Configuring the Tick42 RMDS Bridge

1. Middlewares and transports

OpenMAMA is designed to be able to connect to any market data platform (or middleware) for which a bridge is available. The bridge presents client programs with the OpenMAMA API and translates between them and the native API for the platform.

The terms "bridge" and "middleware" can be used more or less interchangeably - although it is possible that more than one 3rd party vendor (such as Tick42) might produce a bridge for a particular platform (such as RMDS). In this document we will use "bridge" although in MAMA documentation it is often referred to as a "middleware bridge".

A bridge will make one or more connections to the middleware. Exactly what is meant by "*connecting to the middleware*" depends entirely on the middleware. In the case of the RMDS bridge the connection is a TCP socket connection. Each of these connections is referred to as a "transport".

The content of the mama.properties file does not control which bridge(s) get loaded or which transport(s) get created. That is entirely the responsibility of the client program and its configuration. In order to configure the bridge(s) and transport(s), there must be an appropriate set of entries in the mama.properties file.

This allows a single mama.properties file to be used by multiple applications, each of which selects the bridge and transport it will create and the names are matched with the properties file.

When a program creates a transport it specifies a name and this used to form the key (as described above) to obtain its properties

Elsewhere in this guide there will be configuration elements that name which bridges to load and which transports to create. For example RttXL configuration requires specification of bridges and transport names.

2. Mama Configuration

The MAMA components are configured from a properties file. This will configure both bridge specific properties and global properties

Sources for mama properties

The default set of property values is read from a file called "mama.properties" from the directory defined by the environment variable WOMBAT_PATH, or if that environment variable is not defined, from the current directory. Because the latter can be dependent on the host application, it is normally advisable to ensure that the environment variable is defined.

Further, a MAMA application can optionally load another properties file. The contents of this file will be merged with the default file. Properties with the same name will replace those read from the default file.

A properties file consists of a set of key=value pairs. Any line beginning with a '#' is treated as a comment and ignored. The keys are simply case-sensitive text strings and the values are also strings. If a key appears in the file more than once then the second or subsequent value overwrites any earlier values in the file.

There is a convention for forming keys.

Convention for forming property keys

The convention can be represented as

mama.{{<bridge-name>}.transport.<transport-name>}.<property-name>

This allows the definition of

1. global properties - *mama.<property-name>*
2. properties for an entire bridge - *mama.<bridge-name>.<property-name>*
3. properties for a specific transport created by a bridge - *mama.<bridge-name>.transport.<transport-name>.<property-name>*

In the Tick42 RMDS bridge the convention is extended where necessary to

mama.<bridge-name>.transport.<transport-name>.<source-name>.<property-name>

This allows source specific properties to be specified

Path values and environment variables

On WIndows operating systems where path separators use a backslash '\' character it must be escaped to "\\" . On Linux systems this is not an issue and a forward slash '/' is used as normal

On all platforms environment variables are expressed as $(ENV)

So, for example.....

```
mama.logging.file.name=$(AppData)\\Tick42\\RttXL\\Logs\\mama.log
```

Also, note that there is a limitation in the OpenMAMA infrastructure that only the first environment variable in any property value is expanded

This applies to bridge configuration as well as to global mama configurations

3. Global configuration

3.1. Mama logging

These keys are formed at the global mama level, for example

```
mama.logging.level=finest
mama.logging.file.name=$(AppData)\\Tick42\\RttXL\\Logs\\mama.log
mama.statslogging.enable=false
```

4. RMDS bridge configuration

4.1. Bridge properties

These properties are formed from *mama.tick42rmds.<property-name>* and apply to all transports created by the bridge.

property	type	default	description
statslogger.logfile	string	"stats.csv"	Logfile name for bridge statistics logging
statslogger.interval	string	"10"	statistics logging interval - set to 0 to disable statistics logging
statslogger.maxAgeDays	string	"5"	Period after which the statistics logging file is recycled
consolelogging	bool	false	All mama log file output is echoed to the console. This is useful when debugging using console applications

Examples

```
mama.tick42rmds.statslogger.logfile=${APPDATA}\\Tick42\\RttXL\\Logs\\RmdsStats.log
mama.tick42rmds.consolelogging=false
```

4.2. Transport Properties

These properties are formed from *mama.tick42rmds.transport.<transport-name>.<property-name>*

Transport names

In all the examples in this section it is assumed the transport names will be *sub* and *pub*. This is completely arbitrary.

4.2.1. Connection Properties

These settings are concerned with connecting to the RMDS

property	type	default value	description
hosts	string	none	comma separated list of hostname:port pairs that is used for a subscriber to connect to an RMDS. If no value is specified for this property then no connection to the RMDS will be made.
retrysched	string	"0,3(3),10(3),30(6)"	string defining a series of connection retry intervals and the number of attempts at each interval
pubhosts	string	none	comma separated list of hostname:port pairs that is used for a non-interactive publisher to connect to an RMDS.

pubport	string	none	The TCP port on which an interactive publisher will listen for connections. Interactive publishing on RMDS is based on the publisher accepting a connection from the RMDS unlike subscription and non-interactive publishing where the connection is made to the RMDS

The pubhosts and pubport settings are mutually exclusive. A transport can only support a non-interactive OR an interactive publisher, not both

Examples

```
# try to connect to p2ps1 then to p2ps2. In each case use the RSSL default port 14002
mama.tick42rmds.transport.sub.hosts=p2ps1:14002 ,p2ps2:14002

# try to connect a non-interactive publisher to adh1 then to adh2. In each case use
the RSSL default port 14003
mama.tick42rmds.transport.pub.hosts=adh1:14003,adh2:14003

#interactive publisher accepts connections on port 14011
mama.tick42rmds.transport.pub.pubport=14011
```

4.2.2. Authentication

Login to an RMDS requires a user name and application ID which may be subject to DACS authentication. This is separate from and independent of any Wombat authentication

property	type	default value	description
user	string	none	The user name used in the RMDS login. If this property is empty then the login username is taken from the mama infrastructure and if that is not present then from the operating system as the currently logged in user. Any value in this property is overridden by the value of the *mama.entitlement.altuserid* property if present
appid	string	255	The application id used in the RMDS login. This setting will be used if there is no value set in the mama infrastructure.. If this value is numeric then it will be used, otherwise the default 255 is used

Examples

```
mama.tick42rmds.transport.rmds_tport.user = fred
mama.tick42rmds.transport.rmds_tport.appid = 245
```

4.2.3. Subscription throttling

These settings control the behaviour of the subscriber and may be of use when making large numbers of subscription requests

property	type	default value	description
maxdisp	int	1000	Maximum number of subscription requests dispatched before the subscriber thread reads incoming data. This allows a balance to be obtained between processing subscriptions and processing incoming data. Its generally only relevent when large numbers of subscriptions are made in a short period of time, for excample when opening a big spreadsheet
maxPending	int	1000	Maximum number of subscriptions requested from the RMDS but not yet returned. This prevents swamping the RMDS

Normally there is no need to set these properties as the default values are adequate

4.2.4. Data dictionary settings

The mama dictionary that is delivered when a mama client makes a dictionary subscription is built from a set of mama field definitions and the field mapping table that is used to convert fields in the rssl message into mama fields. The mapping process is described elsewhere in this document

property	type	default value	description
fieldfile	string	none	Location for a file containing the RMDS data dictionary. If this setting is not specified, the transport will subscribe to the data dictionary from the RMDS. The file is usually named "RDMFieldDictionary"
enumfile	string	none	Location for a file containing the RMDS enum type definitions. If this setting is not specified, the transport will subscribe to the defintions from the RMDS. The file is usually named "enumtypes.def"
fieldmap	string	none	Location of the field mapping file. This file is used to map specific RMDS fields to mama fields. If no file is specified then fields are mapped from the RMDS dictionary according to the setting of the unmapfld property.
mama_dict	string	none	Location of a mama dictionary file. If no file is specified then a predefined set of mama fields are used. These are the "reserved" mama fields and the mamda book fields.

unmapfld	bool	true	Controls how unmapped fields are handled. If false then only those RMDS fields specified in the fieldmapping file are processed. If true then all RMDS fields are converted. Those that are not specified in the field mapping file are given a fid that is allocated sequentially starting from the value specified in the 'fidoffset' setting and a mama type that best corresponds to the TR field type. The TR field names are used
fidoffset	int	0	Starting value for fids allocated for unmapped fields. If no value (or 0) is specified then fid allocation stars at the next available fid value. This will be the highest value found in the mama_dict file (or the predefined fields) and in the fieldmap file (if one is specified)
dictsource	string	"WOMBAT"	The source name used to publish the data dictionary. In most cases existing (and new applications) will follow the example in mamalistenc and subscribe on the default source. This option allows applications that expect a different name to run unchanged.

Dictionary and mapping defaults

If you want to use RMDS field names in your mama client application then simply allow ALL of this set of properties to default.

The result will be

- The RMDS dictionaries are subscribed from the RMDS
- All the RMDS fields are mapped with the names unchanged and sensible defaults for the types
- The mapped RMDS fields are allocated fids that do not clash with any of the mama reserved fields
- The mama dictionary, if your application subscribes to it, will contain all the mama reserved fields and book fields, plus all the RMDS fields

File paths

In all cases, if there is a full path specified then that is where the file will be loaded from. If no path is specified then the file is searched for in current directory and then, if not found, in the WOMBAT_PATH environment variable.

Examples

```
# mama_dict - path for the mama dictionary file
mama.tick42rmds.transport.rmds_tport.mama_dict=mama_dict.txt



# fieldmap - the path for the csv fields map from RMDS to Wombat names.
mama.tick42rmds.transport.rmds_tport.fieldmap=$(APPDATA)\\Tick42\\RttXL\\fieldmap.csv


# fidsoffset - the value of the first synthesised FID of RMDS fields that are not
explicitly mapped.
mama.tick42rmds.transport.rmds_tport.fidsoffset=20000

# fieldfile - the path for the RMDS fields definitions file
mama.tick42rmds.transport.rmds_tport.fieldfile=RDMFieldDictionary


# enumtype - the path for the RMDS enumerations values and strings
mama.tick42rmds.transport.rmds_tport.enumfile=enumtype.def
```

4.2.5. Publisher Properties

While OpenMAMA effectively provides a single publishing API, RMDS provides 3 distinct publishing mechanisms. These mechanisms are described in more detail elsewhere

<TODO - add link to publisher description >

The different publishing types require different connection properties. This section deals with all the other publisher properties

property	type	default value	description
sources	string	none	The name of the interactive source that will be published on the transport. Only required if the transport is going to be used to publish an interactive source.
nisource	string	none	The name of the non-interactive sources that will be published on the transport. Only required if the transport is going to be used to publish a non-interactive source
sub_tport	string	none	Identifies a transport whose subscriber will be used to receive 'new item' requests from an interactive publisher. In the default case this will just be the transport with the interactive publisher itself.
useseqnum	bool	false	specfies whether "post" publishers on this tranport should include the mama sequence number in the rssl message. Typically post is used when several applications are posting to the same source so the default is not to include sequence numbers
onstreampost	string	"off"	Controls whether message posts are constrained to be "onstream" and require a subscription stream in order to be able to post.
<sourcename>.pubtype	string	"i"	"post", "ni", or "I" Specfies the rmds publishing mechanism that will be used when a mama publisher is created for the specified source.For example a setting "mama.tick42rmds.transport.rmdspub.source1.pubtype = post" would mean that all publishers created for the source "source1" on the transport "rmdspub" would use the post mechanism. Depending on the value of this setting other settings may be required
<sourcename>.serviceid	string	"0"	The RMDS service ID for the source name. This should match the serviceId configured in the RMDS for this source. This property is not required for "post" publishers, only for interactive "I" and non-interactive "ni"

Examples

```
# interactive publisher
mama.tick42rmds.transport.pub.sources=SOURCE1
mama.tick42rmds.transport.pub.SOURCE1.serviceid=1
mama.tick42rmds.transport.pub.SOURCE1.pubtype=i


# we do posting on a subscriber
mama.tick42rmds.transport.pub.SOURCE2.pubtype=post
```

4.2.6. OMM Domain

While some middlewares differentiate between Level 1 and Level 2 data by symbol name, RMDS requires an OMM Domain to be specified. The Tick42 bridge supports 2 mechanisms.

The default is to prefix the symbol name with a 'b' for full L2 data (MarketByOrder in RMDS terms) or with an 's' for price aggregated L2 data (MarketByPrice). The prefix is stripped from the symbol name for the request to the RMDS. This provides consistency with conventions used on WOMBAT feeds.

Problems with prefixes

Although in many contexts the default mechanism works well it has 2 problems

1. An 's' or a 'b' character may be a legitimate first character of a symbol name
2. If MAMA is being used for entitlement checking then the prefixed names will not be in the entitlements database - this is not a problem when entitlement checking is being deferred to DACS because that will see the symbol name stripped of its prefix

The alternative mechanism is to attribute the source with a property indicating which OMM domain to use. All symbols are passed unchanged and requests are made on the specified domain.

property	type	default value	description
<sourcename>.domain	string	"any"	Specifies the OMM domain to be used for all subscriptions on this source. Values can be "mp" (MarketPrice), "mbp" (MarketbyPrice) and "mbo" (MarketByOrder) force requests to be made on the specific domain. If the value is set to "any" then the default prefix mechanism is used

Example

```
mama.tick42rmds.transport.sub.SOURCE1.domain=mbp
```

4.2.7. Tick42 Enhanced Properties

The Tick42 enhanced version of the RMDS Bridge provides some additional features including

- Source Discovery
- Ability to process MarketFeed format data

The properties in the table below are all based on *mama.tick42rmds.transport.<transport-name>* (i.e. can be defined per transport)

property	type	default value	description
enhanced.enabled	bool	false	Enables the enhanced properties
enhanced.source	string	"_T42INFO"	Names a source which applications can use to request additional "enhanced" data, such as a list of sources
enhanced.servicelist	string	"_T42_SERVICES"	Names the symbol that can be subscribed on the source to obtain a list of sources (services)
enhanced,servicenamefield.name	string	"T42ServiceName"	The field in which the list of services is returned. The field has the type MAMA_FIELD_TYPE_VECTOR_STRING
enhanced.servicenamefield.fid	string	"9901"	FID for the service name field
disabledataconversion	bool	false	If true, requests that the ADS does not convert market feed nessages to RSSL. Data is delivered to the bridge as MarketFeed messages.