

# Getting Started with MOF, XMI, and JMI

The MDA Metadata Management Standards

## Summary

This article provides a short introduction to MOF, XMI, and JMI.

**Created on:** 2 May 2006

## Author Bio



David Frankel's career in the software industry spans over 25 years, during which he has had vast experience as a software developer, architect, and technical strategist. He is the author of many published articles and sole author of the book Model-Driven Architecture®: Applying MDA® to Enterprise Computing, published by John Wiley & Sons in 2003. He also is lead editor of the book The MDA Journal: Model Driven Architecture Straight from the Masters, published by Meghan-Kiffer Press in 2004. He served several terms as an elected member of the OMG Architecture Board, and was intimately involved in the OMG's launch of Model Driven Architecture. He is the co-author of several industry standards, including COM-CORBA Interworking, the UML® Profile for CORBA®, and the UML Profile for EJB. He currently is Lead Standards Architect for Model-Driven Systems at SAP Labs.

## Table of Contents

What are MOF, JMI, and XMI? .....	2
What is Metadata? .....	2
The MOF Core.....	2
XML Metadata Interchange (XMI) .....	3
The Biggest Misconception About XMI.....	3
Java Metadata Interface (JMI).....	3
MOF 1.x and 2.0 .....	3
MOF, XMI, and JMI at SAP .....	3

## What are MOF, JMI, and XMI?

MOF is the metadata management standard that lies at the heart of the OMG's Model-Driven Architecture. MOF addresses a common problem in today's enterprises whereby different kinds of metadata are stuck in silos, creating barriers to integration.

MOF and its sister standards JMI and XMI define common mechanisms for:

- Representing metadata via Java APIs
- Serializing metadata in XML
- Tracking different versions of metadata and
- Managing the lifecycle of metadata.

## What is Metadata?

Metadata literally means data about data. Originally, the term "metadata" was applied exclusively to data models, i.e. database schemas. Data models are data about the data that makes up the rows in a database. What the MDA world calls metadata today also includes UML models, workflow models, business process models, business rules, service definitions, configuration and deployment descriptors, and so on.

Consistent mechanisms for managing different kinds of metadata help to break down metadata silos, making it easier to integrate tools that use the various kinds of metadata. They also make it possible to automate many aspects of metadata management, avoiding the need to hand-code special APIs, serialization logic, parsers, version tracking, and lifecycle management for each kind of metadata.

## The MOF Core

The MOF Core standard specifies a language used to model metadata. A model of a kind of metadata is called a *metamodel*, so we call the MOF Core language a *metamodeling language*. MOF's metamodeling language is a subset of [UML](#)'s class modeling language; thus, in practice, we use UML modeling tools to create MOF-compliant metamodels.

The MOF Core metamodeling language is independent of data technologies such as XML, relational database, and so on, and is also independent of programming languages and other platform technologies. Sister standards define mappings of the Core to specific technologies.

## XML Metadata Interchange (XMI)

XMI maps the MOF Core to XML technology. XMI is an OMG standard.

Tools that implement XMI consume MOF metamodels, and generate code that serializes metadata to XML documents and that parses metadata represented in XML documents. They also generate XML schemas and DTDs that specify the XML format for the metadata.

## The Biggest Misconception About XMI

The OMG fed the metamodel of UML to an XMI tool, and the tool used XMI's MOF-to-XML mapping to produce a DTD that defines the XML format for UML models. This DTD became the official XMI format for exchanging UML models among tools. Many people are under the misconception that this DTD *is* XMI. In reality, XMI is not this particular DTD for UML. XMI is the *mapping* that, in this case, consumed the UML metamodel and produced this particular DTD for UML. The OMG and other players have fed other metamodels to XMI tools to produce other XMI DTDs and schemas.

## Java Metadata Interface (JMI)

JMI maps the MOF Core to Java technology. Interestingly, the OMG did not define JMI. Instead, the Java Community Process defined it, as JSR #40.

Tools that implement JMI consume MOF metamodels, and generate code that defines and implements Java APIs for representing metadata as Java objects. The APIs include operations for creating, reading, updating, and deleting metadata from an in-memory or persistent store.

## MOF 1.x and 2.0

Recently, the OMG finalized MOF 2.0. MOF 2.0's metamodeling language is tightly aligned with UML 2.0's class modeling facilities. MOF 2.0 also defines versioning mechanisms, lifecycle management mechanisms, and model transformation mechanisms.

## MOF, XMI, and JMI at SAP

SAP implements MOF, JMI, and XMI in the Composite Application Framework (CAF) and in the Business Intelligence (BI) Development Kit. This means that CAF and BI metadata conform to these standards.

### Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.