

Practical Application of XML for Algorithmic Trading Integration



By **Kevin M. Young**,
Senior Business Systems
Analyst The Capital Group
Companies, Inc. & FPL
Algorithmic Trading Working
Group, Business Team

<Defining Algorithms in XML>

eXtensible Markup Language (XML) documents are powerful in their simplicity. They are capable of describing and containing data, as well as providing a format for storage, as in a database. More recent XML technologies introduced the XML Schema Definition (XSD) language. An XSD file is itself an XML document that contains a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema. The combination of XML and XSD languages provides powerful conditional logic, dependencies, error checking and validation that the FIX community can leverage.

The Business Team of The Algorithmic Trading Working Group put forth their initial requirements for an XML schema describing algorithms and their parameters in April, 2006. It is

expected the Technical Team of The Algorithmic Trading Working Group will publish an XML/XSD schema that the FIX community can adopt as the standard for defining algorithms, their parameters and business logic needed for algorithmic trading.

“The combination of XML and XSD languages provides powerful conditional logic, dependencies, error checking and validation that the FIX community can leverage.”

Similar to the process of a specification document today, we envision the algorithm providers will publish their algorithms in XML documents that conform to this standard (see Figure 1) . Quant shops, software vendors and OMS/EMS providers will be the consumers of these documents, importing them into their trading applications and extensively using them for critical algorithmic functionality in the GUI, application server and/or database.

The XML file contains information about the algorithmic provider, one or more strategies of the provider, the parameters of the strategies, and the data itself, as

demonstrated in the diagram below. An accompanying XML file may contain the formatting, logic and rules associated with the attributes of a provider, strategy or parameter.

Figure 1. Sample XML Document

```
<?xml version="1.0" encoding="utf-8"?>
<firm name="Algorithm Provider"/>
<strategy
  name="Sample Algorithm"
  fixTag="847"
  wireValue="1"
  locations="NA EMEA"
  applicableMsgTypes="D E"
  supportsCxlRpl="false"
  disclosureDoc="http://www.algoprovider.com/
  disclosure/sample"
  algoExtModel="Custom">
  <parameter
    name="StartTime"
    parameterType="UTCimestamp"
    fixTag="6100"
    required="Y"
    format="yyyyMMdd-HH:mm:ss"
    initValue="09:30:00"/>
  <parameter
    name="EndTime"
    parameterType="UTCimestamp"
    fixTag="6101"
    required="N"
    format="yyyyMMdd-HH:mm:ss"
    initValue="16:00:00"/>
  <parameter
    name="BenchmarkType"
    parameterType="list"
    fixTag="6102"
    required="N">
    <enumPair uiRep="Last Price"
      wireValue="L"/>
    <enumPair uiRep="Mid Point"
      wireValue="M"/>
    <enumPair uiRep="Open"
      wireValue="O"/>
    <enumPair uiRep="Specific Price"
      wireValue="S"/>
  </parameter>
  <parameter
    name="BenchmarkPrice"
    parameterType="price"
    fixTag="6103"
    required="N"
    percision="2"
    increment="0.01"/>
  <constTag
    fixTag="57"
    wireValue="ATG"
    overrideUserInput="true"/>
  <constTag
    fixTag="18"
    wireValue="1"
    appendToUserInput="true"/>
</strategy>
```

The Algorithmic Trading Working Group

The Algorithmic Trading Working Group is part of the Global Technical Committee of FIX Protocol Limited (FPL). The Working Group is comprised of representatives from the financial community including sell-side, buy-side and financial technology providers. The goal of the group is to evaluate and recommend solutions for current algorithmic trading integration issues and is organized into two teams focusing on business and technical issues. The Business Team is responsible for reviewing strategy parameters currently in use in the industry and developing business requirements for the Technical Team. The Technical Team will identify integration options and develop the technical requirements for the group's proposed solution. Potential benefits we are discussing include shortening the deployment time for algorithmic strategies, improving the user experience, and minimizing the use of FIX custom tags.

Another major activity of the Business Team is to create a questionnaire for the buy/sell side vendors on the challenges and opportunities around algorithmic trading integration. The Business Team intends to interview all of the major OMS/EMS providers to fully understand the challenges in the current environment and also discuss our proposed solution.



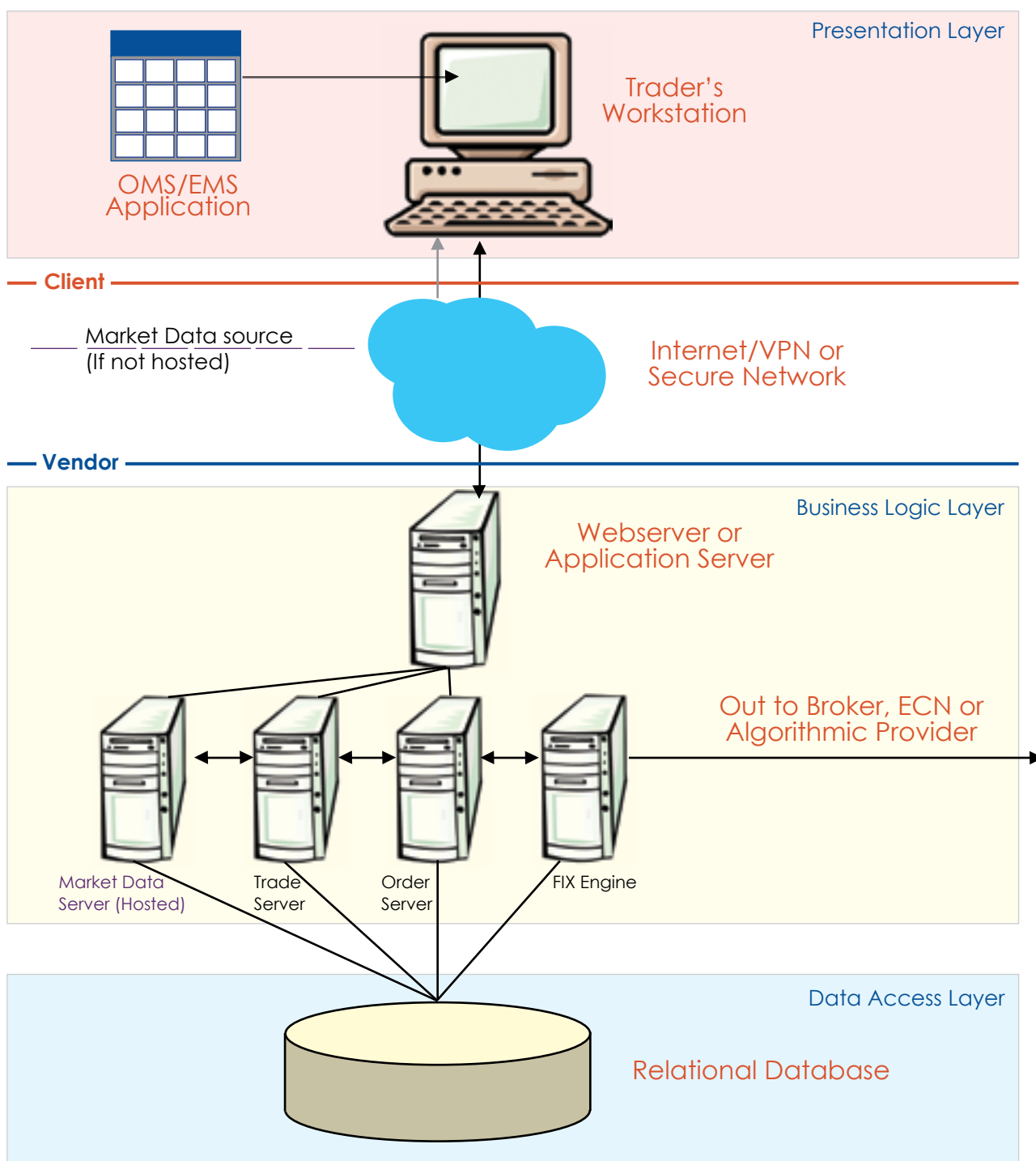
< Overview of OMS / EMS Architecture >

Generic client/server architecture has two types of nodes on the network; clients and servers. As a result, these generic architectures are referred to as “two-tier” architectures. Some networks consist of three different kinds of nodes; clients, application servers which process data for the clients, and database servers which store data for the application servers. This is known as three-tier architecture (a presentation layer, business logic layer and a data access layer). Typically OMS/EMS products are developed as three-

tier or n-tier, meaning greater than three, so we will assume this architecture for purposes of this discussion.

OMS/EMS vendors provide several deployment options to buy-side and sell-side clients with the Hosted ASP model (Application Service Provider) being the most common. In this model, the vendor hosts the infrastructure of the trading application (application servers, database servers, connectivity to Brokers, ECNs, etc.) and provides a front-end application which the client runs locally on the Trader’s desktop (see Figure 2).

Figure 2. Overview of OMS/EMS Architecture



The FPL proposed algorithm definition XML can be applied in any/all three layers of an n-tier OMS/EMS architecture providing a reusable component for quicker development, integration testing, deployment and ongoing maintenance.

<Using XML in the Presentation Layer>

XML is commonly used in the presentation layer for user preferences and application settings. An algorithm definition XML file can be used as reference data for the GUI, providing the data necessary for choosing an algorithmic strategy, defining the parameters of a strategy, valid values for those parameters and logic for conditional fields, dependencies, error checking and validation. If OMS/EMS providers use client-side reference data, error checking and validation, leveraging the XML definition locally can help improve performance by saving a round trip server side validation. Some OMS/EMS providers completely decouple the presentation layer from the business logic layer; utilizing XML in this scenario is covered in the next section.

Taken a step further, XML based technologies like XUL and XAML actually define the application presentation itself in XML. OMS/EMS providers can leverage these technologies by writing an adapter to transform an XML algorithm definition file into application-specific XML used by the XUL or XAML rendering engine.

In this instance, the XML file is not only used for reference data and validation, but also as an integral part of the application itself for rendering the order ticket and its UI controls (or widgets). We envision it possible to develop GUI components that simply load a new algorithm definitions XML file whenever a broker changes or adds algorithms. This loading of XML may be developed without requiring a new compile, build and release as well as a dramatically shortened QA and integration cycle.

<Using XML in the Business Logic Layer>

XML algorithm definitions can also be utilized in the business logic layer. Server-side architecture varies widely between OMS/EMS vendors, but servers are generally "componentized" into groups of logical functions. Some of these functions may include an order server to manage inbound and outbound order flow, a smart-router server for routing to Brokers/ECNs/Algorithmic Providers, a trade server for processing inbound execution messages, etc. Servers process messages according to a set of rules. These rules can be programmatic in the code or can be obtained from an external file or database.

OMS/EMS providers can leverage XML technologies by writing an adapter to transform an XML algorithm definition file into application-specific business rules utilized by their application servers. For example, if OMS/EMS providers do server-side error checking and validation, leveraging the XML definition file in order server or routing server logic

provides a reusable component which can be utilized across multiple brokers and provides easier on-going maintenance. We envision it possible to develop server components that simply load a new algorithm definitions XML file whenever a broker changes or adds algorithms. This loading of XML to change business logic may be done without requiring a new server compile, build and release as well as a dramatically shortened QA and integration cycle.

<Using XML in the Data Access Layer>

XML algorithm definitions can also be utilized in the database layer. In addition to storing transactional data, databases are often used for storing user preferences, permissions and application-specific reference data. OMS/EMS providers can leverage XML technologies by writing an adapter to transform an XML algorithm definition file into a database schema, data elements and reference data of algorithmic providers. These tables and data could then be loaded by the GUI in the order ticket and UI controls and/or used as reference data by the application servers for server logic.

<Conclusion>

You can see the benefits and flexibility offered by the varied application of XML technologies in the presentation, business logic and data access layers of an n-tier OMS/EMS application. We at the FPL believe these benefits lead to adopting and standardizing XML technologies as a solution for current algorithmic trading integration issues. We are always seeking the feedback of the FIX community and seek your adoption and involvement in this developing proposal. Thank you to our members and vendors for your continued support. **FIX**



Any thoughts on this or other articles?
Please send any comments, referring to
this article as Vol 1 Issue 12 GL2, direct to
Edward at edward@fixglobal.com