

## Tools and Sources for Application Integration Metadata

**Integration metadata is held in diverse locations, making it hard to find. Comprehensive tools for managing integration metadata do not exist as commercial products. Enterprises must make do with partial solutions and custom invention.**

---

### Core Topic

Application Integration and Middleware:  
Application Integration

### Key Issue

What new technologies will be instrumental  
in enabling the integration of packaged  
applications with legacy systems?

The benefits of having comprehensive integration metadata are clear (see “Best Practices: Managing Application Integration Metadata,” TG-15-1933). However, the means of collecting and managing this metadata are not at all straightforward. First, integration metadata is held in a variety of software facilities using disparate data models. Second, the commercial software tools for maintaining an application integration repository are incomplete.

## Where Is Integration Metadata?

Integration metadata includes the descriptions of the communication content (what is in the data being sent between application systems), the touchpoint mechanics of the sending and receiving systems, and the interaction process and business implications (see “Identifying Useful Kinds of Application Integration Metadata,” TU-15-1931). In all enterprises, this detailed metadata is held in different formats in different tools, such as:

**The application systems** — They store their metadata in many forms, including database management system (DBMS) catalogs, object request broker (ORB) interface repositories, packaged application metadata repositories, text documentation and program source code libraries (including COBOL copy books and program headers). This metadata is generally authoritative because it reflects the physical implementation of the communication (the actual data structure sent or received by an application touchpoint). However, the metadata may be difficult to find or manipulate since it is in so many places, each using a different format and different semantics.

### Gartner

Entire contents © 2002 Gartner, Inc. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Gartner shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.

**XML tools** — Document-type definitions (DTDs) and schemas may be captured and maintained using tools such as XML Spy from Altiva, XMetal 3 from SoftQuad or X-Slate from XYZFind. These are helpful just for XML data, but incomplete with respect to touchpoint protocol and quality-of-service issues.

**Systematic development tools** — Database design tools (e.g., Computer Associates' ERwin ModelMart or Sybase's PowerDesigner MetaWorks), business process modeling and management tools, rules engines and object-oriented (OO) modeling tools (e.g., Rational Rose's ClearCase) contain metadata. Some of this is relevant for integration, although much of it is intra-application in scope.

**Integration broker suites** — Products such as IBM's Websphere MQ Integrator, Microsoft's BizTalk, SeeBeyond's e\*gate, Tibco Software's ActiveEnterprise, Vitria Technology's BusinessWare and WebMethods' Enterprise have metadata facilities that define detailed message schemas and mapping rules specifically aimed at integration. These products require such metadata to enable transformation, content-based routing and adapters. The vendors mostly call these facilities "repositories," although Gartner traditionally refers to these as "message dictionaries." These are incomplete as integration repositories because they only cover data that is transmitted through their middleware.

They also do not hold metadata for other kinds of touchpoints, such as files, databases, screen formats (for screen scraping) or calls (e.g., to RPC, COM, CORBA, RMI), except where their adapter-building tools have created gateways from these foreign environments to the broker suite middleware. These integration broker repositories generally lack much semantic information and more closely resemble DBMS catalogs than fully rounded repositories.

**Other middleware** — Data transformation tools (e.g., from Informatica), electronic data interchange (EDI) tools (e.g., from GEIS, Peregrine/Harbinger and Sterling Commerce) and extraction, transformation and loading (ETL) tools (e.g., from ETI) manage their own integration metadata for files, databases and sometimes message-oriented middleware (MOM) messages. However, they ignore other forms of communication, such as method calls, RPCs and screen scraping.

**UDDI directories** — An increasing number of application integration touchpoints are being built as Web services. Web services are characterized by XML message formats and the use of the Simple Object Access Protocol (SOAP). Web service method signatures, "where from" and some dialog characteristic

data will usually be documented using the Web Services Description Language (WSDL). WSDL is then stored in a Universal Description, Discovery and Integration (UDDI) directory. UDDI directories, public and private (intraenterprise), will be a useful source of integration metadata. However, UDDI directories only apply to Web service touchpoints and, for the foreseeable future, will not contain all potentially useful metadata, even for Web services.

**Generalized enterprise IT metadata repositories** — These repositories (e.g., Allen System Group's Rochade or Computer Associate's Platinum PR/OEE and PR/MVS) are intended to encompass many different kinds of metadata, including that held in some of the tools listed above. These products are mostly focused on the internal data model (i.e., the data that is within the application system, which is irrelevant to integration work). However, they also contain some specifications for database tables or other objects that represent transaction data or events that are sent to or received from other application systems (i.e., touchpoints used for integration). These portions of an application data model are part of the exchange data model and are thus logically part of an application integration repository. However, IT metadata repositories are not used by all enterprises; where they are used, they may not cover all of the application systems, so the majority of integration metadata will not be found in these sources.

### **Combining Metadata From Disparate Sources**

Each of these specialized software facilities contains valuable, detailed metadata that is needed for enterprise integration development and management purposes. Collectively, these implementation-specific software facilities hold all of the automated metadata relevant to integration. (Some metadata needed for integration is not here because it has not been created.) If these tools are centrally managed by an integration competency center, they can be considered part of a virtual enterprise application integration repository.

However, even the combination (union) of this metadata is not a complete set of information needed for application integration. An integration competency center will usually need to generate additional new metadata on semantics, throughput, quality-of-service attributes, message warehousing, security and other aspects of integration interactions.

Enterprises have a choice of three strategies for coping with metadata diversity (see "Strategies for Managing Application Integration Metadata," T-15-5501):

- Do nothing — leave integration metadata spread out and unmanaged
- Create a comprehensive centralized metadata repository to physically hold a complete set of integration metadata
- Create a small, centralized repository with some complementary integration metadata, but leave most integration metadata in physically separate specialized software tools that are centrally managed

For the “do nothing” strategy, no special additional software is required because no new metadata will be managed. For the second and third strategies, however, the integration competency center will have to buy or build some sort of software to manage the new centralized metadata repository. None of the software tools described above is general enough to be used for this purpose, because none is broad enough to span all of the protocols and attributes that are required.

### **Enterprise IT Metadata Tools Are Not Aimed at Integration**

Even OO modeling tools and the enterprise IT metadata repositories do not track many of the properties essential to interapplication exchanges (although they do often support good configuration management facilities, such as checkout, check-in and versioning).

Examples of enterprise IT metadata management tools:

- Adaptive’s Information Manager (formerly Unisys’ UREP)
- ASG’s ASG-Rochade, ASG-Manager and ASG-Vista
- Computer Associates’ CA/Platinum
- International Software’s E-GEN/MAP
- Microsoft’s SQL Server 2000 Meta Data Services
- Oracle’s Repository
- Rational Software’s ClearCase (the repository for Rational Rose)
- Soamai’s Wincap
- Softlab’s Enabler

These tools tend to focus their attention on the traditional notions of an object model or data model, i.e., the information models that are primarily internal to an application system or set of application systems from the same development team, rather than to the externally focused exchange information model. For example, these products generally do not capture transformation rules (because they assume a consistent data model),

communication protocols, details on “where from” and “where to” and other aspects of application system interactions.

## Analysis and Recommendations

A truly comprehensive integration repository would be inherently difficult to implement because the semantics of the metadata in each of the specialized sources are so different. For example, the attributes that describe a 3270 screen, an XML document, a database table or an EDI file differ radically. There are also many organizational obstacles to consolidate metadata into a single repository. Each business unit and its respective application system project teams will need to maintain control over the metadata in its applications. The ownership of metadata for application integration will be as difficult to clarify as the ownership of traditional intra-application metadata (see “Data Warehousing Technical Metadata Management Issues,” TG-09-8995).

For the foreseeable future, enterprises will need to use pragmatic design and management practices, along with some custom coding, to deal with the technical and organization diversity of integration metadata. We are not aware of any commercially available integration metadata tool that offers a complete solution for comprehensive integration metadata management in a single product.

Some partial solutions are available. For example, Contivo's Thesaurus and Repository enable detailed metadata management (schema mapping and transformation, including support for ontology and vocabulary concepts) in a manner that is compatible with several different integration broker suites. Zonar's Information Sharing System also implements a powerful middleware-independent mechanism for manipulating information semantics and transformation using a unique vocabulary-based approach, although it lacks off-the-shelf connections to most other integration tools. These products can be helpful for enterprises that are implementing ambitious forms of transformation and integration metadata management, although neither is a truly comprehensive repository tool.

**Bottom Line:** Commercial products for managing the metadata needed for systematic application integration are fragmented and incomplete. Integration architects must design their own custom solutions by combining disparate partial solutions and custom coding.

### Acronym Key

<b>COM</b>	Component Object Model
<b>DBMS</b>	Database management system
<b>DTD</b>	Document-type definition
<b>EDI</b>	Electronic data interchange
<b>ETL</b>	Extraction, transformation and loading
<b>MOM</b>	Message-oriented middleware
<b>OO</b>	Object-oriented
<b>ORB</b>	Object request broker
<b>RMI</b>	Remote Method Invocation
<b>RPC</b>	Remote procedure call
<b>SOAP</b>	Simple Object Access Protocol
<b>UDDI</b>	Universal Description, Discovery and Integration
<b>WSDL</b>	Web Services Description Language