

Elektron Message API

Product Overview

The Elektron Message API (EMA) targets access to Thomson Reuters Elektron, Thomson Reuters Enterprise Platform (TREP) and Thomson Reuters Direct Feed via an ease of use solution.

EMA targets users in need of light weight Open Message Model (OMM) access with minimal lines of code.¹

This document provides a very brief overview of EMA and complements the accompanied interface and example usage. Presently, EMA solely defines an abridged interface to solicit feedback. Intent is for Thomson Reuters to release an interface and library implementation in the future.

Presume readers of this document are moderately familiar with Elektron, TREP, Direct Feed and OMM APIs.

Features

Ease of use functionality:

- Simplified content access through minimal interface management and minimal method calls.
- Minimal administration through implicit start-up and shutdown.
- Defaulted configuration, yet with ability to selectively override fine grain parameters.
- Binary and standards-based OMM formats (e.g., JSON, XML).
- Programmatic simplification (e.g., fluent interface style, strongly typed operations, minimal downcasts).
- Optional implicit dictionary management.

Rich session level functionality:

- Ability to specify, interpret and distribute content whether simple or across a comprehensive hierarchy.
- Rich consumer functionality including subscription, posting, generic messages and private streams.
- Rich operational model, enabling dispatch in API or user thread of control.
- Comprehensive defect handling facilities (e.g., exceptions, logging, tracing).
- Decoupled interface to enable future session level functionality (e.g., interactive and non-interactive publishing).

Deployment Package

EMA deployment package includes the following:

- This document providing a very brief overview.
- Abridged interface definition for typical Level1 and Level2 consumer access, via diagrams and C++ header files.
- Multiple Level1 and Level2 ease of use example implementations.

Future

Intent is for Thomson Reuters to release an interface and library implementation in the future. EMA would exist across multiple programming languages (e.g., C++, Java, .NET). EMA would be fully visible source residing atop the Ultra Performance API (UPA). Providing the full interface and library implementation is contingent on feedback and prior commitments.

¹ Disclaimer: Users may incur loss in performance dependent on underlying API implementation.

