

# igniteXML

**Customize and adopt  
ACORD and IIA  
to create your Canonical Model**



*Model Manager*



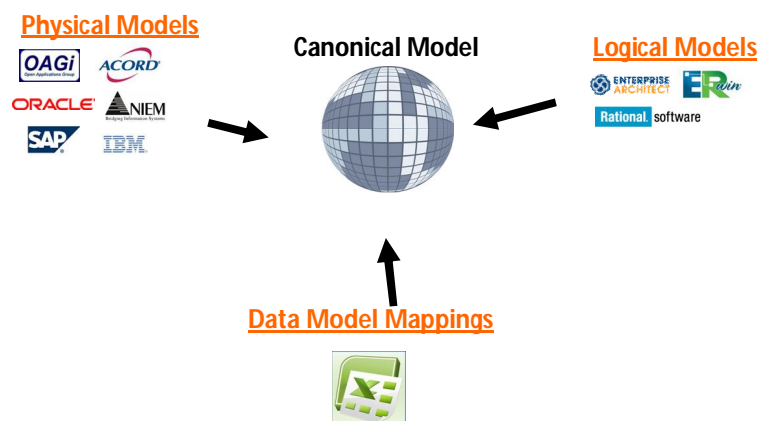
*Business Analyst*



*Integration developer*

The majority of large scale Insurance sector organizations are moving away from point-to-point integration. Somewhere along that journey many have realized that it is essential to adopt a single Canonical Model to ensure data interoperability for data in motion.

## Multiple sources blended into a single model



An effective Canonical Model will need to span both the Logical level (so it can be understood by the business) as well as the Physical Level, as ultimately it will be used to generate XML messages with their mappings to backend systems.

There is often great value in leveraging industry standard physical models as the basis of an organizations or project's Canonical Model. The Insurance sector is well served by standards like ACORD and IIA from IBM. These have created a well defined and general message standard that addresses message content, naming standards and message architecture using common business language. These can be a great starting point for insurance sector organizations looking to apply a reusable, extensible and open standard to their modelling approach.

The flip side of these standards is that they are very large, as well as being complex, especially when it comes to extending them to fit your specific circumstances.

Consequently most people who adopt these standards need to find a way to easily extend them and trim the combined model (model + custom extensions) in order to produce runtime payloads that are efficient.

In the schema management world this is very hard to achieve. Multiple versions of the model are created by multiple people as the Canonical Model is created and updated to meet business and integration needs. Each version needs to be collaboratively managed which is very hard to do when using XML Editors and Source code repositories in the Schema Management world. This task becomes even harder when the standard is updated and the task is to try to work out the implications of the changes to the extended model and XML schemas used in production.

XML messages that are created from the model should ONLY contain what is need for the message requirement. ACORD's size and the way everything hangs off a root element makes it very hard to trim using Schema Editors. Consequently many early attempts to adopt these standards result in messages with a lot of "schema bloat". We recently worked with a company that were stuck in the schema management paradigm and were deploying messages across their ESB that were so large they had to switch off the validation options due to performance reasons. This is a high risk strategy.

## Re-using the Model NOT XML Schema



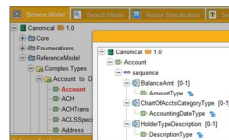
In the Canonical Model Management world, schema is decomposed into an object model where every object is individually managed in a collaborative repository. Extensions, updates, changes, versions, and impact analysis are done at an object level with a role based approval process. Model managers are able to attach metadata to individual objects to provide data lineage, drive model adoption, and assist with mappings.

It is easy for the model manager to import ACORD and IIA into igniteXML. When doing so these large models are automatically broken down into our object model which mirrors the structure of the schema world. Once imported, the model is easy to extend in a way that takes account of the need to handle later updates to ACORD or IIA.

The model is made available to integration developers and business analysts as reusable objects allowing them to easily navigate the model, find what they are looking for, build or reuse specifications and select components to auto-generate XSD messages and mapping reports.

The integration developers can make full use of the web based trimming functionality in igniteXML to remove the schema bloat and produce runtime optimized messages and schema that are 100% compliant with the standard but are a fraction of the size. This ensures all messages are consistent and derived from the latest version of the model – igniteXML combines governance and reuse.

## A common reference point for all



Business Analyst

Reference the model in specifications



Integration Developer

Build Messages and mappings from the model

- All messages
  - 100% consistent
  - Optimized for runtime
  - Clearly understood
  - Easy to maintain
  - Mapped to data model
- The result...
  - Reuse and Agility
  - Runtime performance
  - Less work
  - Lower costs
  - Data governance

digitalML, the makers of igniteXML, is the enterprise Canonical Model Management company. igniteXML is used by Integration and Service Oriented Architecture (SOA) teams in Fortune 2000 and large Government organizations. Through patented technology, igniteXML has changed the ways that customers manage and ensure adoption of their Canonical Model.

**North America Corporate Headquarters:**

digitalML-USA Inc.  
580 California Street, Suite 1600  
San Francisco, CA 94104  
Tel: 1.415.373.0302 Fax: 1.415.373.0307  
Email: [jbogard@digitalml.com](mailto:jbogard@digitalml.com) and [support@digitalml.com](mailto:support@digitalml.com)

**UK Corporate Headquarters:**

digitalML Ltd.  
Atrium Court  
The Ring  
Bracknell, Berkshire, RG12 IBW  
Tel: 44 1344 390542  
Email: [jsindall@digitalml.com](mailto:jsindall@digitalml.com) and [support@digitalml.com](mailto:support@digitalml.com)

Why not invest an hour in an online demo to  
simplify your Enterprise Integration ?



Call: USA 415-373-0302  
UK (+44) 01344-390542  
Email; [jbogard@digitalml.com](mailto:jbogard@digitalml.com)