnections().

5.46.3.26 MAMAExpDLL mama_status mamaTransport_getAllServer-Connections (mamaTransport *transport*, mamaServerConnection ∗∗ *result*, uint32_t ∗ *len*)

Return a list of all servers connected to this transport.

The result and len must be passed to mamaTransport_freeAllServerConnections() to free resources allocated by this function.

5.46.3.27 MAMAExpDLL mama_status mamaTransport_freeAllServer-Connections (mamaTransport *transport*, mamaServerConnection ∗ *result*, uint32_t *len*)

Free resources allocated by mamaTransport_getAllServerConnections().

5.46.3.28 MAMAExpDLL mama_status mamaTransport_setLbInitialCallback (mamaTransport *transport*, mamaTransportLbInitialCB *callback*)

Set the callback which will decide which member of the load balancing group to use initially.

If no callback is set then one will be chosen at random.

5.46.3.29 MAMAExpDLL mama_status mamaTransport_setLbCallback (mamaTransport *transport*, mamaTransportLbCB *callback*)

Set the callback which will allocate the next member of the load balancing group to use.

The callback, if set, will be invoked for each new subscription. If no callback is set then the currently selected member the same one will be used for all subscriptions

5.46.3.30 MAMAExpDLL mama_status mamaTransport_requestConflation (mamaTransport *transport*)

Request that publishers conflate incoming messages.

Publishers that support conflation (currently only Wombat TCP middleware) will enable conflation for this transport if possible.

5.46.3.31 MAMAExpDLL mama_status mamaTransport_requestEndConflation (mamaTransport *transport*)

Request that publishers stop conflating incoming messages for this transport.

Publishers that support conflation (currently only Wombat TCP middleware) will stop conflating messages if possible. Note that the publisher may continue to send conflated messages at its discretion.

5.46.3.32 MAMAExpDLL mama_status mamaTransport_setDeactivateSubscriptionOnError (mamaTransport *transport*, int *deactivate*)

5.46.3.33 MAMAExpDLL int mamaTransport_getDeactivateSubscriptionOnError (mamaTransport *transport*)

5.46.3.34 MAMAExpDLL mama_status mamaTransport_getNativeTransport (mamaTransport *transport*, int *index*, void ** *nativeTport*)

Return a pointer tothe underlying native transport.

Applications should avoid this method if possible as it may result in non-portable, middleware specific code. Callers must cast the nativeTport to the appropriate type (ie. tibrvTransport*).

Only implemented for TIBRV.

5.46.3.35 MAMAExpDLL mama_status mamaTransport_getNativeTransportNamingCtx (mamaTransport *transport*, int *index*, void ** *nativeTportNamingCtx*)

Return a pointer to the underlying native naming context the transport is using.

Applications should avoid this method if possible as it may result in non-portable, middleware specific code. Callers must cast the native naming context to the appropriate type (ie. wmwns*).

Only implemented for WMW version 5 and above.

5.46.3.36 MAMAExpDLL mamaStatsCollector mamaTransport_getStatsCollector (mamaTransport *transport*)

5.46.3.37 MAMAExpDLL mama_status mamaTransport_getClosure (mamaTransport *transport*, void ** *closure*)

Return the closure.

Parameters:

transport The transport.

closure holds the result.

5.46.3.38 MAMAExpDLL mama_status mamaTransport_setClosure (mamaTransport *transport*, void * *closure*)

Set a closure.

Parameters:

transport The transport.

closure holds the closure to be added.

5.47 types.h File Reference

```
#include "wombat/port.h"
#include <stdlib.h>
#include <string.h>
```

Defines

- #define MAMA_QUANTITY_EPSILON ((mama_f64_t)0.00000000001)
- #define mama_isQuantityNone(q) ((q) < MAMA_QUANTITY_EPSILON)

  Macro to determine if a quantity is zero.

- #define mama_isQuantityEqual(lhs, rhs)

  Macro to determine if a quantity equality.

Typedefs

- typedef int8_t mama_i8_t
- typedef uint8_t mama_u8_t
- typedef int16_t mama_i16_t
- typedef uint16_t mama_u16_t
- typedef int32_t mama_i32_t
- typedef uint32_t mama_u32_t
- typedef int64_t mama_i64_t
- typedef uint64_t mama_u64_t
- typedef int8_t mama_bool_t
- typedef float mama_f32_t
- typedef double mama_f64_t
- typedef double mama_quantity_t
- typedef uint16_t mama_fid_t
- typedef size_t mama_size_t
- typedef uint32_t mama_seqnum_t
- typedef mamaBridgeImpl_ * mamaBridge
- typedef mamaPayloadBridgeImpl_ * mamaPayloadBridge
- typedef mama_u64_t * mamaDateTime

  Flexible date/time format.

- typedef void * mamaTimeZone

  Time zone utility type.

- typedef void * mamaPrice

  Flexible price format (contains display hints as well as value).

- typedef mamaMsgImpl_ * mamaMsg

  Opaque definitions for all major MAMA data types.

- typedef mamaMsgFieldImpl_ * mamaMsgField
- typedef mamaMsgIteratorImpl_ * mamaMsgIterator
- typedef mamaTransportImpl_ * mamaTransport
- typedef mamaSubscriptionImpl_ * mamaSubscription
- typedef mamaManagedSubscriptionImpl_ * mamaManagedSubscription
- typedef mamaSubscriptionManagerImpl_ * mamaSubscriptionManager
- typedef mamaDictionaryImpl_ * mamaDictionary
- typedef mamaPublisherImpl_ * mamaPublisher
- typedef mamaDQPublisherImpl_ * mamaDQPublisher
- typedef mamaDQPublisherManagerImpl_ * mamaDQPublisherManager
- typedef mamaInboxImpl_ * mamaInbox
- typedef mamaQueueImpl_ * mamaQueue
- typedef mamaDispatcherImpl_ * mamaDispatcher
- typedef mamaTimerImpl * mamaTimer
- typedef mamaFieldDescriptorImpl_ * mamaFieldDescriptor
- typedef mamaIoImpl * mamaIo
- typedef mamaSymbolSourceFileImpl_ * mamaSymbolSourceFile
- typedef mamaSymbolStoreImpl_ * mamaSymbolStore
- typedef mamaMsgQualImpl_ * mamaMsgQual
- typedef mamaSourceManagerImpl_ * mamaSourceManager
- typedef mamaSourceImpl_ * mamaSource
- typedef mamaSymbologyImpl_ * mamaSymbology
- typedef mamaSymbologyManagerImpl_ * mamaSymbologyManager
- typedef mamaSourceGroupCbHandleImpl_ * mamaSourceGroupCbHandle
- typedef mamaConnectionImpl_ * mamaConnection
- typedef mamaServerConnectionImpl_ * mamaServerConnection
- typedef mamaConflationManager_ * mamaConflationManager
- typedef mamaStatsLoggerImpl_ * mamaStatsLogger
- typedef mamaStatImpl_ * mamaStat
- typedef mamaStatsCollectorImpl_ * mamaStatsCollector
- typedef mamaStatsGeneratorImpl_ * mamaStatsGenerator
- typedef mamaMsgReplyImpl_ * mamaMsgReply

5.47.1 Define Documentation

5.47.1.1 #define MAMA_QUANTITY_EPSILON ((mama_f64_t)0.00000000001)

5.47.1.2 #define mama_isQuantityNone(q) ((q) < MAMA_QUANTITY_EPSILON)

Macro to determine if a quantity is zero.

5.47.1.3 #define mama_isQuantityEqual(lhs, rhs)

Value:

```
((lhs) < (rhs) ? (rhs) - (lhs) < MAMA_QUANTITY_EPSILON : \
                         (lhs) - (rhs) < MAMA_QUANTITY_EPSILON)
```

Macro to determine if a quantity equality.

5.47.2 Typedef Documentation

5.47.2.1 typedef int8_t mama_i8_t

5.47.2.2 typedef uint8_t mama_u8_t

5.47.2.3 typedef int16_t mama_i16_t

5.47.2.4 typedef uint16_t mama_u16_t

5.47.2.5 typedef int32_t mama_i32_t

5.47.2.6 typedef uint32_t mama_u32_t

5.47.2.7 typedef int64_t mama_i64_t

5.47.2.8 typedef uint64_t mama_u64_t

5.47.2.9 typedef int8_t mama_bool_t

5.47.2.10 typedef float mama_f32_t

5.47.2.11 typedef double mama_f64_t

5.47.2.12 typedef double mama_quantity_t

5.47.2.13 typedef uint16_t mama_fid_t

5.47.2.14 typedef size_t mama_size_t

5.47.2.15 typedef uint32_t mama_seqnum_t

5.47.2.16 typedef struct mamaBridgeImpl_* mamaBridge

5.47.2.17 typedef struct mamaPayloadBridgeImpl_* mamaPayloadBridge

5.47.2.18 typedef mama_u64_t* mamaDateTime

Flexible date/time format.

5.47.2.19 typedef void* mamaTimeZone

Time zone utility type.

5.47.2.20 typedef void* mamaPrice

Flexible price format (contains display hints as well as value).

5.47.2.21 typedef struct mamaMsgImpl_* mamaMsg

Opaque definitions for all major MAMA data types.

Must be allocated through corresponding functions from the API.

5.47.2.22 typedef struct mamaMsgFieldImpl_* mamaMsgField

5.47.2.23 typedef struct mamaMsgIteratorImpl_* mamaMsgIterator

5.47.2.24 typedef struct mamaTransportImpl_* mamaTransport

5.47.2.25 typedef struct mamaSubscriptionImpl_* mamaSubscription

5.47.2.26 typedef struct mamaManagedSubscriptionImpl_* mamaManagedSubscription

5.47.2.27 typedef struct mamaSubscriptionManagerImpl_* mamaSubscriptionManager

5.47.2.28 typedef struct mamaDictionaryImpl_* mamaDictionary

5.47.2.29 typedef struct mamaPublisherImpl_* mamaPublisher

5.47.2.30 typedef struct mamaDQPublisherImpl_* mamaDQPublisher

5.47.2.31 typedef struct mamaDQPublisherManagerImpl_* mamaDQPublisherManager

5.47.2.32 typedef struct mamaInboxImpl_* mamaInbox

5.47.2.33 typedef struct mamaQueueImpl_* mamaQueue

5.47.2.34 typedef struct mamaDispatcherImpl_* mamaDispatcher

5.47.2.35 typedef struct mamaTimerImpl* mamaTimer

5.47.2.36 typedef struct mamaFieldDescriptorImpl_* mamaFieldDescriptor

5.47.2.37 typedef struct mamaIoImpl* mamaIo

5.47.2.38 typedef struct mamaSymbolSourceFileImpl_* mamaSymbolSourceFile

5.47.2.39 typedef struct mamaSymbolStoreImpl_* mamaSymbolStore

5.47.2.40 typedef struct mamaMsgQualImpl_* mamaMsgQual

5.47.2.41 typedef struct mamaSourceManagerImpl_* mamaSourceManager

5.47.2.42 typedef struct mamaSourceImpl_* mamaSource

5.47.2.43 typedef struct mamaSymbologyImpl_* mamaSymbology

5.47.2.44 typedef struct mamaSymbologyManagerImpl_* mamaSymbologyManager

5.47.2.45 typedef struct mamaSourceGroupCbHandleImpl_* mamaSourceGroupCbHandle

5.47.2.46 typedef struct mamaConnectionImpl_* mamaConnection

5.48 version.h File Reference

Defines

- #define MAMA_VERSION_MAJOR 2
- #define MAMA_VERSION_MINOR 2
- #define MAMA_VERSION_RELEASE 1.1
- #define MAMA_VERSION "openmama DEVRELEASE2.2.1.1"

Variables

- const char * mama_version

5.48.1 Define Documentation

5.48.1.1 #define MAMA_VERSION_MAJOR 2

5.48.1.2 #define MAMA_VERSION_MINOR 2

5.48.1.3 #define MAMA_VERSION_RELEASE 1.1

5.48.1.4 #define MAMA_VERSION "openmama DEVRELEASE2.2.1.1"

5.48.2 Variable Documentation

5.48.2.1 const char* mama_version