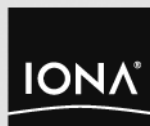




IONA Artix Data Services

Technical Overview

IONA Technologies May 2007



Making Software Work Together™

Executive Summary

IONA Artix Data Services is a metadata management, data modelling, transformation and integration toolkit based on open technologies. It can be used either stand-alone or orchestrated into application frameworks to provide parsing, validation and transformation services, not just for XML-derived data models, but also for most any data format.

The Artix Data Services Standards Libraries are specialized messaging libraries providing bundled sets of components and services that extend the base toolkit for highly targeted and business-specific usage scenarios.

The Artix Data Services product is a development efficiency toolkit. It significantly decreases the time to deploy new integration components, and reduces the cost and time burden of ongoing support and maintenance. Artix Data Services eliminates specific technical lock-in by abstracting the messaging data services from the underlying transport and integration infrastructure.

Artix Data Services components can be deployed on any Java run-time deployment infrastructure environment although they are predicated on none. Artix Data Services is designed to leverage open technology standards and to be deployed on application server platforms, in Spring containers, in Jini environments, with various open and closed source ESB implementations, and the native IONA platforms - and the list keeps evolving based on real world demand and adoption.

The following is a technical introduction the major components of Artix Data Services:

- ❑ **Artix Data Services Designer** - design-time environment that includes metadata management and semantic constraint and validation rule implementations
- ❑ **API** - run-time environment application program interfaces
- ❑ **Artix Data Services Standards Libraries** – support for specific industry standards
- ❑ **Reference Implementations** – deployed solution examples

Artix Data Services is one component in the IONA Artix advanced SOA infrastructure suite, designed to streamline, modernize and lower the operating costs of complex and heterogeneous IT environments. IONA has a proven track record of delivering mission-critical infrastructure, and has built many of the earliest and largest SOAs for Global 2000 customers including Credit Suisse, BellSouth, Raymond James & Associates, Marconi, and Deutsche Post (DHL).



The Artix Data Services Designer

The Artix Data Services Designer is the design-time Integrated Development Environment [IDE] used for defining and managing the data models, transformations and deploying Artix Data Services beans.

The Artix Data Services Designer uses configuration files (.iop) to describe the file system locations for data model [.dod], transformations [.tfd] and other associated files. Working within a specific problem domain is usually reflected by creating a specific Project.

Artix Data Services deployed components are used in a variety of contexts. Hence their creation is customisable via a series of options specified in the Deployment Descriptors. This gives a developer the flexibility to customise and persist their design time preferences based on any specific requirements.

Data model and component versioning complements the metadata management capabilities of Artix Data Services by enabling versioning of data model components and resultant deployed code at any level. Through advanced usage of interface implementation, this allows a single object to be presented as many different versions concurrently.

Data model specializations enable a user to inherit a base data model, be it an industry standard or firm specific model, and to specialise the model by restricting the structure or valid values, or extending the model by adding additional fields or relaxing validation in certain contexts. This is a very powerful feature in implementing internal normalised data models based on industry standards, or enforcing market practice conventions on a bilateral or multilateral basis.

The transformation definition feature provides a graphical user interface to associate dissimilar source and target data models and the semantic transformation logic that is required to transform between them. Once the transform is defined, one deploys Java code from it in the same way as one deploys code from data models. The generated Java code executes the transform at runtime on instance data compliant with the source data model(s). Since the transformation code has been written as Java code it is highly performant when compared with other black box transformation solutions or XSLT-based approaches.

The Diff and Merge Tool enables a detailed, component-level comparison and optional merge of two data models. This is invaluable for metadata management tasks and impact analysis for revisions to base or specialized data models.

The Viewer is an integrated tool for visualizing the deployed code components generated by the Artix Data Services Designer. This is useful to test and debug data models, associated validation rules, and instance data. The Runner does for Artix Data Services transforms what the Viewer does for data models. It provides the user with a GUI environment within which they can run the generated transformation code under controlled conditions to test that the semantics of the transformation rules they have defined.

The Artix Data Services design-time environment is commonly used by technical teams for the development of integration logic. To facilitate this, the Designer also generates ANT scripts with specific ANT targets that can be easily integrated into a larger build environment. Artix Data Services also includes integration with collaborative development Version Control Systems such



as Concurrent Version System [CVS], IBM Rational® ClearCase®, and SubVersion [SVN] to ensure consistent and managed control of development projects.

Metadata Management

Artix Data Services is a powerful tool for manually creating and maintaining metadata definitions, but is frequently not the only metadata repository in use. Therefore Artix Data Services is designed to exchange metadata structure information with various pre-existing data sources and targets:

- ❑ Artix Data Services supports the import and export of metadata models as **W3C XML Schema Documents [XSD]** and or **XML Document Type Definitions [DTD]**.
- ❑ The ability to import **XML Instance** documents is invaluable where only XML instance data exists and there is a need to automatically derive the structure directly from these instance documents. Artix Data Services can also import and export metadata definitions in the OASIS **RELAX-NG** schema language.
- ❑ The **Text File Importer** enables automatic creation of data models from text based, flat, fixed-length, or delimited format files. This facility is particularly useful when there are multiple heterogeneous file formats such as CSV exports from spreadsheets, or for transforming highly complex structured messages into delimited file formats for legacy application logic.
- ❑ The **Java Class Introspection** feature enables the automated creation of data models directly from Java classes. This feature is commonly used where existing Java code exists and the associated Java objects need to be transformed into other data models or referenced directly as Artix Data Services beans.
- ❑ The **XMI Metadata Interchange [UML and MOF]** feature provides the capability to import and export data models to a format that can be rendered by specialist UML and architectural design tools such as Rational Rose, Together, MagicDraw, Sparx Systems.
- ❑ The **Database Importer** feature provides the ability to interrogate an RDBMS and import data model structures from it based on table relationships, SQL queries or stored procedures. Database structures can be recreated from a data model automatically by running Hibernate over the generated code.
- ❑ **HTML Documentation Generation** generates human readable HTML designed for reference documentation of complex data models.

Semantic Constraint and Validation Rules

As previously observed, Artix Data Services supports XML and non-XML data models. It also supports the overlay of semantic rules and content validation using a range of constraint implementations. These are used extensively within the Artix Data Services Standards Libraries and can be implemented on any data model.

Artix Data Services includes the ability to specify XPath expressions to evaluate cross-field dependencies. A simple common example is to verify that a trade settlement message has the settlement date after the trade date.



Artix Data Services also has the capabilities to include Java code that implement specific validation logic. A common example is IBAN validation in SWIFT FIN payment messages.

On the basis that a large amount of data will already reside in existing applications or databases, Artix Data Services enables Custom Factory Classes to be invoked within the validation rules. This may be a class that validates against external static data in real time, updates external systems, or may be a piece of pre-existing logic that can be reused in this context. These custom services can be coded to do anything you may require.

For situations where validation is dependent upon scheme information found elsewhere in the instance data, Domain Constraints can be defined to allow run-time configurable custom validation base on standard enumerations or other custom written validation rules. Domain constraints are manifested in FpML as Coding Schemes and can also be applied in the SWIFT FIN context to add data source scheme validation based on industry

guidelines.

Artix Data Services API

The core of the Artix Data Services run-time is the application program interface [API] providing access to the deployed components via a series of standardised access methods irrespective of the origin of the source data model as XML or non-XML data sources.

Artix Data Services beans are Java classes generated from data model files, either via the Artix Data Services Designer or via ANT scripts invoking the Java class generation via the API. These Java beans contain compile-time, type-safe get and set methods as well as optimized serialization, cloning and versioning mechanisms. The style of the generated code can be controlled through the use of deployment profiles thereby allowing one data model to generate java code in various packages, with various class suffixes and in different styles. Artix Data Services beans inherit derivation and implementation hierarchies defined in the original data model but all eventually subclass components within the Artix Data Services API. This makes them generically addressable and interchangeable and therefore allows the Artix Data Services API to present the following interfaces:

- ❑ **Document Object Model (DOM)** - enables consistent access to data model structure and content irrespective of the data source. It allows in memory traversal of the structure by 'dumb' code, allowing developers to write 100% generic code that can interpret your parsed data structures.
- ❑ **XML (SAX)** - can be configured to be an active producer and consumer of SAX events to allow highly optimized data interchange with other systems and APIs. The lightweight nature of SAX events provides an extremely efficient, lossless conversion to and from the Artix Data Services API.
- ❑ **XPath 2.0** - for access to any XML or non-XML Artix Data Services bean object via XPath expressions. This gives you a powerful lookup mechanism without requiring any Java coding. The XPath 2.0 interface provides a fully type safe, namespace aware language for accessing data.



- ❑ **XQuery 1.0** - can be used to access Artix Data Services beans. This provides what is commonly termed a "SQL for XML" interface to Artix Data Services beans. It gives another powerful, easy to use, type safe access mechanism via either the basic or FLWOR XQuery syntax.
- ❑ **XSLT 1.0** - all Artix Data Services bean objects can also be accessed directly via XSLT 1.0 for the definition of transformation relationships between dissimilar data models. This means populated data model structure can be used as the *source* of any XSL transformation in the same way as one can use a standard DOMSource or SAXSource.
- ❑ **Hibernate 3.1** - for high performance database persistence and query by generating the Hibernate descriptors directly from the data model. This gives a quick and easy, hassle free Object Relational Mapping (ORM) from a hierarchical or object-oriented data structures into the relational form used by a database. It also gives an easy to use query mechanism via primary key values, SQL or the Hibernate Query Language (HQL).

Artix Data Services Standards Libraries

Each Artix Data Services Standards Library provides fully maintained data models and, with the design-time packages, specific test instance data for every model. The Standards Libraries provides a maintained high confidence level baseline of data models and test cases that can be quickly integrated within automated build and testing development environments, and provide proven production strength ready to go messaging data services. Available Standards Libraries include

- ❑ **SWIFT FIN MT** - includes all system and common group message types (MT0nn system, MT n9n common group, and MT1nn-MT900 categories). The Enterprise Standards Libraries includes all FIN message types as published by SWIFT, while the Select Standards Libraries enables users to select any subset of only those SWIFT FIN message types that are relevant to a specific business scope. The Standards Libraries further includes a full set of test messages for every FIN MT, and every network validation rule – in total in excess of 20,000 test cases. The released Standards Library is always kept synchronized with the current SWIFT Standards Release, and historical SWIFT Standards Releases from 2002 onwards are supported for off-net and historical integration requirements.
- ❑ **SWIFTNet Solutions /SWIFT MX** - SWIFTNet Solutions and services run on SWIFT's private Secure IP Network (SIPN). The Artix Data Services Standards Libraries for SWIFTNet Solutions include support for such as SWIFTNet FUNDS, Trade Services Utility, Cash Reporting, Exceptions & Investigations, and new and piloted standards as they become available.
- ❑ **ISO 20022 UNiversal Financial Industry message scheme (UNIFI)** - Artix Data Services adds a level of abstraction above ISO 20022 metadata in the same way that it does with other XML Schema standards such as FpML, and non-XML standards such as SWIFT FIN. The same design environment can be used irrespective of metadata source, within a consistent design environment, and with a consistent API to the deployed Artix Data Services beans. The Artix Data Services Designer includes the examples of the current



ISO 20022 UNIFI standards release XML Schema, XML instance documents, and applicable message definition reports. The ISO 20022 Standards Libraries adds implementation of the constraint validation rules as described in the standards releases, and a guarantee to ensure that the Artix Data Services design-time tools and Standards Libraries support the current and emerging standards, and create production-quality solutions.

- ❑ **Single European Payments Area (SEPA)** –The SEPA Standards Library is an ISO20022-based implementation under the European Payments Council (EPC) scheme rulebooks for the SEPA Direct Debits and Credit Transfer instruments. The SEPA Standards Library includes supported implementations of the data models, the validation rules and a suite of instance test documents.
- ❑ **Financial products Markup Language (FpML)** - This Standards Library is designed to meet the needs of firms transacting derivatives business who need their applications to communicate effectively using the ISDA FpML standard, both internally and with external trading partners. Subscription to the Standards Library gives access to a collection of components and services focused very specifically at the FpML power user. The FpML Standards Library includes data models pre-imported from the entire range of historical FpML schemas published by ISDA, implementations of the full set of published FpML validation rules, and is accessible from the Designer. This easily extends and specializes the base FpML models and associated validation rules to suit the requirements of unique trading and application environments.
- ❑ **DTCC Deriv/SERV Trade Information Warehouse** and **SwapsWire** are specialization of the ISDA FpML standard by the respective entities. The Standards Libraries for these include the base generic FpML Standards Library, overlaid with the data model and rules specializations of the published specifications.
- ❑ **TWIST** - The TWIST standards initiatives are targeted at driving adoption of non-proprietary XML-based standards for a range of financial services. The TWIST standards implementations are considered complementary to other standards bodies such as MDDL and SWIFT and in fact in some instances wrap subsets these standards within TWIST implementations. IONA provide the TWIST Standards Library as a fully implemented, maintained and supported library of TWIST standards in the same way it does with the other standards, and where these standards cross reference each other maintains the integrity of the interdependencies. As with other open XML-based standards, Artix Data Services provides the metadata management capabilities to extend and specialize the models, and deploy on a common architecture with the confidence that comes from having committed current and future support of the standard.
- ❑ **Financial Information eXchange Protocol (FIX)** – the FIX Standards Library supports the FIX and FIXML standards. The FIX standards are frequently specialized by exchanges and firm. The ability to model these specializations and deploy generated model driven parsers enables faster and more maintainable deployments.
- ❑ **CREST DEX and CREST ISO** - CREST is the United Kingdom's real-time electronic settlement system for UK and International shares, and UK government bonds (Gilts).



IONA support the definitions of all the different message types as published by CREST Co in the File Transfer Data Exchange Manual (or 'DEX' standards), and also the CREST ISO 15022 based messages. In both cases the Standards Library implements and supports current and future CREST standards.

Artix Data Services Reference Implementations

The Artix Data Services Reference Implementations are packages of Artix Data Services deployed solution examples. They are intended for system architects and developers as executable examples that demonstrate the layers of functionality described in this technical overview. They are targeted at a selection of common specific integration solution domains, and illustrate the capabilities of the technology within these contexts. These are useful example starting points to review usage in the real world.

The reference implementations include:

- ❑ Deployed java source code and Apache ANT scripts
- ❑ Associated class libraries
- ❑ Developer reference JavaDoc
- ❑ Test instance documents and test cases
- ❑ Source code for all examples
- ❑ Instance of the API .jar for run time deployments

The reference implementations packaged examples are available online, and are also packaged and installed within the standard installation package.



IONA Artix: Advanced SOA Infrastructure Suite

Artix Data Services is one component in the IONA Artix advanced SOA infrastructure suite.

IONA Artix comprises a comprehensive suite of products to streamline, modernize and lower the operating costs of complex and heterogeneous IT environments. The suite includes

- ❑ **Artix ESB** – connects any service consumer with any service providers using any middleware by deploying, managing and securing a SOA without requiring a centralized hub
- ❑ **Artix Registry/Repository** – a phonebook-style listing of all available services with automatic provisioning and monitoring of services, to maximize reuse and ensure continued adherence to enterprise policies
- ❑ **Artix Orchestration** - facilitates the composition of fine-grained functionality into reusable services using BPEL to create business-level services
- ❑ **Artix Data Services** – a metadata management, data modeling, transformation and integration toolkit to abstract data services from the underlying transport and integration infrastructure
- ❑ **Artix Mainframe** – a service-enablement engine that extends mainframe systems to integrate with off-host systems without the additional expense of running all applications on the mainframe
- ❑ **SOA Management provided by AmberPoint** - governs and manages a SOA implementation by monitoring the health of Artix ESB endpoint

IONA Technologies PLC
The IONA Building
Shelbourne Road
Dublin 4 Ireland

Phone +353 1 637 2000
Fax +353 1 637 2888

Support: support@iona.com

WWW: www.iona.com

IONA Technologies Inc.
200 West Street
Waltham MA 02451
USA

Phone +1 781 902 8000
Fax +1 781 902 8001

Training: training@iona.com

IONA Technologies Japan Ltd
Akasaka Sanchome Building 7/F
3-21-16 Akasaka
Minato-ku Tokyo Japan

Phone +813 3560 5611
Fax +813 3560 5612

Sales: sales@iona.com

IONA, IONA Technologies, the IONA logo, Orbix, High Performance Integration, Artix, Celtix, Celtix Enterprise and Making Software Work Together are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that may appear herein are the property of their respective owners.

While the information in this publication is believed to be accurate, IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA shall not be liable for errors or omissions contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

COPYRIGHT NOTICE. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photo-copying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third-party intellectual property right liability is assumed with respect to the use of the information contained herein. This publication and features described herein are subject to change without notice.

Copyright © 1999-2007 IONA Technologies PLC. All rights reserved.

Any trademarks, service marks, or product names that may appear herein are the property of their respective owners.



Making Software Work Together™