SOMA 3.1 Information Discipline

Tool Mentor Paper on IBM's Software Products

IBM Information Server
(WebSphere Business Glossary,
WebSphere Information Analyzer)

Rational Data Architect

Version 1 – August 31, 2007

Authors:
Guenter Sauter (gsauter@us.ibm.com)
Peter Worcester (pworcest@us.ibm.com)

Abstract and Purpose

This paper describes the role that key software products from the Information Management division of IBM Software Group play in SOA, and specifically when following the Service Oriented Modeling and Architecture (SOMA) method. We will focus on products that implement the activities of the SOMA 3.1 Information Discipline (see related White Paper, Reference [1]).

The purpose of this paper is demonstrate to an architect how some tools from IBM Software Group can be used in an SOMA engagement and how they relate to the activities of the SOMA 3.1 Information Discipline. The paper describes the key features of the product that relate to the SOMA activities and refers to detailed product manuals rather than explaining how step-by-step how to install and use the product in detail.

There is a variety of products and assets from IBM Software Group Information Management that help to implement Information as a Service. We will focus in this paper on some of the SOA design related tools, and in particular WebSphere Business Glossary, Rational Data Architect, WebSphere Information Analyzer, and the underlying metadata management platform from IBM Information Server.

Acknowledgements

We would like to acknowledge the contributions from Anson Kokkat, David McCarty, Brian Byrne and John Kling that helped significantly to improve this paper.

Contents

1. Introduction & Overview

We assume in this paper that the reader is familiar with the general concepts of a service oriented architecture (SOA), IBM's service oriented modeling & architecture (SOMA) method (see Reference [1]), and more specifically the information discipline of SOMA. We define in this paper the products from IBM Software Group's Information Management division that help to implement the SOMA information discipline.

The following table summarizes some of the activities from the SOMA 3.1 information discipline and the related products that we will describe in more detail in this paper.

SOMA 3.1 Information Discipline Activity	IBM SWG Information Management Product
SOMA Identification Phase	
• Create a business glossary	WebSphere Business Glossary (see Sect. 3)
• Create a conceptual data model	Rational Data Architect (see Sect. 5)
SOMA Specification Phase	
• Create a logical data model	Rational Data Architect (see Sect. 5)
• Commence data quality analysis	WebSphere Information Analyzer (see Sect. 4)

Table 1: Mapping of SOMA Information Discipline Activities to IBM SWG Information Management Products

This list does not include IBM's Industry Models (such as IAA, IFW, etc.). The IBM Industry Models support many of the core SOMA activities and go beyond the SOMA information discipline. Brian Byrne and Ali Arsanjani have published a paper that describes the role of the Industry Models in SOMA in great detail (Ref. [2]) so that we will not describe them further in this paper.

In general, the products from IBM's Information Management division in Software Group are focused on Information On Demand. Most of the products play a key role in SOA in that they help to implement Information as a Service. Since this is a fairly broad area that includes many products that can play a role, we focus in this paper primarily on the SOA design aspects when following the SOMA method as shown in Table 1. Before we zoom into this more focused scope, we want to briefly highlight the various domains of Information as a Service and list some key products from IBM Software Group Information Management that help to implement the services.

- **Analytic Services**

  IBM has various capabilities to provide analytic insight as a service. Some of the key products are:
  - **DB2® Data Warehouse Edition** integrates and simplifies the data warehouse environment to deliver dynamic warehousing. With a single integrated software package, IBM delivers all of the capabilities needed to cost effectively consolidate, manage, deliver and analyze your business information.
  - **IBM® OmniFind™ Analytics Edition** provides a rich analysis interface for both structured data and unstructured content.
  - **IBM® Identity Resolution** helps any organization solve business problems related to recognizing true identity. It turns inconsistent, ambiguous identity and attribute data into a single resolved entity across multiple data sets, even despite deliberate attempts at misrepresentation.
  - **IBM® Relationship Resolution** provides a new level of identity awareness for corporate and governmental organizations. This groundbreaking technology finds out "Who Knows Who" to determine potential value or danger of relationships among customers, employees, vendors and other external forces.

  - **IBM® Anonymous Resolution** enables data sharing that protects the privacy of customers, employees, partners and citizens. It enables multiple organizations to compare proprietary data in a manner that identifies relationships and develops leads but never exposes sensitive data values.

- **Master Data Services**

  IBM Multiform Master Data Management manages master data domains (customers, accounts, products) that have a significant impact on the most important business processes and realizes the promise of SOA.
  - **WebSphere® Customer Center** provides real-time, transactional customer data integration. It helps organizations keep a single, complete and accurate record of their customers across the enterprise.
  - **WebSphere® Product Center** is a product information management solution for building a consistent central repository. It links product, location, trading partner, organization and terms of trade information, which is typically scattered throughout the enterprise.
  - **IBM® WebSphere® RFID Information Center** securely manages large volumes of serialized product data between trading partners. The ePedigree feature creates an EPCIS-compliant electronic certificate of authenticity for every uniquely serialized drug passing through the supply chain.

- **Information Integration Services**

  The IBM Information Server is the strategic product to provide information integration services and is described in more detail in Sect. 2.

- **Content Services**

  IBM software for enterprise content management integrates and delivers critical business information that offers new business value, on demand. Enterprise content management software and solutions support multiple information types - such as images, documents, e-mail, and e-records - and provide the appropriate content, based on user intent and relevancy. The IBM Enterprise Content Management portfolio is designed to help transform business with improved productivity and streamlined compliance.
  - **FileNet P8** provides a reliable, scalable and highly available enterprise platform for ECM and BPM. It enables you to streamline and automate business processes, access and manage all forms of content, and automate records management to help meet compliance needs.
  - **IBM® OmniFind Enterprise Edition** powers secure intranets, corporate public Web sites, and information extraction applications. OmniFind™ Enterprise Edition delivers high-quality, scalable and secure enterprise search to maximize the value of corporate information.
  - **DB2® CommonStore for Exchange Server** manages e-mail archiving and retrieval.
  - **IBM® DB2® Content Manager** manages all types of digitized content across multiple platforms, databases and applications. Built on a multi-tier, distributed architecture, it provides the scalability to grow from a single department to a geographically dispersed enterprise.
  - **DB2® Content Manager OnDemand for Multiplatforms** provides instant access to bills and invoices, supports customer service.
  - **IBM® DB2® Document Manager** provides a secure and robust platform to manage the complete lifecycle of business documents.
  - **Records Manager** is a records management engine and infrastructure tool to enable e-records management in applications.

- **Data Services**

  The IBM Data Server products allow to manage service data and to expose data through a service interface:

  - **DB2®** 9 is the next-generation hybrid data server with optimized management of both XML and relational data.
  - **Information Management System (IMS)** is IBM's premier transaction & hierarchical database management system. The latest capabilities enable SOA exploitation, secure your investment and enable new application development.
  - **IBM® Informix data servers** are known worldwide for being exceptionally easy to manage while maintaining high availability and blazing online processing capabilities.

One of the key products that support the SOMA information discipline is the IBM Information Server (overview in Sect. 2) and in particular its components WebSphere Business Glossary (see Sect. 3), WebSphere Information Analyzer (see Sect. 4), and the metadata platform which is provided by IBM Information Server (see Sect. 2.2). Another key product is Rational Data Architect that we describe in Sect. 5.

2. IBM Information Server Platform

IBM Information Server is a platform that helps organizations derive more value from the complex, heterogeneous information spread across your systems. It provides a single layer of shared services that feed trusted enterprise information to all the people, processes and applications that need it. The IBM Information Server has various components that we will introduce in Sect.2.1. Two of those components (WebSphere Business Glossary and WebSphere Information Analyzer) implement activities in the SOMA 3.1 information discipline and are therefore described in greater detail in Sect. 3 / Sect. 4 respectively. One of the significant advantages of the IBM Information Server is the integration of all of its components through a common metadata platform that we will describe in Sect. 2.2. This metadata platform allows the various components to consistently and effectively share their artifacts within the IBM Information Server and with other critical components in SOA. In Section 2.3, we describe how the information-related metadata can be shared with various SOA related tools.

2.1 Motivation & Overall Product Scope

The IBM Information Server consists of multiple components that can be categorized according to the following overview diagram:

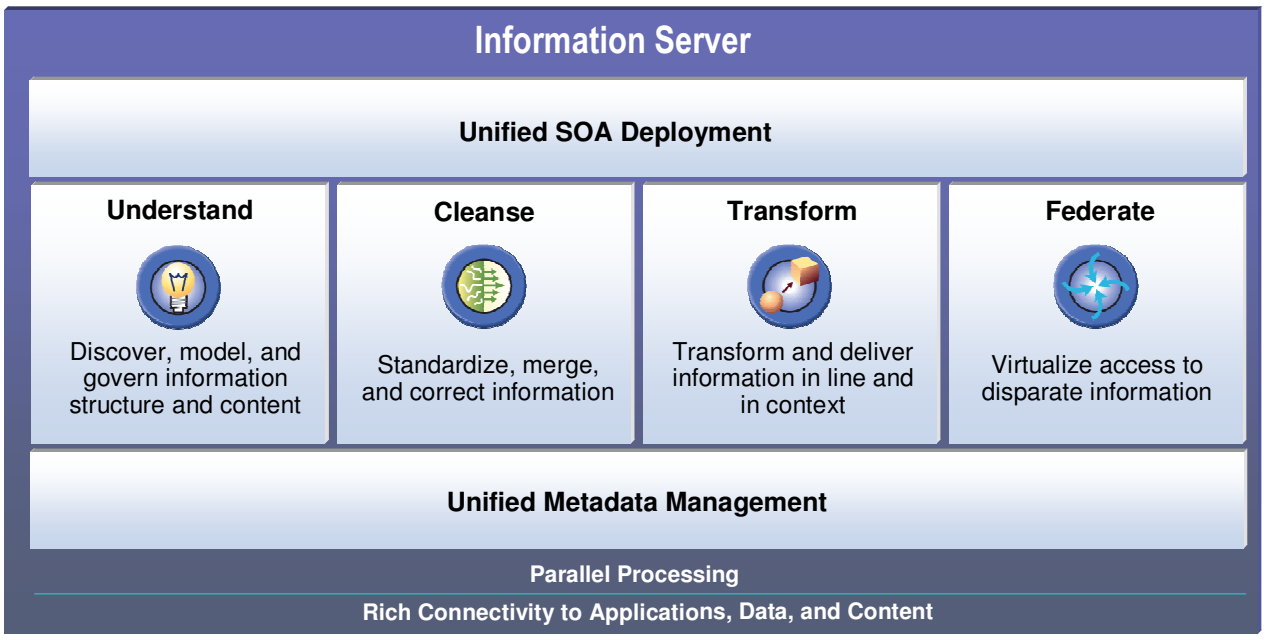


Figure 1: IBM Information Server Overview

The IBM Information Server supports the end-to-end process of integrating data from heterogeneous sources through in a scalable and comprehensive environment. The following categories of capabilities are required when integrating data:

- **Understand**

  Before we start moving data from sources to targets, we need to understand what the business needs to see in an integrated view and what sources we have and how they are structured.

  WebSphere Business Glossary creates, manages, and searches definitions of terminology. It Create and manage a controlled vocabulary that enables a common language between business and IT.. WebSphere Business Glossary is described in more detail in Sect. 3 since it implements one of the major activities in the SOMA 3.1 information discipline.

  WebSphere Information Analyzer profiles and establishes an understanding of source systems and monitors data rules. It uncovers missing, inaccurate and inconsistent data early in your data integration project lifecycle. We descirbe WebSphere Information Analyzer in more detail in Sect. 4 since it also supports a major activity of the SOMA 3.1 information discipline.

  IBM Metadata Worbench provides end-to-end metadata management, depicting the relationships between sources and consumers. It allows you to explore and analyze technical metadata about sources of data, business metadata for data meaning and usage, and operational metadata that describes what happens within the integration process itself. It visually depicts these relationships from the sources of information to the places where information is actually used—even across different tools and technologies. It outlines the complete lineage of fields from applications, reports or data warehouses back to source systems, including what types of processing were performed on them along the way. IBM Metadata Workbench allows IT staff to assess the full impact of any system change before the change is made – even when the impact spans across different tools. For example, it would show the business intelligence report fields that would need to be changed if a source table were changed. It allows project teams to manage metadata within IBM Information Server and related technologies – allowing guided ad hoc queries, search, and visual navigation through the metamodel, and allowing metadata to be manually edited where appropriate.

- **Cleanse**

  Based on the understanding of what type of information the business expects you to deliver and the understanding of your current legacy environment you may have to standardize and cleanse the information before it is transformed and delivered..

  WebSphere QualityStage standardizes and matches information across heterogeneous sources. Its industry's leading matching engine ensures that the information that runs your enterprise is based on your business rules, reflect the facts in the real world and provide an accurate view across your enterprise. It provides a single set of standardization, cleansing, matching and survivorship rules for your core business entities - executed in batch, real time, as a web service. Its seamless data flow integration brings data quality to any data integration situation.

- **Transform**

  Your legacy data does not always exist in the form and at the place as required by the business. You may need to integrated data from different sources, you may need to restructure it so that you can process and visualize it more effectively, etc.

  WebSphere DataStage cxtracts, transforms, and loads data between multiple sources and targets. It supports the collection, integration and transformation of large volumes of data, with data structures ranging from simple to highly complex.

- **Deliver**

  IBM Information Server gives our clients the ability to transform and move data with the extensive capabilties of WebSphere DataStage. It also provides capbilities to leave the data in place and to virtually pull heterogeneous data together in real-time.

  WebSphere Federation Server defines integrated views across diverse and distributed information sources, including cost-based query optimization and integrated caching. It accesses multiple diverse data and content sources in real time as if they were a single source. It provides a framework to

quickly integrate various types of data sources incl. a very broad range of databases, XML files, spreadsheets, etc.

- **Rich Connectivity**

  IBM Information Server provides direct, native access to relevant sources through shared connectivity services. It allows to capture changed data and event-based publishing of data. These connectivity products can be used standalone to support specific application requirements, or in conjunction with the other products in the platform to provide composite solutions.

- **Parallel Processing**

  IBM Information Server is based on a highly scalable parallel engine to meet the needs of highly complex data integration projects.

- **Unified Metadata Management**

  All of the tools within the IBM Information Server platform are based on a unified metadata management approach that we will describe in Sect. 2.2.

- **Unified SOA Deployment**

  IBM Information Server is applied in many customer engagements in a traditional data integration context to support scenarios such as application migration, application consolidation, decision support, and master data management. Beyond this traditional scope, IBM Information Server components – in particular WebSphere DataStage, WebSphere QualityStage, and WebSphere Federation Server – can also expose their functionality as a service.

  WebSphere Information Services Director allows information access and integration processes to be published as reusable services in a service oriented architecture. It enables developers to take data integration logic built using IBM Information Server and publish it as an "always on" service in a SOA. It provides the ability to publish services without coding. The same service supports multiple protocol bindings: SOAP/HTTP (Web services for ubiquitous support), and Enterprise Java Beans (EJB, for high-speed direct Java integration). It defines services as true business objects, deployed natively in your application server, completely hiding the implementation complexity from the service consumer. It provides a resilient environment that supports fault tolerance, load balancing and true parallel execution for high availability. It is built on a pure J2EE architecture, that provides a secure and resilient framework for hosting data integration services and dispatching requests to the IBM Information Server, WebSphere Process Server, or other applications and databases. It maintains a directory of available services, directly linked into the metadata infrastructure underlying IBM Information Server, to enable centralized management of data integration services. WebSphere Information Services Director is built to provide a SOA for data integration that leverages and reinforces the open standards activities of organizations like WS-I, OASIS, W3C, and the Java Community Process.

2.2 Unified Metadata Management Platform

The metadata platform plays an extremely important role in IBM Information Server in that it provides a repository and an interface so that the IBM Information Server (and third party) components can access, maintain and share their artifacts. As described above, the IBM Information Server components cover related

tasks that assist in integrating information. The following diagram illustrates how the components are integrated in an example scenario.

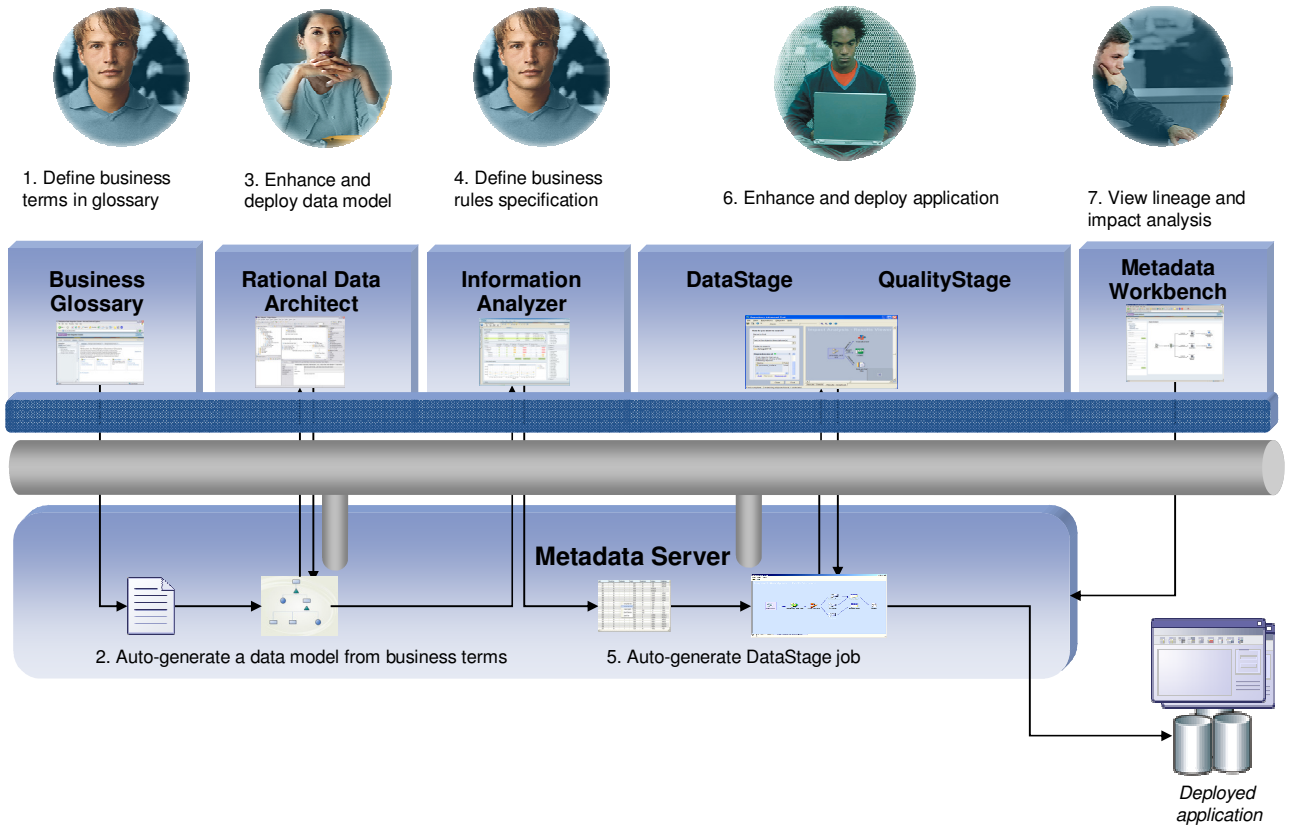


Figure 2: Metadata Flow between IBM Information Server Components

Early in the project, project members from business and IT need to agree on the terminology that they are using in the project. The agreement and definition of terms is often the first step in a project. Some of the terms will be modeled as entities and attributes in the canonical data model. Therefore, the metadata platform needs to facilitate the sharing of the term definition from WebSphere Business Glossary with the data modeling tool Rational Data Architect. This is shown below:

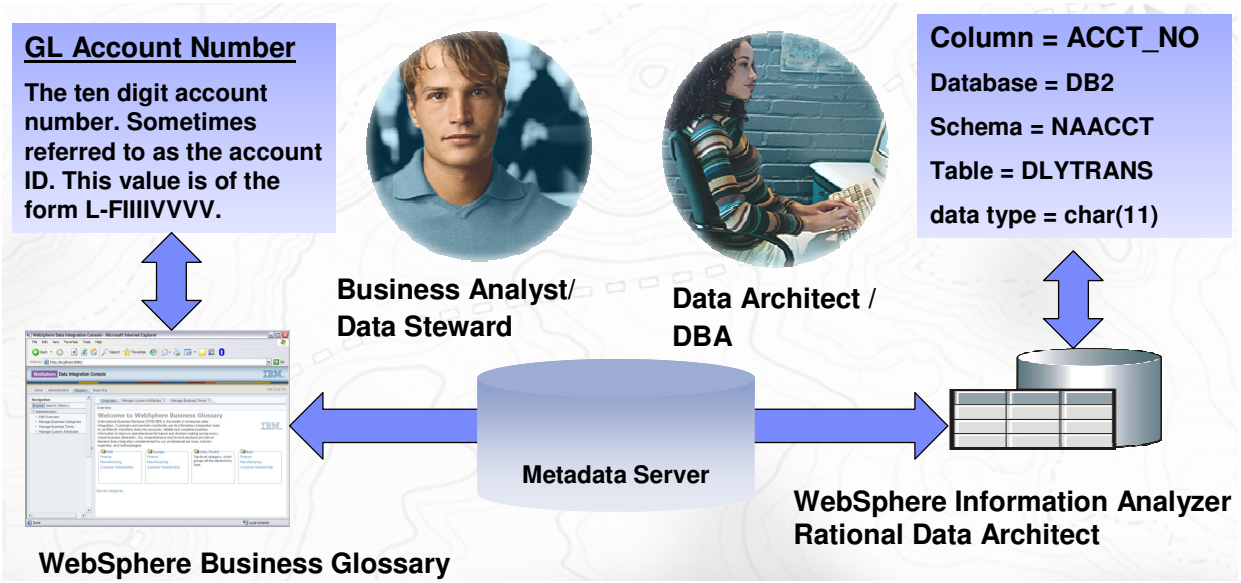


Figure 3: Metadata Exchange between WBG and RDA/WIA

When we analyze the data quality, we need to understand the meaning of the various terms and canonical data model elements in order to determine the consistency of this target environment with the data sources and across the data sources. Access to business glossary and data model artifacts are critical when defining the business rules to address any data quality concerns in WebSphere Information Analyzer.

If the solution requires cleansing or integrating data, the previously developed artifacts will then help to design the corresponding data cleansing and transformation rules that can then be deployed (in WebSphere QualityStage or WebSphere DataStage respectively). After it is deployed, the end-to-end lineage of data can then be visualized and an impact analysis can be studied (in IBM Metadata Workbench). This again requires access to various artifacts from the previous phases.

The metadata management platform in IBM Information Server provides a repository and framework for the various tools to access, maintain, and share their artifacts with other IBM Information Server components and third party tools which are connected to the platform as shown in the diagram below.

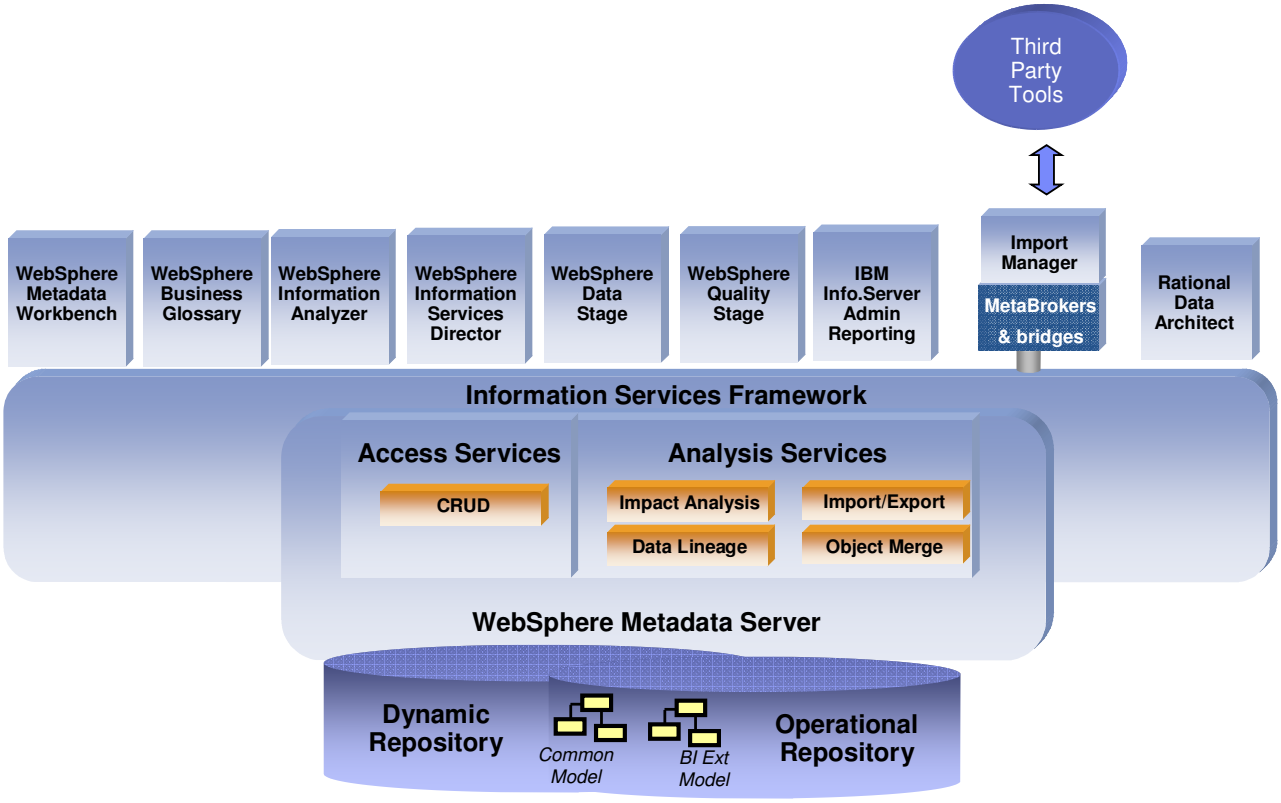


Figure 4: The Unified Metadata Management in IBM Information Server

The unified metadata management platform consists of the following components:

- **Access Services**

  The access services are part of the overall Information Services Framework of IBM Information Server. They provide basic create-read-update-delete (CRUD) functionality for the metadata.

- **Analysis Services**

  The analysis services are also part of the overall Information Services Framework of IBM Information Server. They provide functionality for impact analysis, data lineage (static and dynamic), import/export of metadata, and object merge of metadata. This functionality can be leveraged by any of the IBM Information Server components through the Information Services Framework. One of the primary tools for these metadata services is the IBM Metadata Workbench.

- **Dynamic Repository**

  The dynamic repository of the unified metadata management platform is the primary workspace for the design tools. It supports multiple clients and has a locking facility to coordinate updates of shared elements. The default is optimistic locking but pessimistic locking is also available. The Eclipse modeling framework (EMF) based models are mapped to a relational store for persistence.

- **Operational Repository**

  The operational repository of the unified metadata management platform in supports the operational needs of the IBM Information Server components. The operational metadata is modeled and linkable with the design metadata. This type of metadata is rather time-stamped than versioned metadata.

The repository is designed to support relatively high volume requirements such as 10k-20k transactions per minute.

The **common model** is a clean superset of the metadata used by IBM Information Server components today. It allows for a common vocabulary and semantic sharing of metadata across the IBM Information Server components.

2.3 Metadata Exchange With SOA

Due to the fact that there are different types of metadata (data related, service related, application-related, system-related) that require very specific and different tools to manage that metadata, IBM does not have a single metadata repository and a single metadata tool to manage those different metadata types. Rather, there are domain-specific repositories and tools that are integrated together. The two domains of interest in this paper are the service-related and data-related metadata. We have described IBM Information Server with its unified metadata management platform above which provides the metadata repository and tooling for data-related metadata. We will now discuss how IBM Information Server is integrated with the service-related metadata tools and repository.

In the tool integration scenarios below, you will see that various SOA tools such as WebSphere Integration Developer (Sect. 2.3.2) and WebSphere Portlet Factory (Sect. 2.3.3) can access metadata from IBM Information Server's metadata management platform. The following metadata is available for the SOA tools independent of the type of an information service:

- service,
- contact,
- created by,
- version,
- creation date,
- last modified,
- modified by,
- service description page,
- operation name,
- operation description,
- operation type

The additional metadata is available for an information service that has been implemented based on DB2 or WebSphere Federation Server:

- subtype (e.g. SQL statement),
- operation (e.g. query),
- the query itself,
- database,

- port number,
- server name

The additional metadata is available for an information service that has been implemented based on WebSphere QualityStage and WebSphere DataStage:

- complete report of WebSphere QualityStage / WebSphere DataStage
- the associated report of WebSphere QualityStage / WebSphere DataStage
- the latest report of WebSphere QualityStage / WebSphere DataStage with date.

2.3.1 WebSphere Service Registry and Repository

The WebSphere Service Registry and Repository (WSRR) is IBM's repository and tool to manage service-related metadata. As its product name suggests, it provides a service registry and repository so that service providers can register their services and service consumers can find services and access the service definition.

The strategic service provider of IBM Information Server is **WebSphere Information Services Director** (WISD). This product takes a defined operation in IBM Information Server and exposes that operation as a service. The following three IBM Information Server components can provide such operations:

- WebSphere QualityStage defines cleansing rules that standardize input data and remove duplicates according to specified rules and return normalized data as an output. For example, an address information in any format can be provided as input and a WebSphere QualityStage operation would standardize the address and return it as the output.
- WebSphere DataStage defines transformation operations on input data which can be data in almost any source database. It stores the result either in another database or returns it as the result of the operation.
- WebSphere Federation Server defines a virtual view over heterogeneous information that is stored in multiple databases and returns the integrated information as its result.

The three types of operations listed above are stored in the metadata repository. WISD accesses the repository and allows the user to select the operation that needs to be exposed as a service. The user can then choose among various bindings when specifying the service. Once the service is specified, it can be registered in WSRR from the WISD user interface as shown in the diagram below.

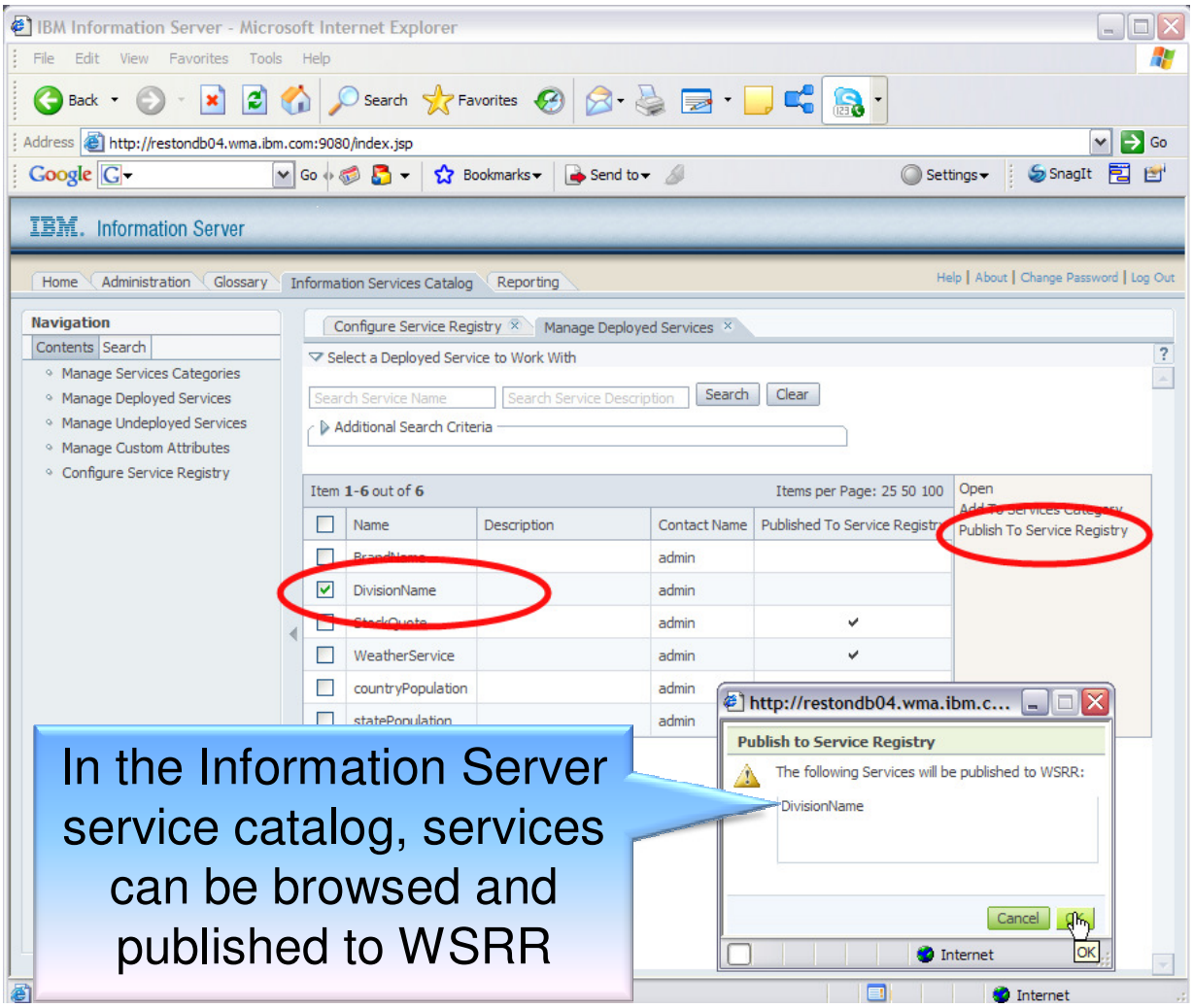


Figure 5: IBM Information Server (WISD) Publishing a Service to WSRR

As shown in the screenshot above, any deployed service in WISD can be published to WSRR. As a result, the WSDL definition of the WISD service will be published in WSRR. To be exact, two WSDL specifications are published: one for the service interface and the binding details and the other for the service port information.

The advantage of the unified metadata management platform in IBM Information Server goes beyond to simply publish a WSDL service definition into WSRR which is supported by almost any service-enabled product from any vendor. When service consumers want to understand more details about information services that are deployed through WISD, they can leverage the metadata platform from IBM Information Server as explained in the following two sections.

2.3.2 WebSphere Integration Developer

The IBM WebSphere Integration Developer (WID) 6.0.2 has an additional plugin (IBM Information Server Plug-In for WebSphere Integration Developer) so that it can incorporate information services as defined by WISD.

The first integration point between WID and WISD / unified metadata management platform of IBM Information Server is when the user wants to add an information service as a BPEL activity as shown in the screenshot below:

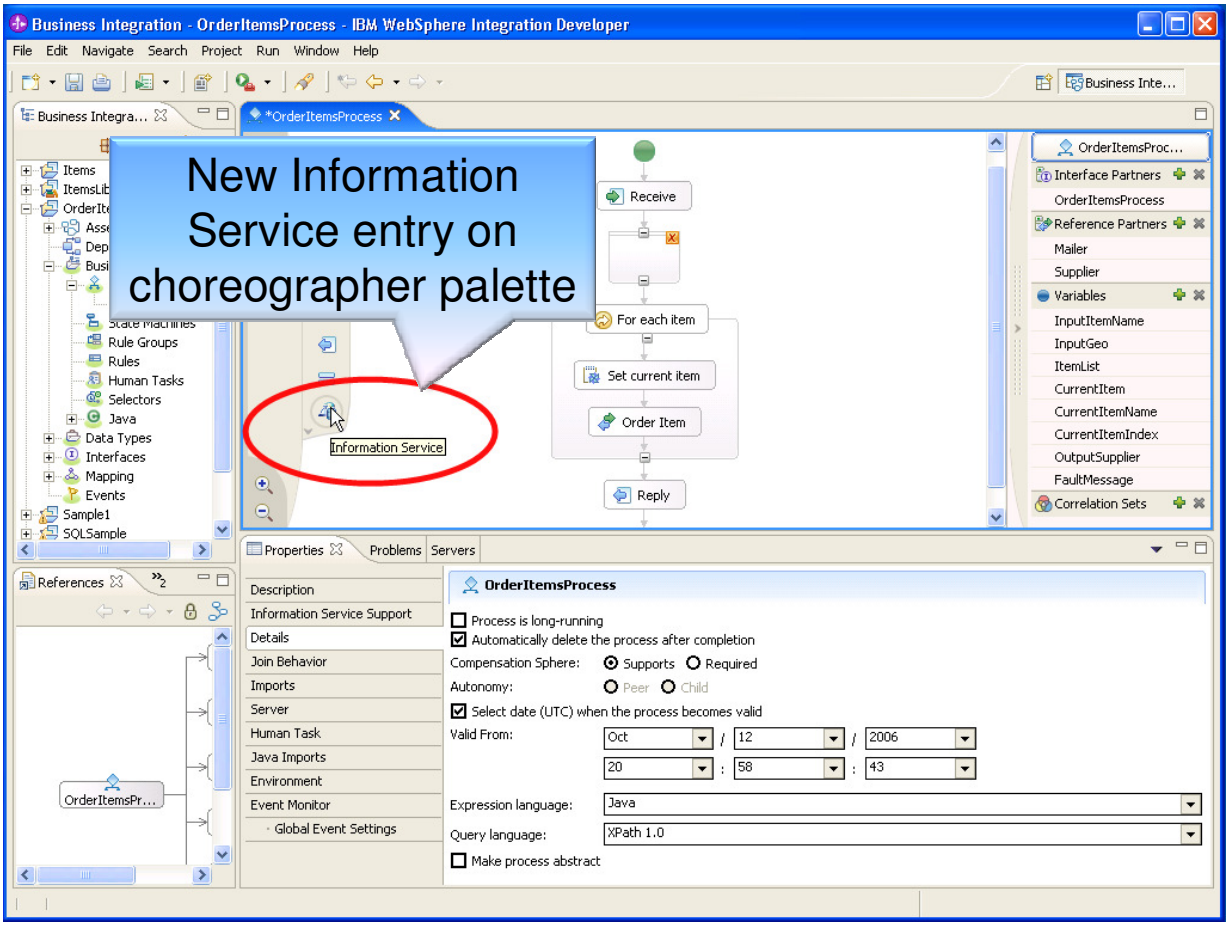


Figure 6: Adding an Information Service as a BPEL Activity in WID

As shown in the screenshot above, the choreographer palette includes an element to incorporate an information services as a BPEL activity. Once the information services is added to the BPEL flow, the user can then browse for a specific operation to implement the information service. At this point, WID establishes a connection to the unified metadata management platform of IBM Information Server and more specifically the Information Services Framework that was described earlier.

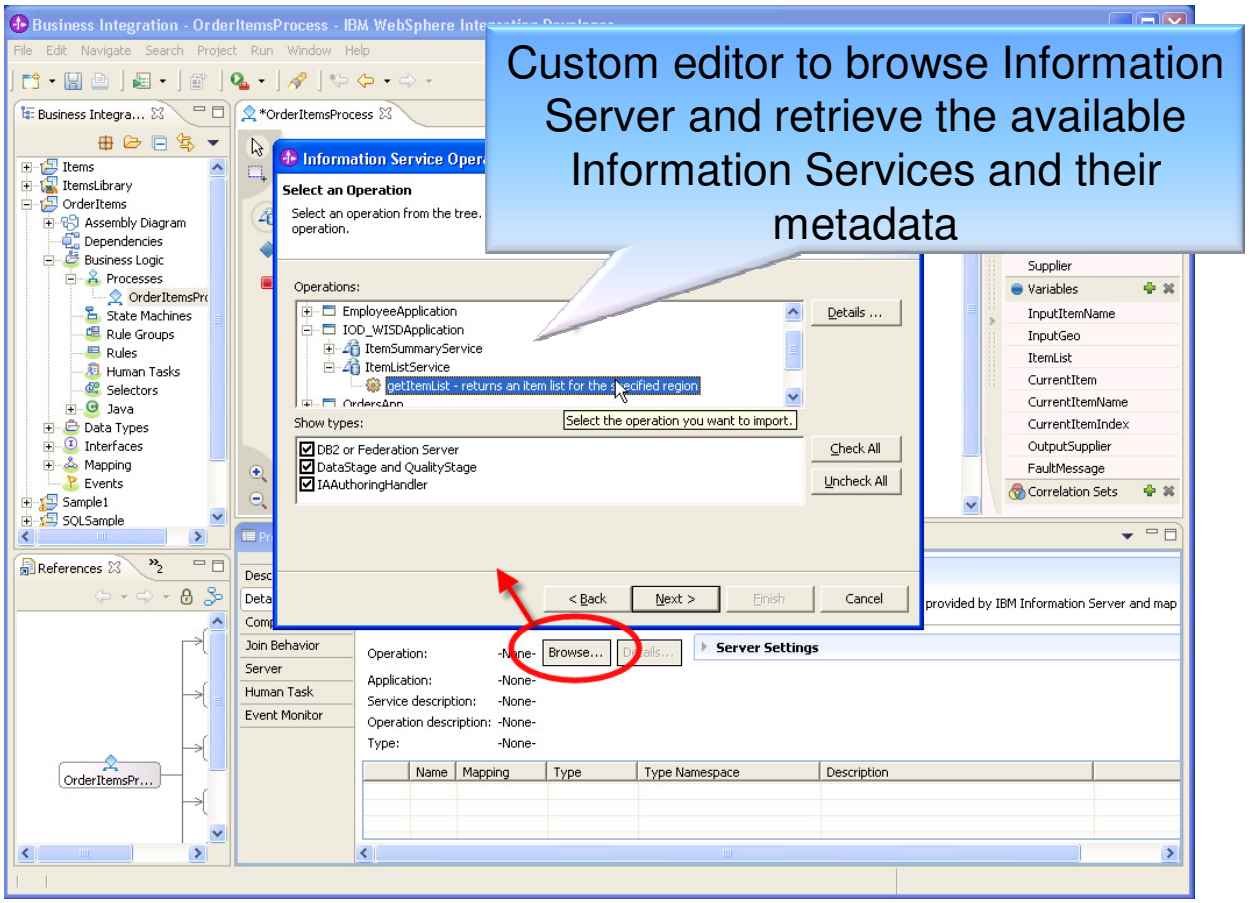


Figure 7: WID Accessing WISD services

In the example above, we want to incorporate the `getItemList` operation in the BPEL flow. As shown in the screenshot, we can select the service that we need, but the standard display offers only limited information about the service implementation and what this operation `getItemList` really does. If the WID user is not the same person as the WISD developer he would probably prefer to have access to more metadata about the information service. IBM Information Server allows accessing the actual implementation of the service as shown in the screenshot below.

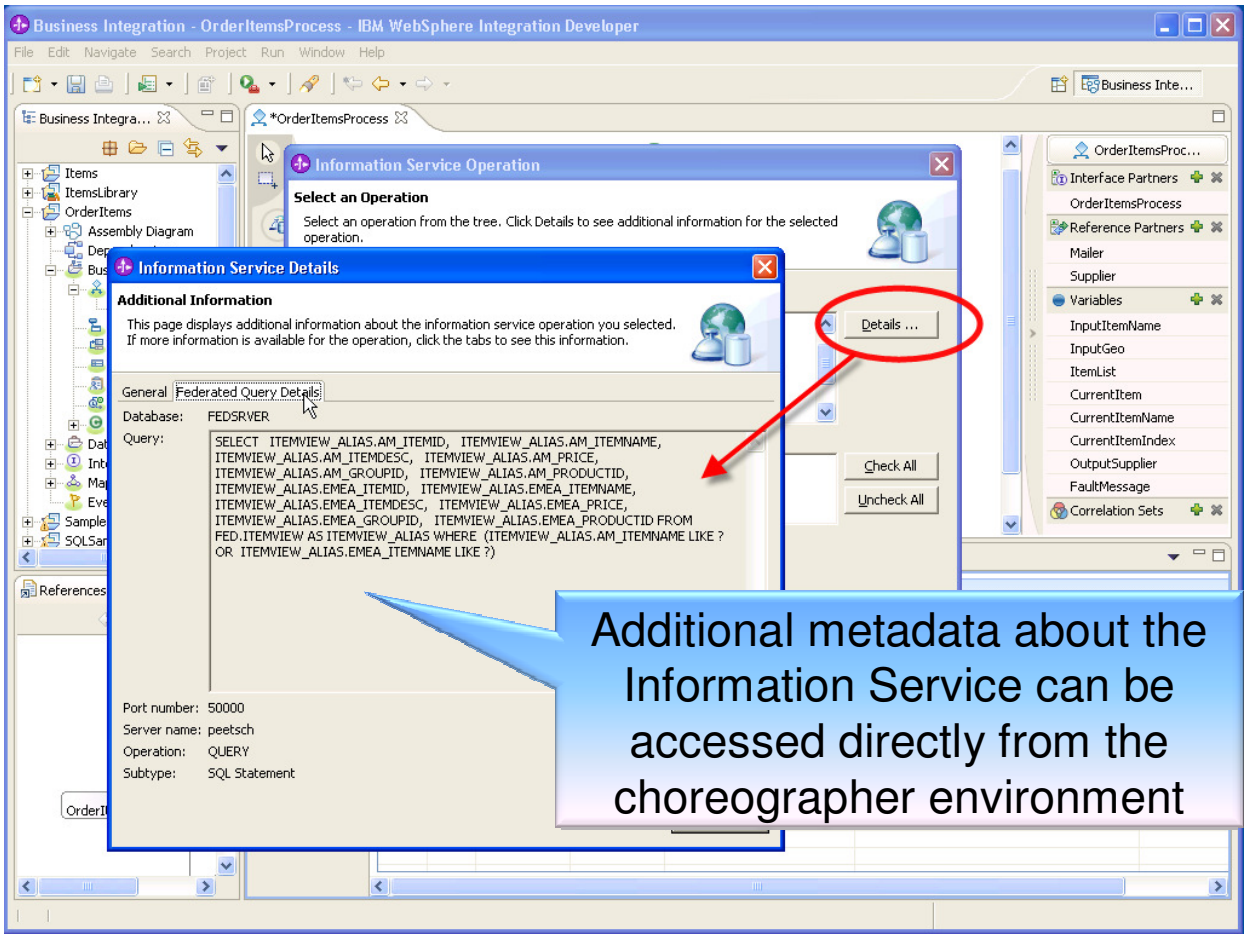


Figure 8: WID Accessing WISD Service Realization Metadata

In this particular example, the `getItemList` operation is actually a federated query. When the user views the details of the operation, he can access the federated query that implements the operation.

The same metadata support from IBM Information Server is available for mediation in WID. When the WID user wants to select an enterprise service resource adapter (SCA binding – enterprise service discovery), the user can access the same metadata as shown above when integrating an information service as a BPEL activity.

2.3.3 WebSphere Portlet Factory

In WebSphere Portlet Factory (WPF), we follow a similar model that we have introduced in the previous section when we incorporated an information services into a BPEL flow using WID. When we design a portal application using WPF, we may need to incorporate an information services that has been defined in WISD. When the WPF user develops a portlet, the developer will use the WPF builder palette to browse for available services. One of the available builder types is an information service call. When the user has selected this builder type and is specifying the service, he has the same service lookup functionality to WISD that has been described above.

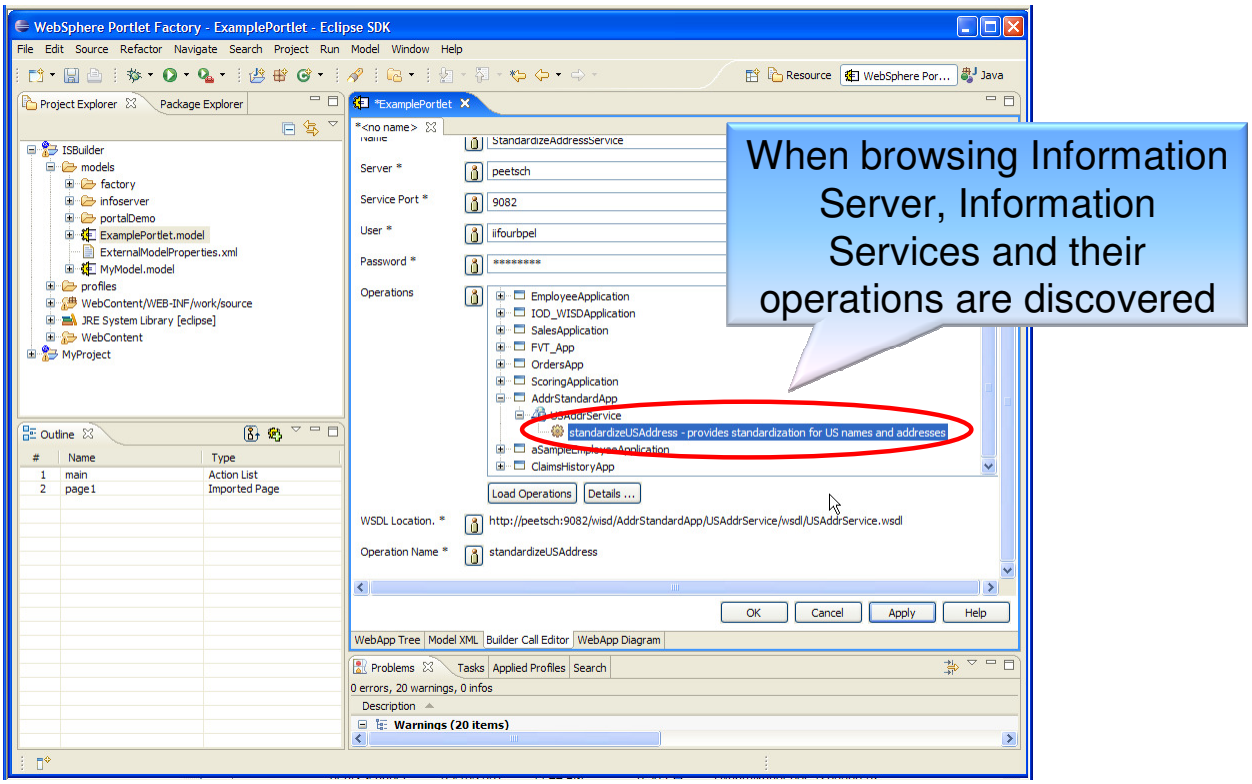


Figure 9: WPF Accessing WISD Services

As shown in the diagram above, the WPF user can now select a service that has been developed in WISD. If the WPF user wants to access more detailed metadata information about the service, he can accesss the same metadata as described in the previous chapter by following the details button below the services list.

3. WebSphere Business Glossary

A business glossary sometimes referred to as a data dictionary is the artifact that will define and contain the agreed up definitions of the terms and data associated with an initiative. Depending on the extent and type of engagement the business glossary will either define the context of a term within a silo, an information domain, or preferable across the enterprise.

Simply put the business glossary is the formal contract between the producers and consumers of information across the enterprise. It is intended to be the artifact or reference that will allow anyone to determine the meaning, type and context of any term and in particular any business data element used in an initiative. Too often we see significant lack of formal definition around data entities even within a single system. Different interpretations of the same term increase the risk of a successful project delivery. There are often embedded business rules that make the data inaccurate out of the context in which it is originally used in. They could be embedded rules in the programs that use the data or additional reference data needed to give the entity context. These are the kinds of things that need to be identified

It is certainly very common for different department or lines of business (LOB) to have different semantic contexts for what would seem to be the same term. To give a simplified example, take the element “address”, to distribution this is likely the address to “ship to”, to accounting this most likely will be a “bill to” address, to Sales and Marketing this will likely be a “call on” or “contact” address. This is a very simplified example and is usually dealt with by some prefix or having three different address fields. Regardless there needs to be a way to document and identify which “type” of address we are dealing with and what each one means. Another example is the meaning of a “member” to a healthcare company. Different organizations may define the meaning differently. Maybe the sales department is mostly interested in members for which the contracts can be renewed. The call center group might be mostly interested in current members. As a consequence, a services that needs to return “members” may have different results if not defined appropriately. This example demonstrates that the success of SOA deployments can depend upon a common definition of the term. If the organization cannot agree on what `getMember` needs to return, the service cannot be implemented successfully. The earlier those challenges become clear, the sooner the risk can be mitigated.

The business glossary defines the language of the business and by extension the language of the project. Therefore, care needs to be exercised that the terms defined in the business glossary are fully qualified and that specific descriptive definitions are provided. To the extent possible, a definition that applies enterprise-wide should be crafted. Different departments may use a term differently; all those definitions should be captured and associated with their appropriate contexts (department).

When an organization builds an enterprise-wide business glossary, it may include both semantics and representational definitions for terms. The semantic components focus on creating precise meaning of the terms. Representation definitions include how terms are represented in an IT system such as an integer, string or date format (see data type). Business glossaries are one step along a pathway of creating precise semantic definitions for an organization.

3.1 Getting Started

WebSphere Business Glossary (WBG) is installed on a server and is accessed through a browser interface. There are 3 roles available for WBG:

- the “Administrator” role,
- the “Author” role, and
- the “User” role.

People with the **Business Glossary User role** can examine the metadata assets in the metadata repository, including the terms and the categories that contain terms. Users can communicate their concerns or information about particular objects to the glossary administrator.

Users can perform the following types of tasks:

- Browsing the structure of categories and terms
- Searching the metadata repository for categories, terms, and other objects
- Exploring the attributes and relationships of all objects in the metadata repository
- Sending feedback to the administrator

Users who are assigned the **Business Glossary Author role** can create and edit terms and categories and use terms to classify objects. The author role is assigned to users who manage categories and terms and who decide how objects are classified and who the stewards are for specific objects.

Authors can perform all tasks that are associated with the Business Glossary User role.

In addition, authors can perform the following types of tasks:

- Creating and editing a hierarchy of categories that contain terms that are used by your enterprise
- Classifying objects in the metadata repository by using terms
- Setting stewardship for objects in the metadata repository
- Uploading terms and categories to the metadata repository
- Specifying values for custom attributes

Users who are assigned the **Business Glossary Administrator role** can set up and administer the glossary so that other users can find and analyze the information they need. Glossary administrators can perform all the tasks that are associated with the Business Glossary Author and Business Glossary User roles.

They can create, edit, and delete terms and categories. They can associate terms and stewards with objects. They can browse the metadata repository and create annotations. They can perform any other glossary task.

In addition, the following tasks can be performed only by people who are assigned the Business Glossary Administrator role:

- Customizing the Overview page of IBM WebSphere Business Glossary to provide users with an starting point that is specific to your enterprise, and that lets them easily navigate the hierarchy of categories

- Setting application options
- Designating users and groups as stewards, and deleting the steward relationship from a user or group.
- Creating, editing, and deleting custom attributes
- Editing and deleting annotations that were created by others.
- Deleting terms and categories that were created by others.

3.2 Motivation & Overall Product Scope

WebSphere Business Glossary (WBG) creates and manages a controlled vocabulary that enables a common language between business and IT. It supports the following key functionalities:

- **Manage Business Terms and Categories (see Section 3.3.1)**

  WebSphere Business Glossary provides a dedicated, web-based user interface for creating, managing, and sharing a controlled vocabulary. Terms represent the major information concepts in your enterprise. Categories are used to organize these terms into hierarchies.

- **Manage Stewardship (see Section 3.3.2)**

  Stewards are people or organizations with responsibility for a given information asset. Using the WebSphere Business Glossary functionality of IBM Information Server, administrators can import stewards profiles from external sources, create and edit profiles in the web interface, and create relationships of responsibility between stewards and business terms or any of the artifacts managed by WebSphere Metadata Server.

- **Customize and extend (see Section 3.3.3)**

  Needs around business metadata tend to differ from one enterprise to the next. For this reason, there is no "one size fits all" meta-model. In addition to being able to customize the entry page to the application, administrators can extend the application with custom attributes on both business categories and business terms.

- **Collaborate (see Section 3.3.4)**

  It is not enough to simply document business metadata. This information must be alive in the enterprise, with open access to all. WebSphere Business Glossary provides a collaborative environment in which users can organically grow this important information asset. There are two aspects to this collaboration, the first is collaboration within the tool itself, which is done with notes and annotations as well as subscriuptions to topics of interest. The other kind of collaboration is WBG and other tools like WebSphere Information Analyzer. This is primarily accomplished through the unified metadata management platform of IBM Information Server (see Sect. 2.2).

Why would we propose using a tool instead of one of the manual methods described above? The simple answer is because it does more than just storing the data that is collected. It allows the data/metadata to be collected or imported from other applications like Rational Data Architect (RDA) saving the manual time and labor having to input this metadata by hand. It also allows for simple and direct natural language query of the metadata as well as it being a collaborative tool so that when a term is updated or refined people who

"subscribe" to that data can be notified of the changes. Any text processing tool can be used to document a term in a simple table form. And the documentation of terms is certainly an important aspect. However, the true value of a Business Glossary can only be accomplished when all relevant users (i.e. architects, developers, etc.) follow that terminology.

The collaborative supports in WBG through its web-based user interface as well as its metadata sharing capabilities are some of the primary reasons to use this tool instead of a manual approach.

3.3 Using WebSphere Business Glossary

Below in Figure 10 is a screen shot of the initial screen of WBG. Several tabs are available that group the functions that are available for a particular role as described above.

The page that you will primarily focus on is the "Glossary" page, where we will manage the overview, business terms and custom attributes for those terms.

On the left hand side is the navigation pane where the user will be able to Browse, Search and manage & maintain the history associated with the business term in question.

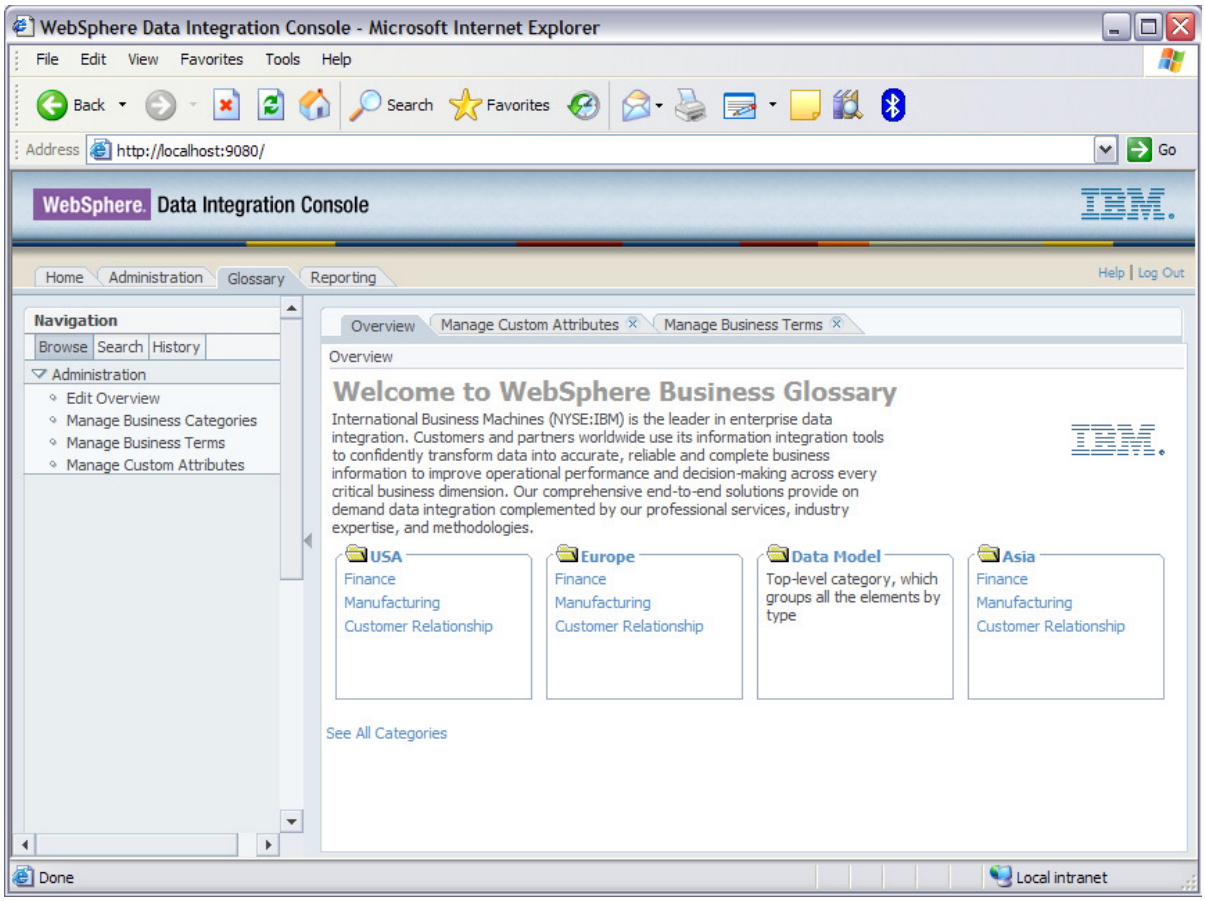


Figure 10: Home Page of WebSphere Business Glossary

WBG is primarily used to collect metadata and the business meaning of a business term, not only about the "technical" details of the data. Take an example, in opening an account, there is going to be a term called "Account Number". Traditionally the information collected would only be something along the lines of what database the account number would be stored in, what schema, table and column and then what data type (VARCHAR(11) for example):

- Name of the term such as GL Account Number
- Description of the term such as The ten digit account number. Sometimes referred to as the account ID. This value is of the form L-FIIIIVVVV
- Steward for this term. A Steward is considered the "Owner" / SME for this data entity.

WebSphere Business Glossary is a tool that enables business users to document the business meaning of these assets and attach to the artifacts that are collected & created by the other tools involved in information management.

What is the Customer objective around a tool such as WBG?

A typical objective for a customer is to document & share critical business aspects of data integration artifacts – especially the data itself.

- There is a core set of information that all customers require
  - Standard names & definitions for date items (terms)

  - Organized as hierarchies
  - With descriptions, examples, abbreviations, and stewardship information
- Additionally, each customer has a unique set of business metadata that is critical to their organization
  - Calculations
  - Policies
  - Validation
  - Authority
  - Security
  - Sensitivity

WebSphere Business Glossary provides a controlled vocabulary. A controlled vocabulary is a glossary that includes agreed upon standard word and terms, reusable labels/tags. It also provides a semantic layer between the user's natural language and the highly structured information in the repository. It typically also includes the notion of ownership/stewardship/responsibility.

After this brief overview of WebSphere Business Glossary, we describe the core functions in more details in the following sections and also present in a walkthrough how the tool is used in a typical use case scenario.

3.3.1 Manage Terms and Categories

Administrators and authors create a logical structure of categories, terms, and classified objects. The choice of procedures that you use when you build the glossary depends on whether you are primarily creating new categories and terms or using categories and terms that already exist in the unified metadata repository of IBM Information Server. In either case, you should map out the desired structure before you build it in the glossary. When planning the structure, consider the following questions:

- What categories do you need?
- Do you have existing categories and terms that can be imported or uploaded into the repository?
- Which categories are the top-level categories, and which are subcategories?
- Which categories do you want to present to the user on the Overview page as the starting point for browsing the metadata repository? These categories do not need to be the same as the top-level categories
- What terms do you need?
- Which categories contain which terms?
- Which categories reference terms that they do not contain?
- Which terms are related to other terms?
- Which terms are synonyms of other terms?

If you answer these questions in detail before you create the glossary structure, you can build a structure that is simple for users to understand and that supports your enterprise goals. You can decide in advance which objects are classified by which terms, or you can wait until after you build the glossary structure to make that decision. If you are building a glossary structure by creating new categories and terms, instead of uploading or importing existing categories and terms into the repository, you must first create the categories, then create the terms, and then edit the categories and terms to set the relationships between them.

Administrators and authors can use terms to classify objects in the metadata repository. A **term** is a word or phrase that can be used to classify and group objects in the metadata repository. For example, you might use the term Africa Sales to classify some of the tables and columns in the metadata repository, and the term Europe Sales to classify other tables and columns.

If your metadata repository includes some different terms that mean the same thing, you can designate such terms as synonyms. If two terms are not synonyms, but are related in some other way that is important, you can designate them as related terms. You can specify which term of a group of terms is the preferred term, and which terms to replace with other terms. You can also specify standard abbreviations of the term.

When you create or edit a term, you can do any of the following actions:

- Specifying term properties
- Specifying term relationships, including synonym terms, related terms, and classified objects
- Editing values for any custom attributes that apply to the term

Administrators and authors can also upload files of categories and terms to the metadata repository, and then specify additional properties and relationships

3.3.1.1 Term properties

You can specify the following properties of a term:

- **Name**
  Term names must be unique. If you attempt to create a term with the same name as an existing term, you are prompted to use a different name.
  Names of terms must start and end with a character that is not a space.

- **Parent Category**
  The category that contains the term. A term must have one and only one parent category.

- **Short Description** (optional)
  Short descriptions are important because they can help uniquely identify a category in a list of other terms with similar names. The text should be no longer than one or two lines. Short descriptions are used in many searches and are displayed in lists of objects.

- **Long Description** (optional)
  They can contain additional information that might be of interest to users who are browsing the term or administrators and authors who are setting its relationships to other objects.

- **Usage** (optional)
  Information about how to use the term, and any business rules that govern its use.

- **Example** (optional)**.**
  An example of how the term is used, or a typical sample value.

- **Status**
  The approval status of the term within the organization. Status has one of the following values:
  - *Candidate*: The default value for new terms.
  - *Accepted*: Accepted by an administrator for general use.

  - *Standard*: Considered the standard for definitions of its type.
  - *Deprecated*: Should no longer be used.

- **Type**

  The classification of a term based on its use for metadata naming standardization. Type has one of the following values:
  - *None*: The type has not been declared. This is the default value.
  - *Primary*: The term describes a major enterprise concept such as a customer or an employee. Primary terms are usually nouns or noun phrases that form the basis for naming objects in data models.
  - *Secondary*: The term identifies a secondary distinguishing characteristic of a business concept, such as an identification number. Secondary terms are usually nouns or noun phrases that form the basis for naming an attribute of an object.

- **Is Modifier**

  Describes whether or not the primary purpose of the term is to provide descriptive information about an object. Is Modifier has the following possible values:
  - *Yes*: The primary purpose of the term is to provide descriptive information about an object.
  - *No*: The primary purpose of the term is to identify distinguishing characteristics of an object. No is the default value.

- **Preferred Synonym**

  The term is the preferred term in a group of synonym terms. Terms with the deprecated status cannot be preferred terms

- **Abbreviations** (optional)

  One or two standard abbreviations of the term.

3.3.1.2 Term relationships

You can specify that terms have relationships to the following types of objects:

- **Steward**

  The person or group that is responsible for the term. A term can have only one steward.

- **Related Terms**

  Terms that are related in some way to the term in question. This relationship can be used for specifying that relationship. The relationship is not symmetrical. If you specify that term A has term B as a related term, that does not imply that term B has term A as a related term. A term can have multiple related terms.

- **Synonyms**

  Terms that have the same meaning. A term can have multiple synonym terms. The relationship is symmetrical and transitive. If term A is a synonym of term B, and term B is a synonym of term C, each term is a synonym of the others.

- **Classified objects**

  Objects that the term classifies. A term can classify multiple objects in the repository. An object can be classified by multiple terms.

3.3.1.3 Custom attributes

Administrators and authors can specify values for custom attributes for any individual term or category that the attributes apply to. You specify these values when you create or edit the term or category

3.3.1.4 Example:

In the simplest form let's say we create a business term for a user "Account" we define the term "Account" and some of its attributes. Below is an example of a user defined attribute of "Account" called "Account Number"

Account Number: The ten digit account number, sometimes referred to as the account ID. This value is of the form L-FIIIIVVVV. Owned By: Controller's Office, Mary Smith X3256 Synonyms: Account ID See Also: Account Type

3.3.2 Manage Stewardships

Stewards are users or groups that have responsibility for one or more metadata objects in the repository. Business Glossary administrators can designate that a user or group in the metadata repository is a steward. Administrators and authors can then specify that the steward is responsible for one or more metadata objects. A steward is typically assigned to the objects that the user or group is responsible for managing or is the appropriate contact for.

When you view the browse page for an object that has a steward, a link to the steward is displayed. The link leads to contact information, which includes e-mail address and phone number.

You can assign responsibility for multiple objects when you designate a new steward or when you edit a steward on the Manage Stewards page. You can also assign an object to a steward from the Tasks list on the browse page of the object, or on the browse page of a user or group who is a steward. In addition, you can assign responsibility for a particular category or term to a steward when you create or edit the category or term.

3.3.3 Customize and Extend

Administrators can define additional custom attributes for categories and terms.

Administrators can create custom attributes to store information about terms and categories, when that information does not fit into the standard attributes and relationships of the glossary model. You can use custom attributes to apply governance standards, enable architecture frameworks, or provide other metadata that is standard for your organization.

When you create a custom attribute, you specify that it applies to either terms or categories, or to both terms and categories. If you apply the custom attribute to both terms and categories, two separate custom attributes are created, one that applies to terms, and one that applies to categories.

Each custom attribute has a name, a description, and a valid value type. The valid value type can be any string or an enumerated list of string values.

You can change the valid value type for a custom attribute at any time. When you change the type, the change does not affect any values that are currently assigned for the attribute. The change determines what will happen the next time a user edits the value for a custom attribute. If you change the type of a custom attribute to String, when users subsequently edit the attribute for any object, they can enter any string value. If you change the type of a custom attribute to Enumerated, when users subsequently edit the attribute for any object, they must select values from the enumerated list of values.

The value of the custom attribute for any particular term or category is initially null. After you create the custom attribute, you can specify its value separately for each term or category that it applies to.

For example, you might create a custom attribute named Data Sensitivity with the following description

> A number from 1 to 5, which indicates the sensitivity of the data. Sensitivity is a subjective measure of the impact of the data being released to unauthorized consumers.

You can specify that Data Sensitivity attribute applies only to terms. You choose the enumerated valid value type and enter the numbers 1 through 5 as valid values. After you create the custom attribute, you choose one of those valid values for each particular term that you want to specify a value for.

3.3.4 Collaborate

There are two aspects of collaboration as mentioned earlier. The first is the notion of collaboration on terms within the tool itself. This is done in the form of collaborating on notes, annotations and sending feedback on the term to the glossary administrator.

- Stewards will control the initial input and oversight of a term. Interested parties can subscribe to a term and be notified anytime changes are suggested or made to a term.
- Administrators and authors can add, edit and delete notes on the browse page of any object.
- From the browse page of an object, you can explore the properties and relationships of the object, and read user notes about the object
- From the browse page of an object, you can send feedback about the object to the glossary administrator.
- The glossary sends feedback using the e-mail application you currently have configured as the target for “mailto:” links.
- From the browse page of an object, you can explore the properties and relationships of the object, and read user notes about the object.
- Administrators and authors can add, edit and delete notes on the browse page of any object.
- The other aspect of collaboration is exchanging or importing information from another tool, such as WIA or IS.
- You can import categories, terms, and custom attributes into the repository of WebSphere Metadata Server.
- You can export categories and terms from WebSphere Metadata Server.
- You can import and export categories and terms by using the Categories and Terms MetaBroker.
- You can import categories, terms and custom attributes by uploading an XML file that contains them.

- In the XML file, you can designate relationships between categories, subcategories, and terms.
- You can specify values for custom attributes for categories and terms.

Prerequisites:

- You must have the Business Glossary Administrator role or Business Glossary Author role to perform this task.
- You must prepare an XML file that complies with the IBM WebSphere Business Glossary category and term data file schema. Links to the schema and to a sample XML file that complies with the schema appear in the Upload Categories and Terms page.
- Names of categories and terms must start and end with a character that is not a space. Names cannot contain any of the following characters:

  . (period)

  , (comma)

  ; (semicolon)

  % (percentage sign)

  ″ (quotation marks)
- You must use the appropriate entity references for reserved characters in XML:

3.4 Walkthrough (Real Life situation)

Here is how WBG is actually used in a real life situation. The goal is to create “Business Terms” that provide the following benefits to a company. Below we will quickly go through an example. We will go through the following steps.

- Create a new term
- Browse existing terms & categories
- Collaborating with other users using notes & annotations
- Searching for terms & categories

We will create a new term called “Account Number” based in the example layout and information below.

Account Number: The ten digit account number, sometimes referred to as the account ID. This value is of the form L-FIIIIVVVV. Owned By: Controller’s Office, Mary Smith X3256 Synonyms: Account ID See Also: Account Type

The initial pages show the “overview” page where a user gets an overview of what is currently in the WBG and can then “manage” existing or add new terms.

The first step is to create a new business term which is shown in Figure 11. The screen below shows where the “terms” name in entered, what category it belongs to, and who the “steward” is. The steward is defined as being the “owner” of the definition of this particular term and accountable for its maintenance. .

Also on the page you can see where the actual agreed upon definition is entered. There are fields for long and short descriptions, usages and as well as an example of the term’s usage. You can also see a “status” drop down; this is used for collaborating on terms

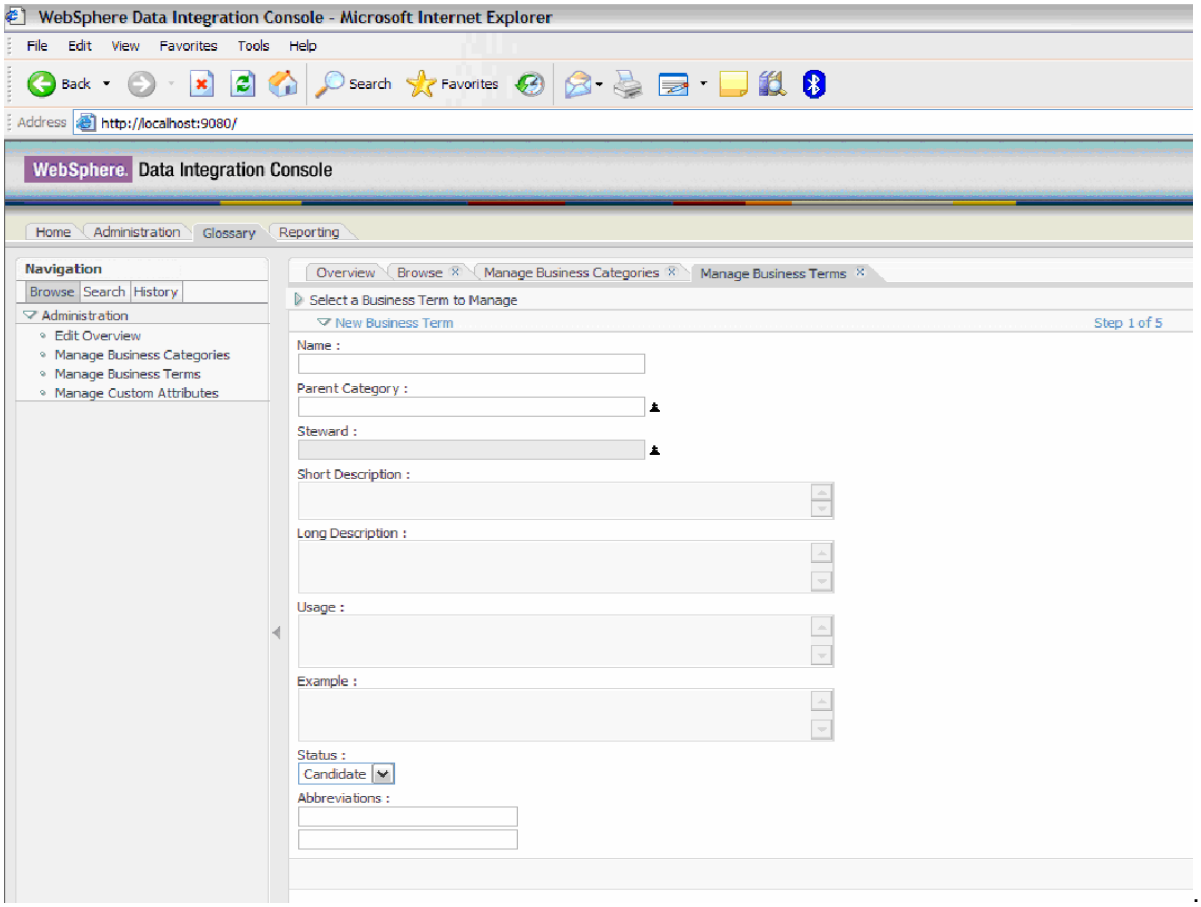


Figure 11: Create a Business Term

You can browse the glossary structure to explore categories, terms, and objects in the unified metadata repository of IBM Information Server.

You can start browsing the glossary from the Overview page, which displays the top-level categories that the glossary administrator has designated as most important for navigation in the metadata repository. You can also search for objects and select an object from the search results.

When you select an object, the browse page of the object is displayed on the Browse Glossary tab, which lists the name, class, steward and other important properties of the object. You can inspect the attributes of the object, browse its relationships to other objects, and send feedback to the administrator. Administrators and authors can add and edit notes about the object.

Next we show a user “Browsing” for business terms by categories once they have been created (Figure 12). You can see a person is browsing the business term “Customer”. In this particular case you will see a description or definition of the term “Customer”, Note: in this screen shot example there is no steward listed.

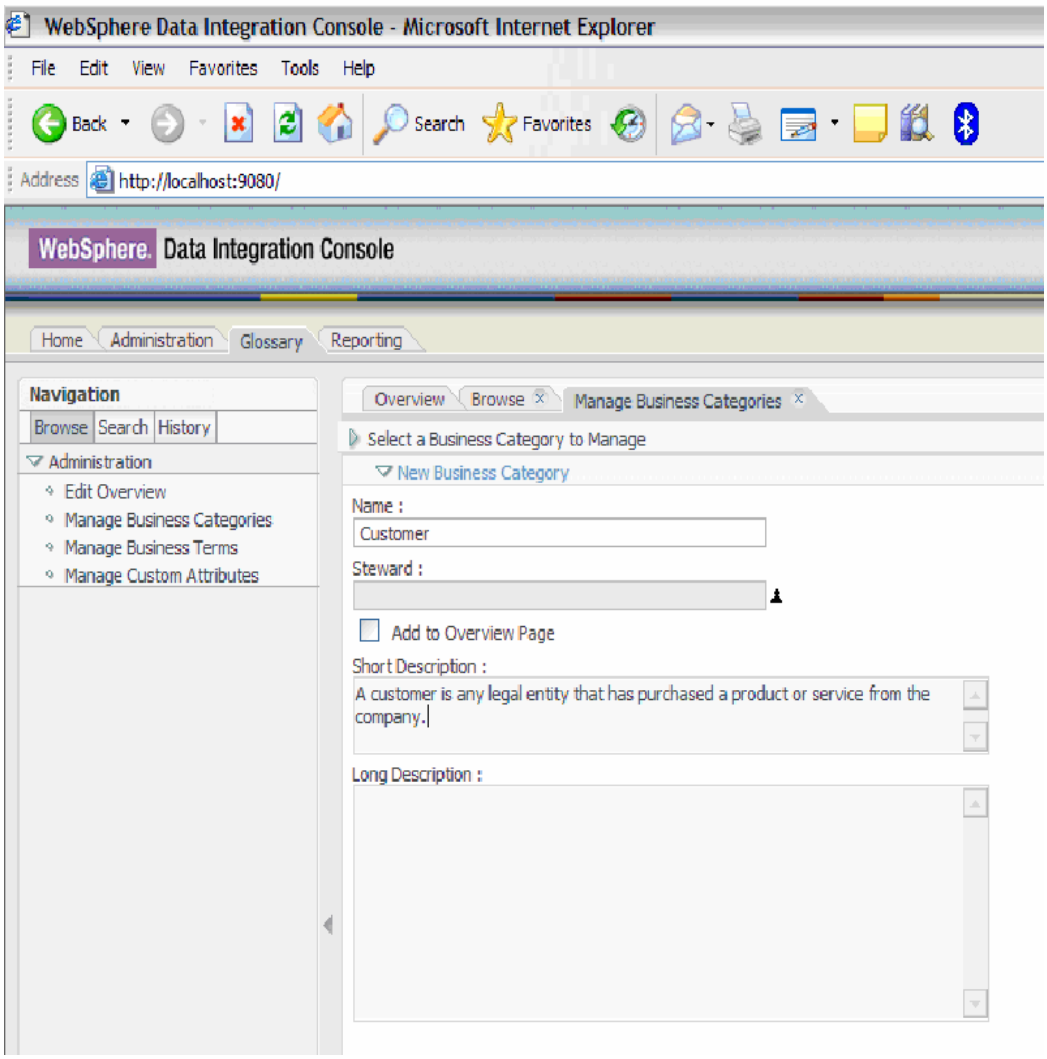


Figure 12: Browse for Business Term by Category

Below you can see an example of how a user would “browse” existing terms that have already been entered into WBG.

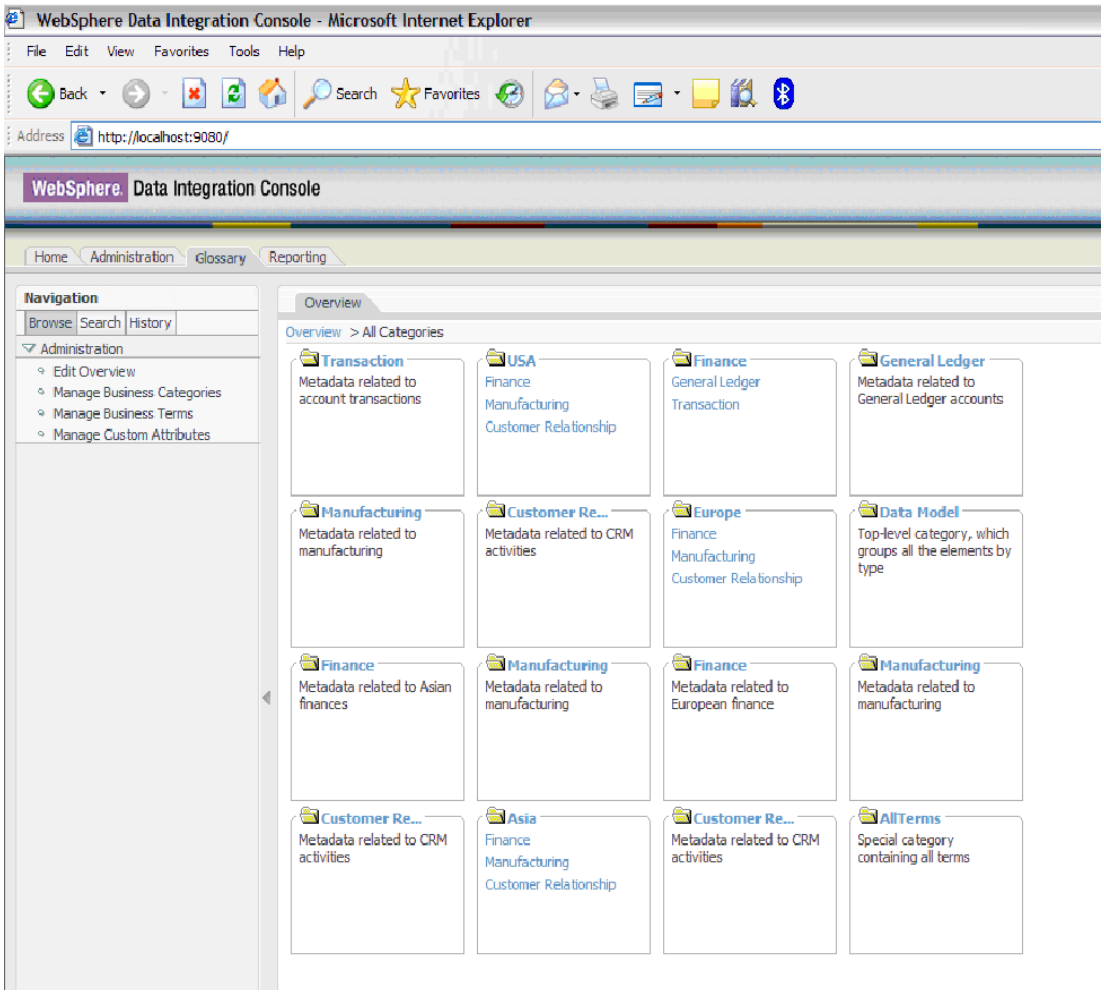


Figure 13: Browsing Through Existing Terms

Once terms have been created and/or imported from an external source they will exist in the glossary. WBG has the ability to accept metadata from a tool such as Rational Data Architect (RDA) if the modeling of a data source has been done in such a tool. If not, the user is going to have to collect the information using the process described in the SOMA 3 information discipline paper (Ref [1]). This would be a manual collection and input of business terms as described in the paper referenced above.

Once that data is collected and entered into WBG all users with the appropriate permission have the ability to collaborate on or subscribe to any term in the Business Glossary that they are allowed access to. In the example of Account Number we might have a Finance person, a Business Analyst, a DBA and the Steward for that entity that are allowed access to the entity “Account Number” for purposes of editing, changing the definition or associated metadata for that entity

What is the purpose of collaboration? The users have the ability to perform the following function on the data/metadata they are allowed access to.

- Add Notes/Annotations to any object in the glossary
- Enables business users to enrich the metadata associated with information assets
- Similar to the user feedback section on many online storefront sites

- Immediately consumable by other users
- Monitored by administrator
- Provide feedback to system administrator

Below is a screen shot of how users might collaborate on a certain business term. Users have the ability to collaborate and annotate various aspects of any defined terms in the glossary. They can ad new notes, comment on existing ones. A user might add a new “Note” to a term for reference by anyone who has access to that term.

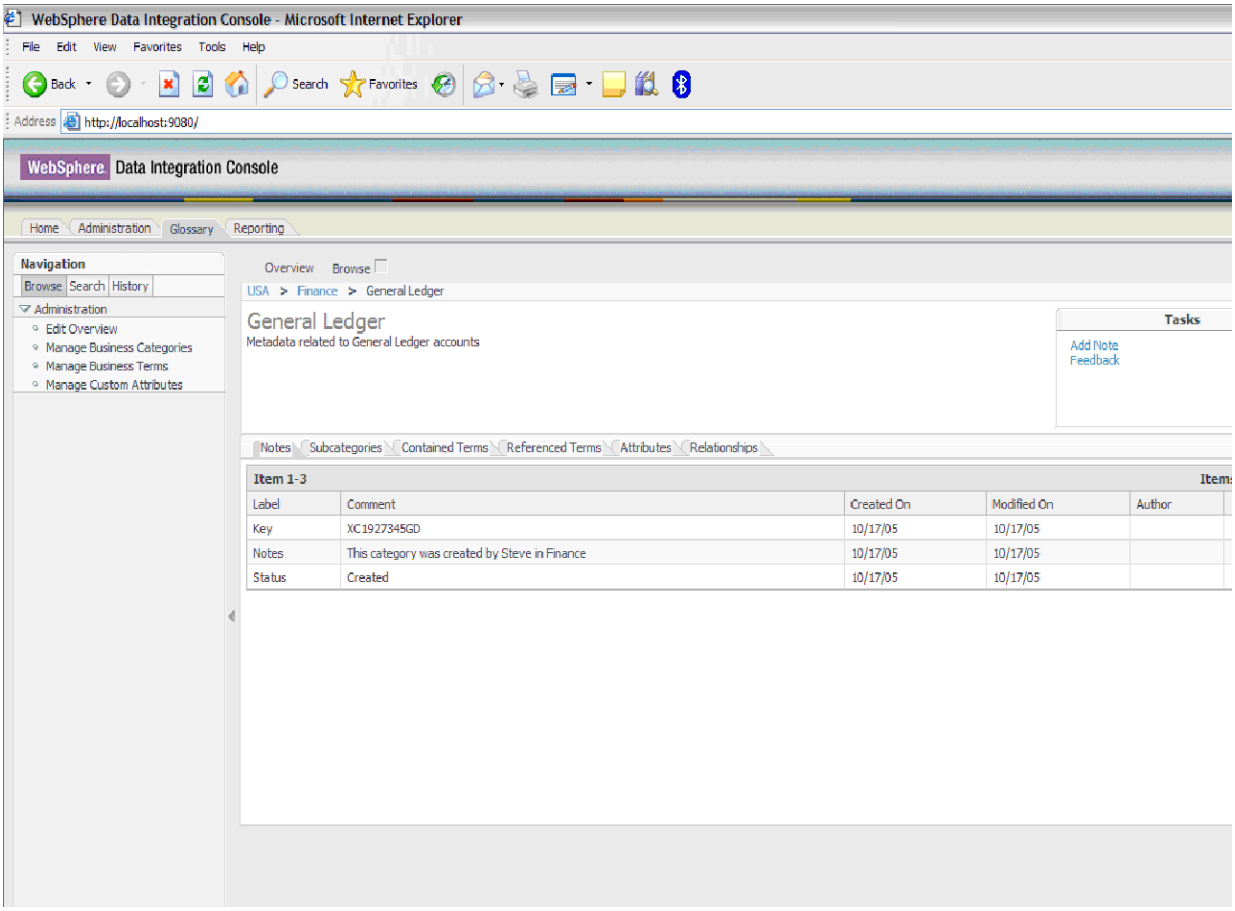


Figure 14: Adding Notes to a Business Term

Users also have the ability to perform simple or advanced searches in the Business Glossary. Here is a screen shot of that action. As you can see there are many possibilities, both simple and advanced, that users can search by.

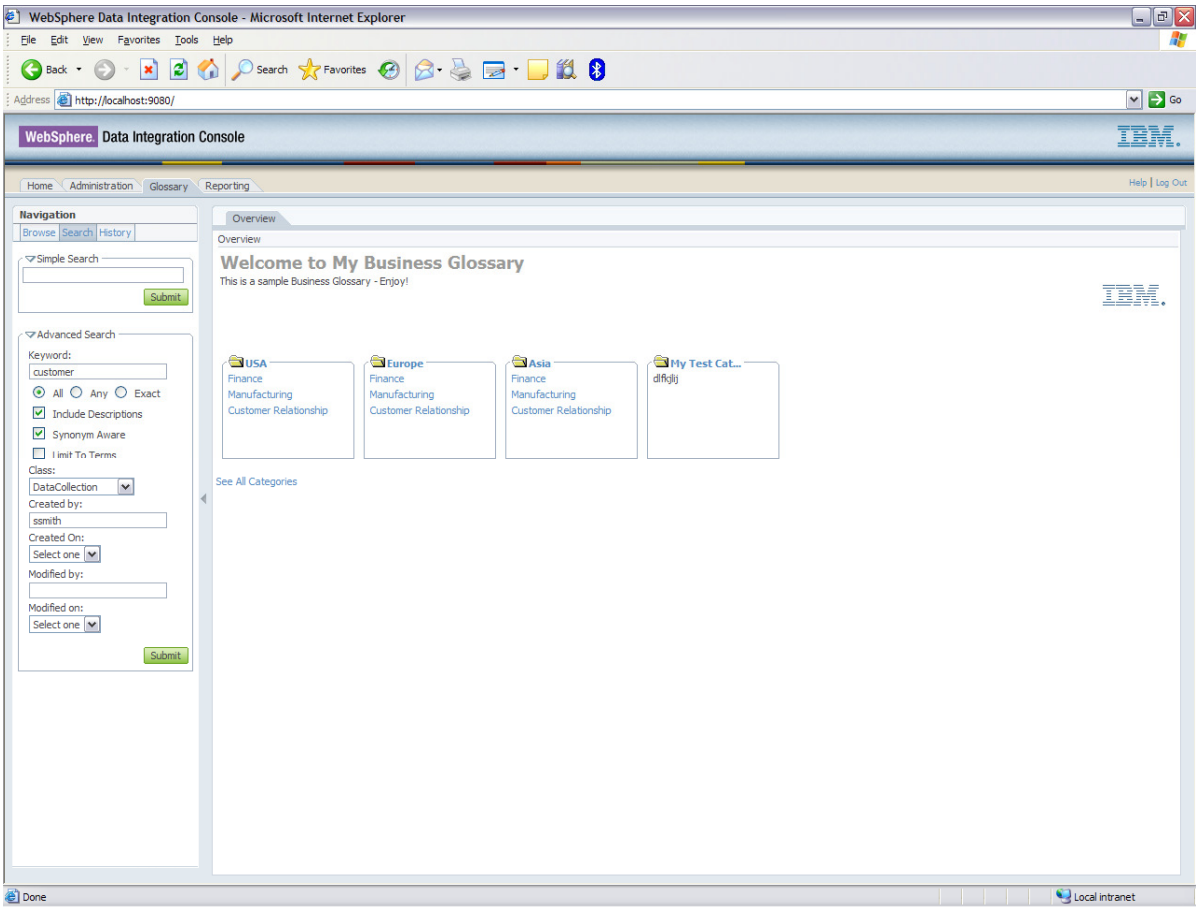


Figure 15: Search

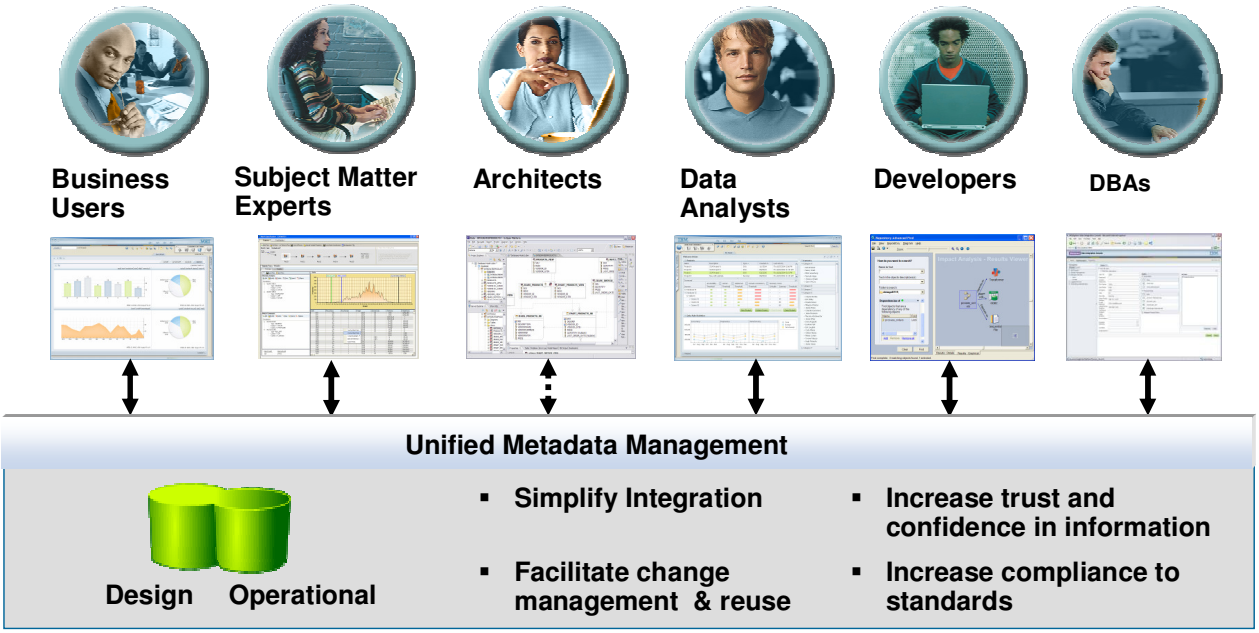
Besides just being able to Search for terms and metadata, the WBG tool allows for some very powerful “Drill Down” to the technical metadata associated with the term.

- Terms are used to describe technical artifacts from other applications
- Enables drill-down to this technical metadata
  - Data Models
  - ETL Flows
  - Business Intelligence
- These technical artifacts are also browsable
  - View all attributes
  - View all relationships
  - Set or view stewardship
  - Run analysis
  - View or add Notes

3.5 Summary

Above are some of the usages and reasons for using a tool like WebSphere Business Glossary. The benefits being that we can use WGB to create a common vocabulary between technical and business users in a common environment that allows for collaborative creation and maintenance of valuable information and metadata about the information.

Here is a common pictorial representation of what this might look like in an initiative. The technical team may be using a tool like Information Server to create data flows and processes, while the business users and stewards are able to see the meanings and metadata that the technical teams are using, in context of the business process.



Another benefit is the collaborative nature of having all these various teams working together in a common environment. An SME might recommend a change to the Data Steward or a business user and using the common environment to suggest the change a governance process can easily be enforced. All interested parties are able to see, comment on and approve all changes, while the technical team can also be kept up to date with any changes being made.

It also means that everyone has immediate access to critical data through this common environment.

4. WebSphere Information Analyzer

WebSphere Information Analyzer (WIA, formerly known as ProfileStage) is a tool that allows for discovery of primarily quality related aspects of existing data stores. It also helps users perform gap analysis between source(s) and target systems.

WIA's primary benefits are to:

- Create a greater understanding of data source structure, content and quality
- Ensure healthy data quality throughout the project life cycle
- Eliminate the risk and uncertainty of proliferating bad data throughout the enterprise

The SOMA 3 information discipline paper (see Ref. [1]) includes an activity during the specification phases to analyze existing information/data assets. The objective is to assess whether or not an existing data source can be leveraged to implement a service and if any additional data transformations or data cleansing operations are required. Consider an example where you have multiple existing systems that you may need to integrate. Irregardless of what integration approach you will ultimately use (ESB, ETL, etc.); you need to know if any data quality issues need to be considered. Do you have matching keys or identifiers that you can leverage to integrate the two systems? WebSphere Information Analyzer also helps determine survivorship and which source might have the best or cleanest data entities when being used to consolidate data stores or map various sources to a single target system.

It is most effective when these stores are some form of relational database, although it also supports XML, flat files and other structured file types.

WIA is built to leverage the unified metadata repository of IBM Information Server. Because the repository is shared across all products, when data profiling occurs using WebSphere Information Analyzer the table definitions and the pertinent profiling information – such as primary key information, foreign keys, notes, etc. – become available to an Information Server user in the DataStage and QualityStage Designer, with no export/import, as shown in the screen below.

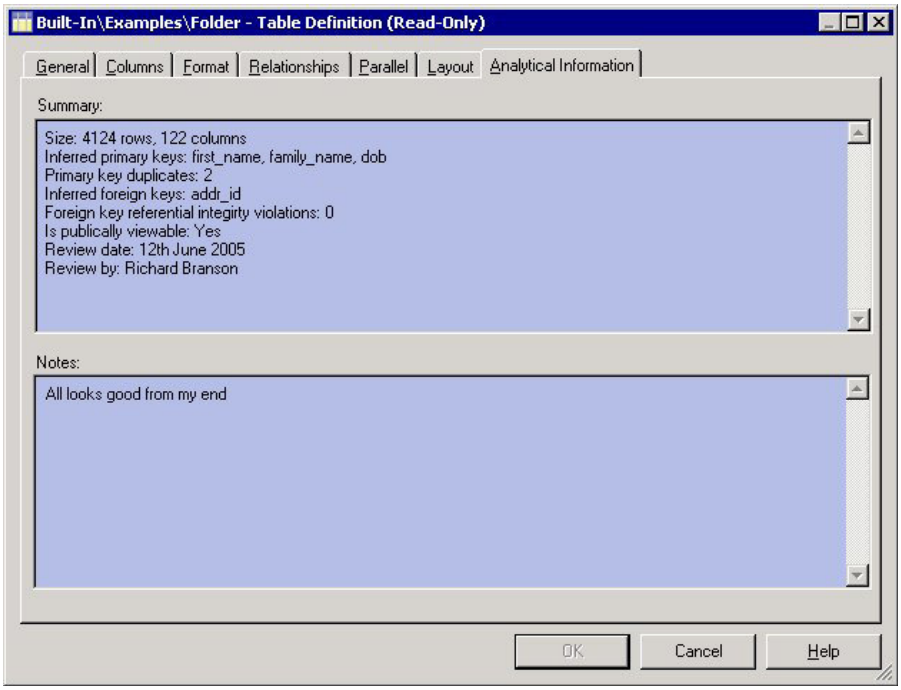


Figure 16: Metadata Access in WebSphere Information Analyzer

The unified metadata management platform of IBM Information Server also provides a number of services, located on the server for performance and scalability, which Information Analyzer utilizes. This provides, for instance, common scheduling services that the Information Analyzer user can use to determine the timing of important profile executions.

4.1 Getting Started

There are two components of WebSphere Information Analyzer:

- the server component that encompasses the engine and the repository (which is the unified metadata repository of IBM Information Server) and
- the client component that is installed on the desktop.

In a typical environment, the server and the client are not on the same physical machines but the configuration may vary depending on usage and number of clients required. While it is possible to have both server and client on the same machine it is recommended that the WIA engine and repository have a dedicated machine and its resources to provide maximum benefit in the enterprise. As long as there is connectivity to the data store(s) being analyzed then it is recommended that WIA's engine have its own resources and not compete with other products for resources.

When you analyze data with WebSphere Information Analyzer, the results of your data analysis are stored in the analysis database. By default, the data analysis database is the metadata repository database and is

accessible to all authorized users. If you accepted the default names, the analysis database is IADB and the metadata repository is xmeta. You can also separate the analysis database from the metadata repository. However, to separate the database from the repository, the database must be a DB2®, SQL Server, or Oracle database. A DSN connection must also exist for the database.

The connection between the ODBC data source and the analysis database must be configured after you install and before you work. An administrator configures the information analysis settings for the analysis database and an analysis engine. The analysis database and analysis engine are components of the IBM Information Server that WebSphere Information Analyzer uses when running analysis jobs.

The analysis database, analysis engine, and source database can be on the same server or on different servers. If you use different servers, two types of configuration are possible:

- Analysis database and source database on one server, and analysis engine on a different server
- Analysis database and analysis engine on one server, and source database on a different server

WebSphere Information Analyzer is supported by a broad range of shared suite components in IBM Information Server. Standard services are provided for data source connectivity, system access and security, logging, and job scheduling.

WebSphere Information Analyzer shares discrete services with other IBM Information Server components. You have the flexibility to configure suite components to match varied environments and tiered architectures.

- **Common core services**
  Provide general security services such as administrative roles and logging.

- **Workbench**
  Provides an interactive and flexible user interface where you access analysis tasks and organize results.

- **Metadata repository**
  Holds the metadata that is shared by all suite components. The metadata repository also stores analysis results and reports.

- **Metadata services**
  Provide query and analysis tasks by accessing the metadata repository

- **Connectors**
  Provide common connectivity to external resources and access to the metadata repository from processing engines.

- **Parallel processing engine**
  Provides a scalable engine with the processing power to analyze large volumes of data.

The IBM Information Server console is a task-oriented user interface that integrates suite components into one unified framework. The console contains workspaces that you use to create and complete information integration tasks such as investigating data, creating job schedules and logs, and deploying applications or Web services. The console consists of five major areas:

- **Workspace navigator menu**
  You use the workspace navigator menu to go to workspaces in the console.

- **Palettes**
  You use palettes to go to open workspaces, and create notes.

- **Main menu**
  The File, Edit, View, and Help menus are in the main menu at the top of the console.

- **Workspace**
  A workspace is an area where you complete tasks.

- **Status bar**
  The status bar shows the progress of activities, error messages, and warnings. The console also contains multiple help features: Instruction panes Instruction panes help you to complete tasks in a task pane.

- **Information center**
  The information center is this web-based help system and knowledge base, in which you can find conceptual and task-based information about the suite, the console, and the tasks that you can complete in the console.

- **Contextual help**
  When you need assistance while you work, you can click the F1 button to open contextual help. For example, if you are in the project properties workspace and need help setting the properties, you can click the F1 button to open the project properties documentation in the information center.

Before you or other users can begin analyzing data, you must create and open a project, establish connectivity, configure system resources, import metadata, configure the project, and set up security.

A project is a logical container providing a secure framework that binds together a set of data sources for analysis with a set of users performing specific roles. To set up a project, you first create a project and provide basic project details. Then, you establish external source connectivity, configure your system resources, import metadata into the metadata repository, associate metadata with the project, and assign users to the project. Finally, you set the analysis options for all analysis that occurs in the project.

A project is an object that contains all the information that you need to build and run an analysis job and view analysis results. Each project is unique, depending on the analytical settings that were configured in the project when it was created. However, multiple projects can access the same metadata that is stored in the repository and share the same users and data sources across suite components.

Typically, a project administrator or a user with administrative rights creates a project and configures all of the data source connections and security options that are associated with the project. After a connection to a data store is created, the administrator imports metadata and registers it to the project. Only the administrator has access to connection information and security options. An administrator uses the security options to assign user roles to authorized users. A user can then access certain project tasks depending on the role that they were assigned.

The following list summarizes the project creation process:

1. Create project
2. Add data sources
3. Add users
4. Configure options

4.2 Motivation & Overall Product Scope

Information Analyzer introduces a new approach to the user interface

- Navigation based on standard life-cycle methodologies, shortcuts and related links to quickly move to desired functions
- Graphically enabled to allow users to easily understand data and find anomalies

The screen shot below shows just a part of the user interface that a user will see during a column analysis when investigating a source repository.

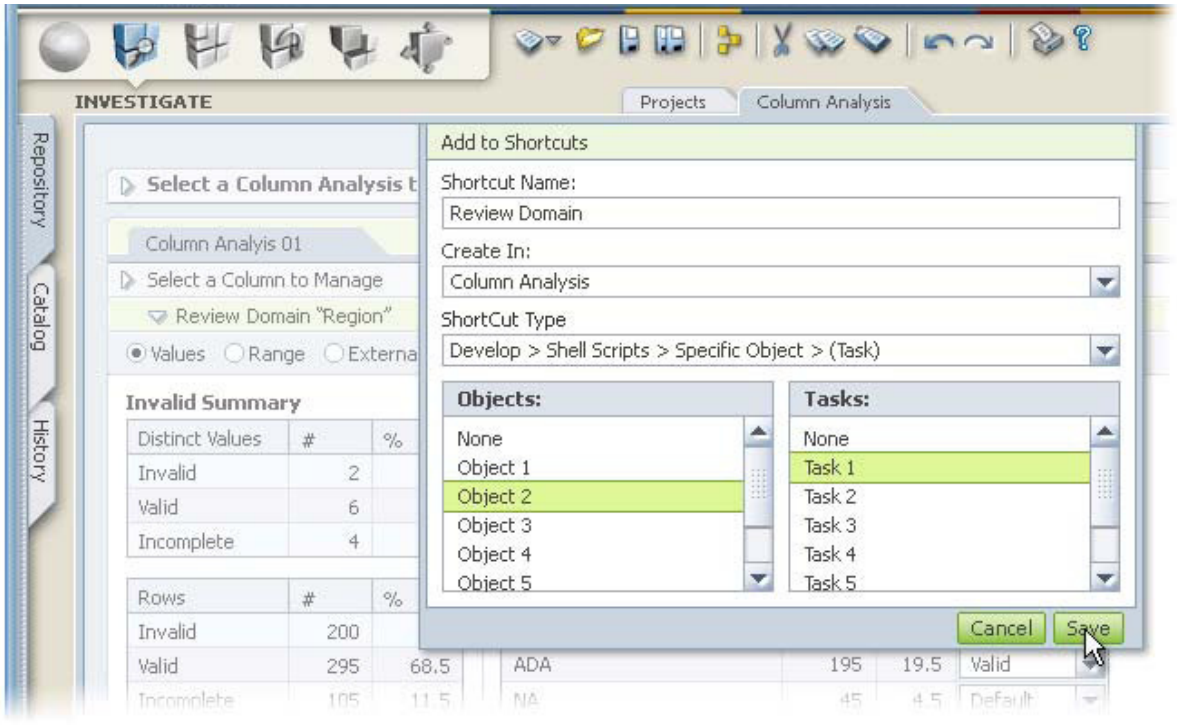


Figure 17: WebSphere Information Analyzer Screenshot

Work in the user interface utilizes a task-based approach to help the user quickly perform what they need to do, when they need to do it. The core menu is built around such common functions of Investigate, Develop, and Operate.

- **Project-based Structure**

  Information Analyzer brings a project-based approach to analysis, allowing users to segregate their work according to their needs, whether based on business unit, data integration project, or any other desired configuration. The project structure allows configuration of users based on role and the incorporation of specific data sources for analysis. Security is also configurable within the project, allowing access level controls as noted above.

- **More Graphical Enablement**

  Information Analyzer incorporates more visual displays and graphs. This allows the user to view information in multiple ways, quickly identifying issues or anomalies, and annotating the findings accordingly.

- **All Standard Data Profiling Functions**

  Information Analyzer brings forward from ProfileStage, the standard data profiling functions of column analysis, primary key analysis, foreign key analysis, and commonality (also called redundancy) analysis. However, there are some significant changes in approach for each of these functions.

- **Column Analysis**

  Column analysis now is performed natively in our parallel engine for maximum performance and always stores full frequency distributions by default. Inferences are made not only for typical structural components (e.g. data type, length), but also for data classification, a process previously only in AuditStage. Analytical results have been streamlined for optimal review by users. Column analysis can also be scheduled through a standard function so that analysis can occur off-cycle or even at regular intervals. Frequency distributions can be viewing graphically or textually.

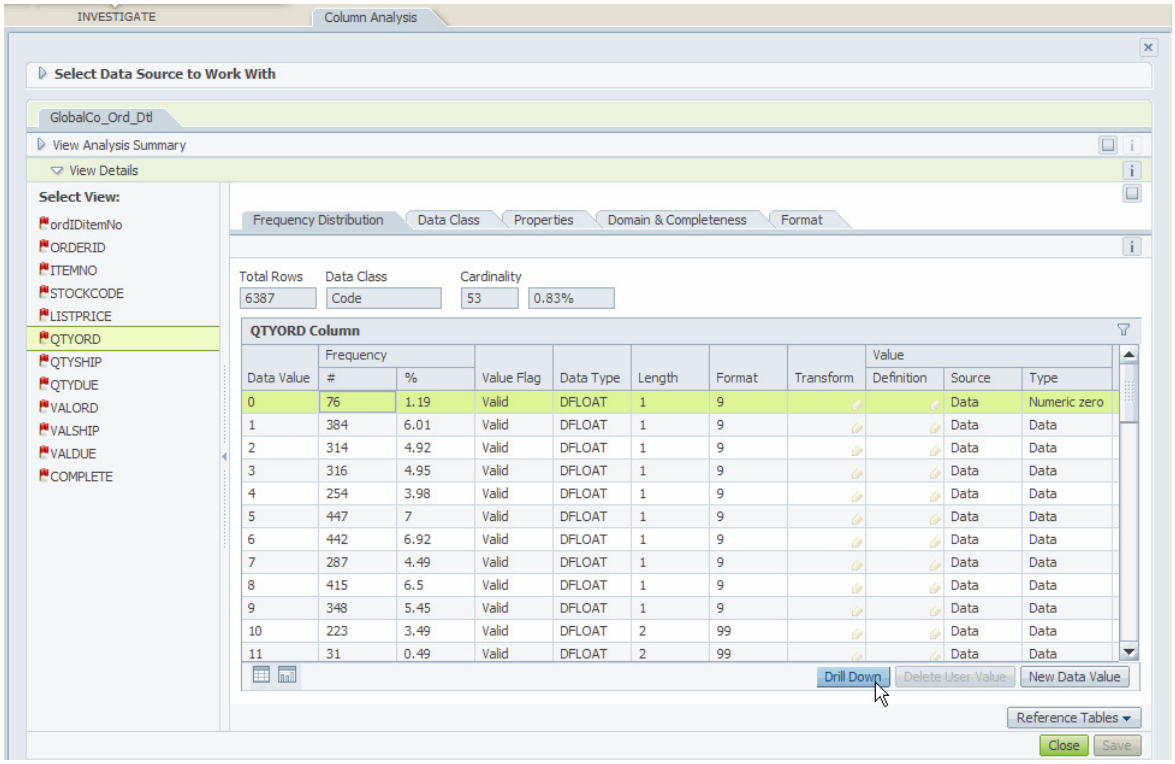


Figure 18: Frequency Distribution Screen in Column Analysis

Similarly, graphical enablement also helps analysts quickly identify issues in data properties such as length (shown below) or data types.

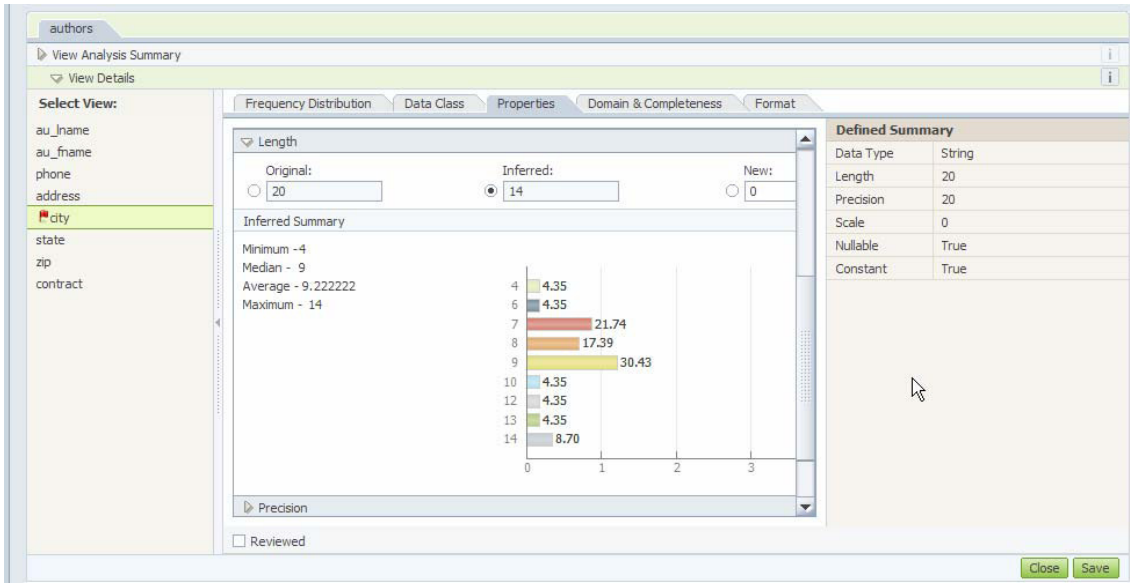


Figure 19: Data Property Issues Screen

- **Domain Analysis**

  Information Analyzer incorporates profiling functionality previously separated in AuditStage. Now users can analyze their column data for completeness and validity, immediately evaluating the data in these two key dimensions. Capabilities include assessing completeness and validity based on actual value, data ranges, or external reference sources. Users can also save their choices into reference files for subsequent use.

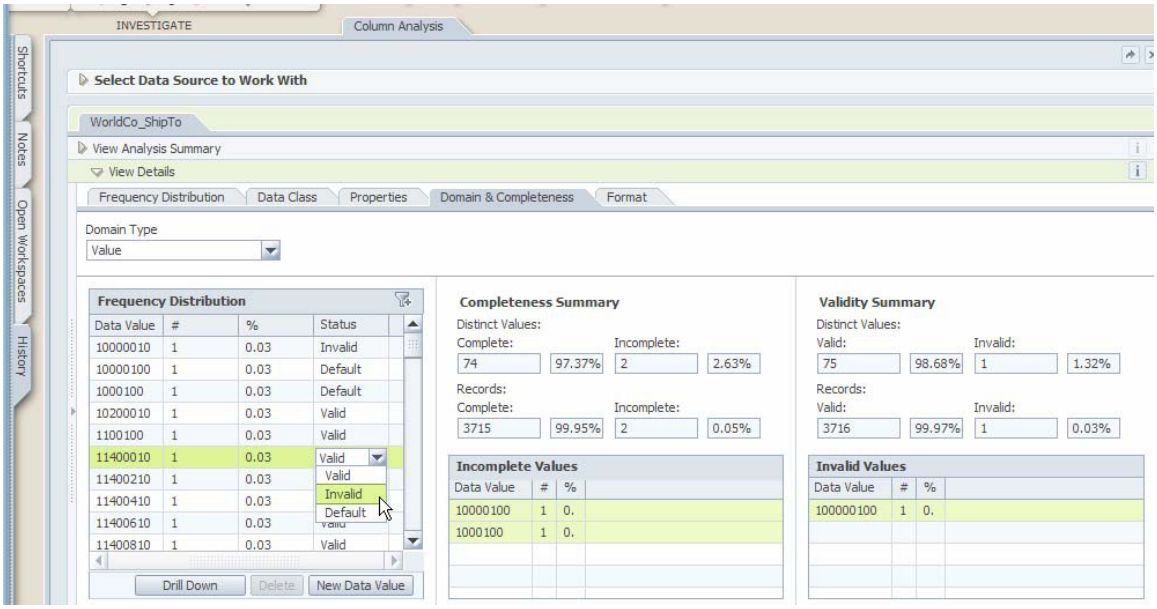


Figure 20: Domain & Completeness Screen

- **Format Analysis**

  Information Analyzer automatically calculates the format for each and every data value, a function previously available only in QualityStage. This allows the analyst to gain insight not only at the value

level, but at the internal structural level of a column. Users can identify the distinct values associated with any specific format, as well as drill down to the underlying records. Violations in formats can be applied directly back to the domain analysis, allowing the user to quickly identify invalid values based on non-conforming formats.

- **Virtual Columns**

  Users can construct virtual columns from any combination of columns within a table and then analyze those virtual columns in turn. Users can assess the pairing of for example State with Zip Code to look for unusual formatting or even compare several virtual columns for common domain values.

- **Sampling & Scheduling Options**

  For situations where an analyst wants to focus initially on a data subset rather than on full volume, Information Analyzer supports the use of data sampling right at the time of execution. Users may choose to sample randomly or sequentially. Further, analysts can take advantage of built-in scheduling services (noted above) to determine when to initiate any core profiling process (and whether it should be repeated on an interval basis).

- **Primary Key Analysis**

  Information Analyzer automatically evaluates all individual columns for uniqueness, null values, and duplicates as a primary key, removing any requirement for additional processing for single columns after the column analysis. Users can extend the analysis to multiple columns, taking advantage of the built-in virtual column support noted above. Through this capability, analysts can evaluate a broad range of column combinations against data samples or specific targeted combinations against the full data volume. Results are readily displayed and allow the user to quickly understand duplicated values or drill back into the full frequency distribution for single or multiple columns as shown below.

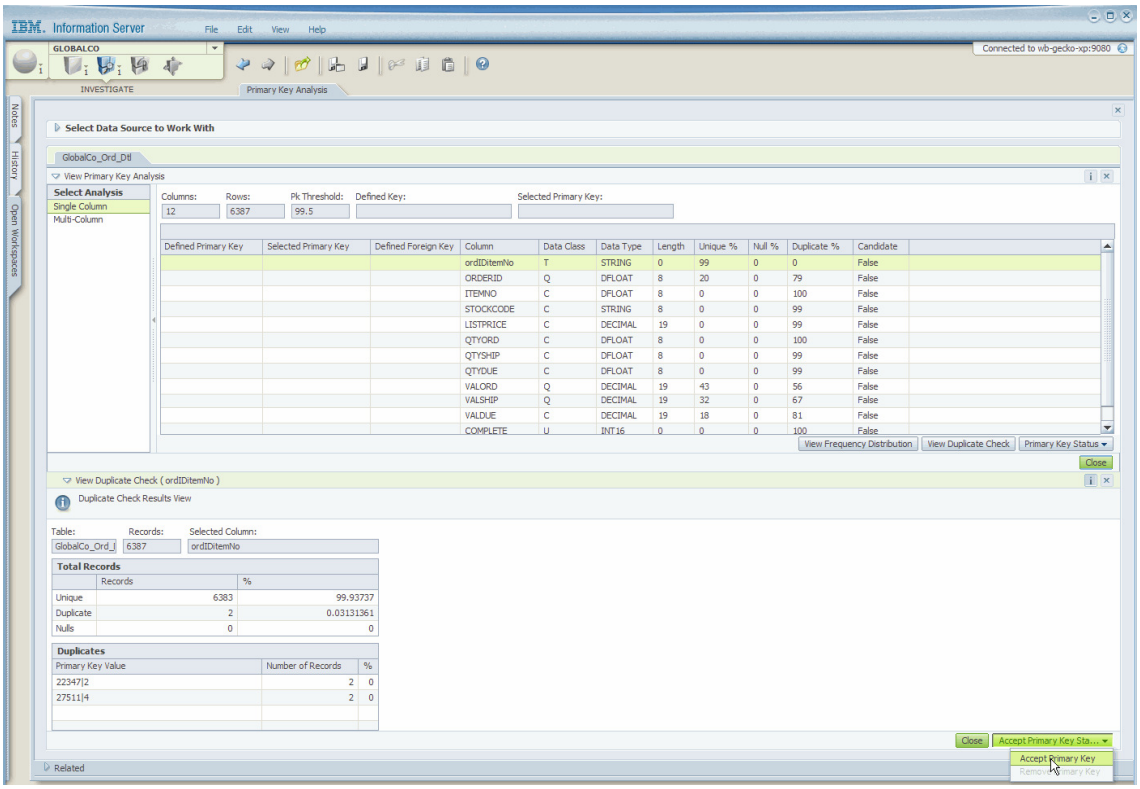


Figure 21: Primary Key Analysis with the Duplicate Check Results View

- **Foreign Key Analysis**

  In foreign key analysis, Information Analyzer supports testing and reviewing of primary-to-foreign key relationships not only between tables of a single data source, but across multiple data sources to help address the challenges of data integration efforts. Users can draw from either defined metadata or inferences out of earlier analysis steps. Further, such analysis is cumulative, allowing the user to focus on items of interest at a specific point in time and then extending that analysis as needed to assess additional key relationships. Details from the foreign key analysis allow the user to understand the overlap of actual values and quickly isolate discrepancies.

- **Referential Integrity**

  Information Analyzer includes an explicit test of referential integrity as part of the foreign key analysis. Violations between primary keys and foreign keys, such as orphaned values or parent records without child records can be viewed as highlighted below.

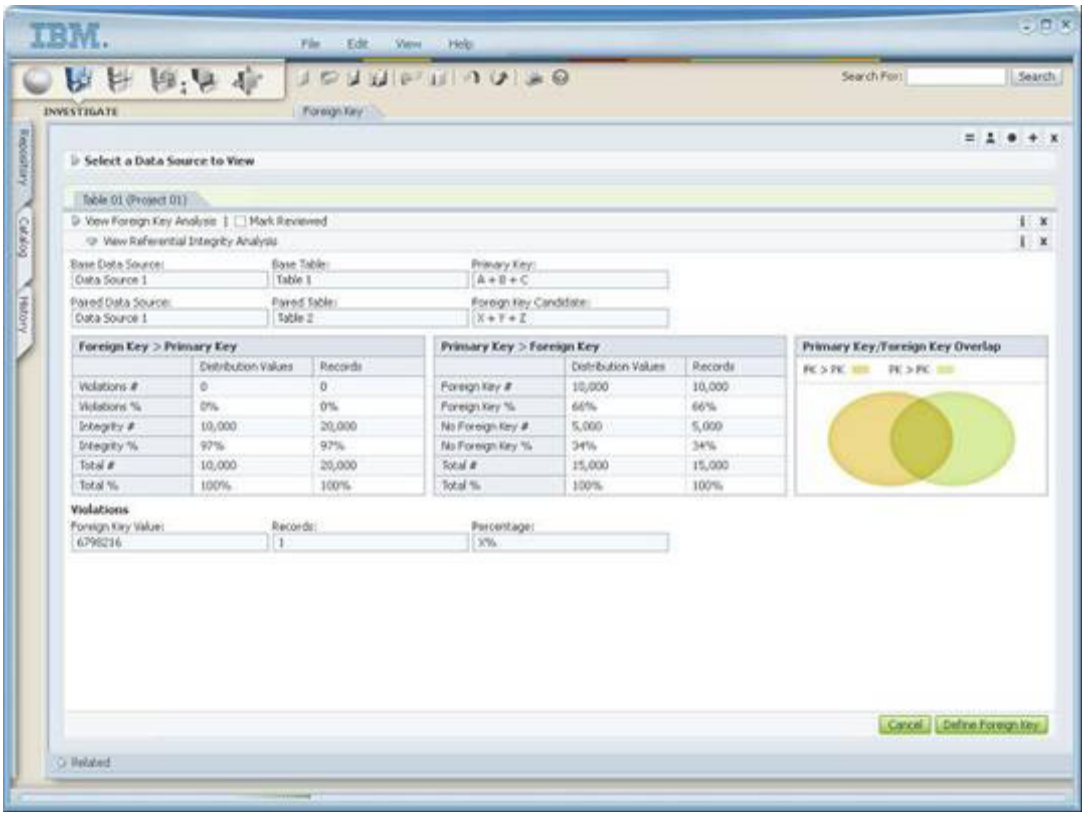


Figure 22: Referential Integrity Analysis

- **Cross-Domain Analysis**

  Utilizing the same techniques applied to key analysis, Information Analyzer extends the capability (previously available in ProfileStage as the relationship analysis) for analysts to quickly review the relationships of column-to-column within a single table, across multiple tables, or across multiple data sources. By leveraging the availability of the frequency distributions from the column analysis, users may focus on columns of interest (for example State Codes across five different systems of record) to understand data overlap or redundancy. Users may explicitly select columns for analysis or allow the system to cross-evaluate columns, automatically weeding out incompatible data pairings. Such cross-domain analysis is cumulative and can be performed as needed over time.

- **Baseline Analysis**

  Information Analyzer provides additional capability to compare the results of one profile execution with another, allowing insight into what may have changed structurally or in content between two points in time. Users establish a baseline with their first column analysis. Subsequently, they may compare another analytical execution of the same tables and columns against the baseline.

4.3 Data Assessment Using WebSphere Information Analyzer

What exactly are we trying to accomplish and how do we most effectively accomplish this analysis? When performing a data assessment on any existing data source we are trying to determine the data's structural makeup. What type of data is it? Is it integer, character etc.

We are also trying to determine anomalies or irregularities in the data beyond the structural aspects. If 99% of the data is integer and in the range of 25 to 30 then anything that lies outside this range may be suspect. So we are looking to statistical anomalies as well as physical structure information.

We are also looking for embedded business rules or logic that might tell applications accessing this data to allow for some form of exception processing for this term. For example, we look at a certain field and find that there is a number like -9999 in the field, it is entirely possible that this is used as an indicator to a program to look elsewhere for the data that should have resided in that field. Most initial assessments find that close to 15% of legacy data has some form of embedded logic or exception handling embedded in the data.

In short, we are determining the state of the data "as it is" today. We are also looking at the effort and complexity of recasting or standardizing/cleansing/enriching the data for the purpose of our initiative. We are also looking for any gaps that exist in the data if we are repurposing it for either a different data model or some enterprise application like WebSphere Customer Center.

Whether that initiative is consolidating several data sources or correcting them in place doesn't really matter. The process and use of WIA is the same. This resultant analysis can be used to reduce project risk and cost as well as to provide more accurate scoping of the project.

Typically there are 3 phases we consider when doing a full data assessment. The table below represents the steps of analysis performed in each phase.

Source System Analysis	Target Analysis	Alignment and Harmonization Analysis
▸ Frequency Distribution ▸ Default value analysis ▸ Not used columns ▸ Constant candidates ▸ Structure analysis ▸ Column content analysis ▸ Embedded business rules ▸ Business/Technical documentation	▸ Attribute Gap Analysis ▸ Data Gap Analysis ▸ Field Length Analysis ▸ Data Migration Scoping ▸ Master Data Structure Analysis	▸ Standardization Analysis ▸ Matching Remediation Analysis ▸ Attribute Alignment Analysis

Table 2: Steps of Data Analysis

We will describe the tasks in each of the phases in more detail in the following sections.

4.4 Source System Analysis

Why do we want to do a source system analysis? Whether or not we are moving or integrating single or multiple sources we will still benefit from understanding each source individually. For each source we will perform the following types of analysis and provide documentation. All of these analysis help provide insight into the value and integrity of our existing data source(s).

- Frequency Distribution
- Default value analysis
- Not used columns
- Constant candidates
- Structure analysis
- Column content analysis
- Embedded business rules
- Business/Technical documentation

Frequency distribution

This is either a graphical or numerical picture of the distribution of values within an entity. In other words if I have 90% of my values between a certain range the frequency distribution would tell me where I have

outlying values and possibly a quality issues in those outliers. The screen shot below shows the graphical representation.

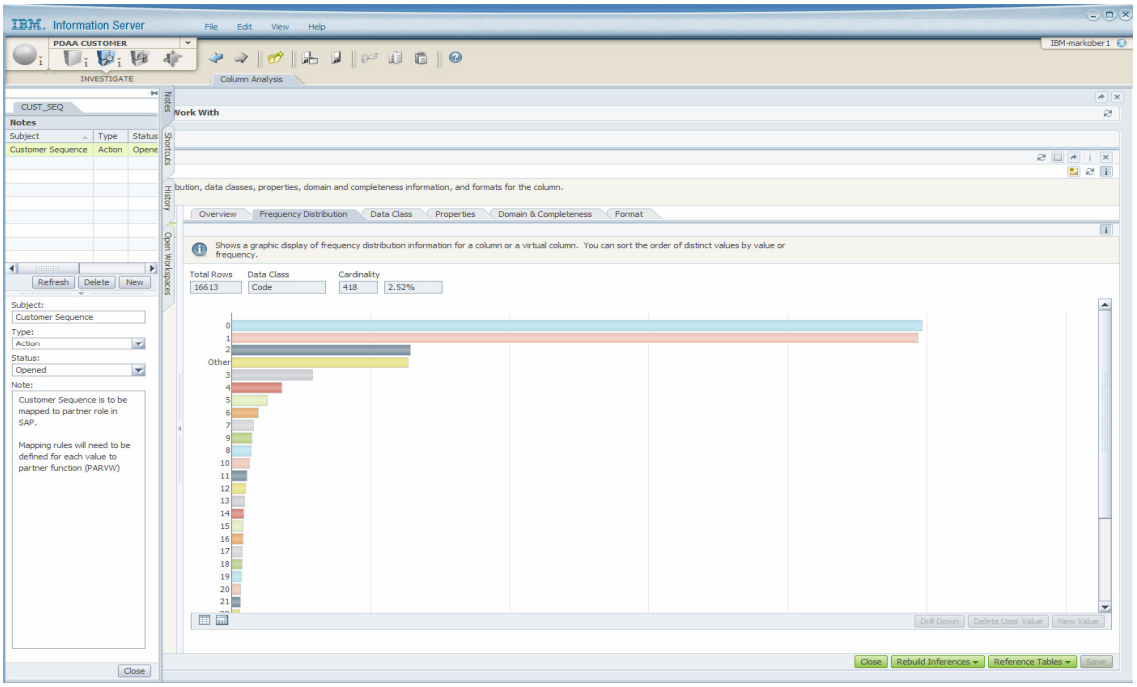


Figure 23: Frequency Distribution Screen

Default value analysis

This is simply an analysis to determine when and how frequently we have encountered a default value being used instead of an entered value.

Not used columns

Simply a valuation of how frequently data is not entered into a column. Indicative of its lack of importance or lack of integrity checks to enforce defaulting.

Constant candidates

This is an analysis that could help to determine where columns of data are so constant that they can be candidates for constant values or business logic used to place default values could be utilized.

Structure analysis

This is a fairly complex output (refer to figure below) where you can see the data type, physical structure/representation, default values, ranges of values etc.

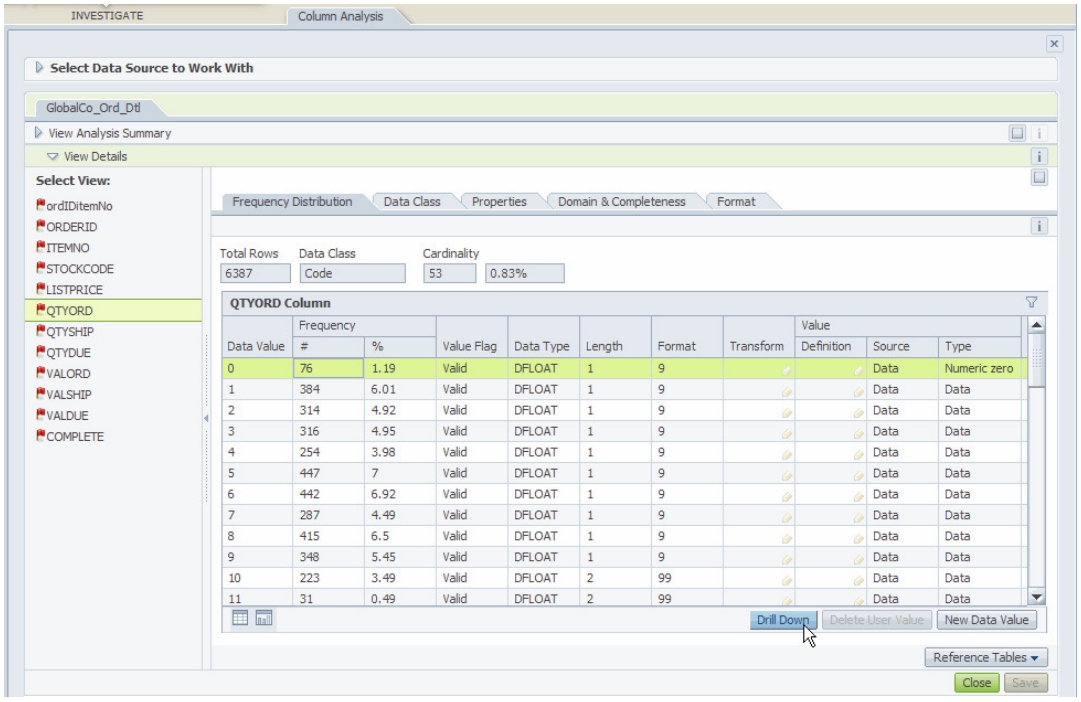


Figure 24: Structure and Content Analysis

Column content analysis

The column content analysis is also a fairly detailed and complex output showing the content of any particular column in a source store. It can tell us whether or not values are missing or outlying. The same screen shot (above) is used for both content and structure analysis.

Embedded business rules

This is an analysis that will determine where existing data sources have broken down and contain embedded logic in the data itself. Take the case where a column constantly contains a value of -9999 and this indicates to programs that consume this data that alternate processing should be performed. This is an example of embedded business logic contained in the column.

Business/Technical documentation

This is a process step that helps document current data source(s) with all of the above analysis. It may also be the transfer of metadata from WIA into the unified metadata repository to be shared with WBG or DataStage/QualityStage jobs. This documentation can be used in requirements specifications, functional specifications and other information services documentation as well as to document open issues.

The results of running this source system analysis are outlined below:

- Complete set of reports of all aspects of each source system that show initial business and technical communities an initial overview of source data

- As results are analyzed, documentation can be captured that can be used in many contexts
  - Mapping rule specifications that will feed the functional and technical specifications
  - Documentation of follow-up items and issues identified that need to be addressed by the business
  - Information that can be used in future projects and other areas of the organization
- Project scoping and initial risk assessment of data conversion effort
- Data extraction accelerators can be re-used for baseline assessment and for development

4.5 Target System Analysis

After analyzing the source system(s) the next recommended step in the data assessment process is to do an analysis on the target system or target model. This is the model that we will be mapping source data to. Not knowing whether or not this is an MDM initiative or a data integration initiative will determine some of the next steps performed. Because eventually services will have to expose underlying data from the eventual target model we need to determine whether or not we have the underlying data in our current source system candidates.

The target data model may be a logical data model as described in the SOMA 3.1 information discipline white paper that reflects a “to be” state, or the model may represent an existing ERP system like SAP, or a master data management system like WebSphere Customer Center. In either case we still need to be able to map source terms to our target model.

For the target system we will perform the following exercises.

- Attribute gap analysis
- Data gap analysis
- Field length analysis
- Data migration scoping

If the initiative we are undertaking is going to utilize master data then we will also perform the following analysis.

- Master data structure analysis

Attribute gap analysis

This is an analysis of whether or not we can adequately define target entities/attributes and whether or not those can be mapped from existing source systems.

Data gap analysis

This analysis will determine whether or not target model can be fulfilled with existing source data and whether or not extensive transformations may have to be performed. One example of a data gap report would be from the figure below. This is a sample gap analysis between existing sources and an SAP system. This graph below shows, quantifiably, where existing system will map to the SAP model. The graph depicts the value company code fails to map on a significant scale to the SAP company code field. Whereas the next field contact category will map much more regularly to the SAP contact category.

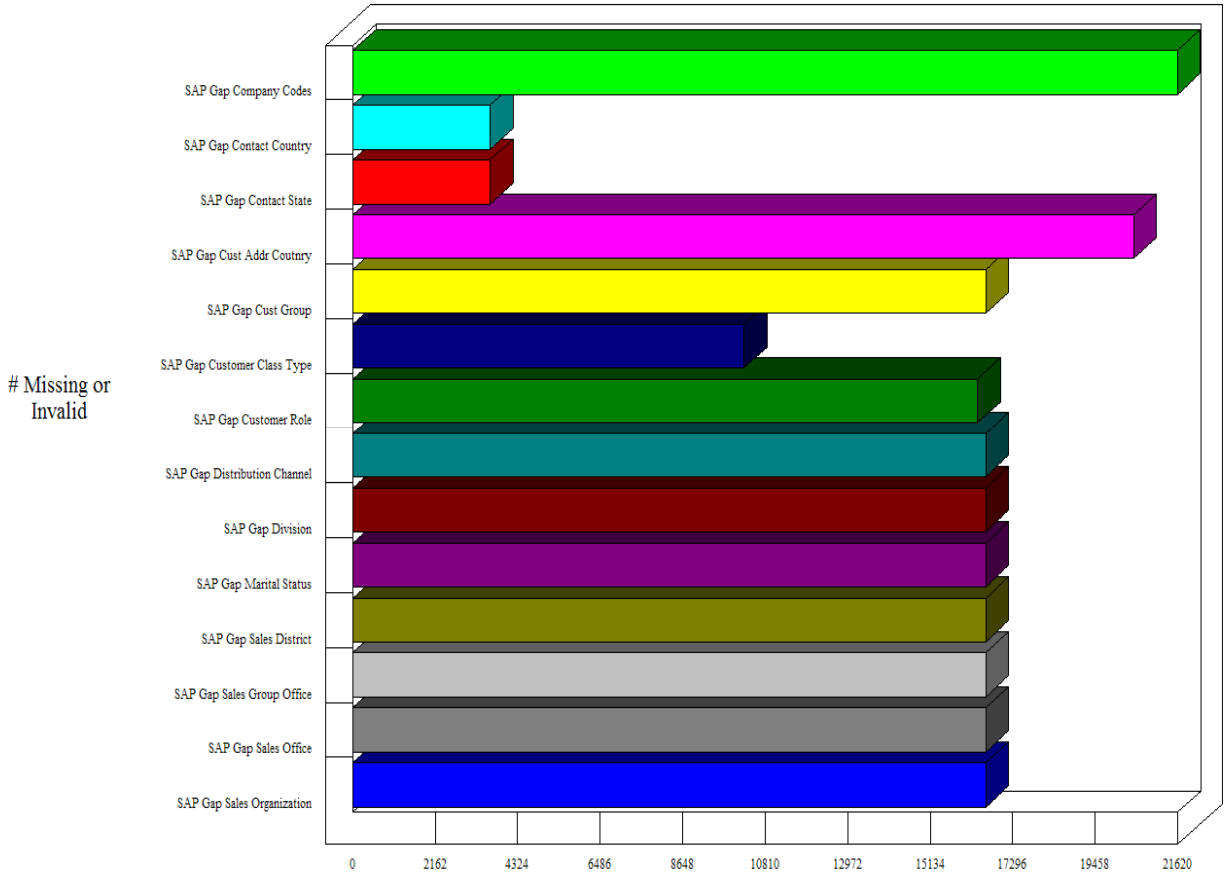


Figure 25: Gap Analysis Report

Field length analysis

The field length analysis is an analysis of whether or not current source systems can map correctly at the field length level of our target model. If not then transformations will have to be performed in the mapping phase. This may also provide feedback to the data architects to adjust the logical data model if necessary.

Data migration scoping

This is a process step that helps scope the magnitude of the data migration effort. Based on whether or not we are building a data warehouse, migrating from one or more sources to an enterprise target or doing an

MDM initiative there will still be a migration effort of some sort. This scoping exercise is intended to help put some perspective on how much effort may be required, whether or not we have all the available source data that is needed and what kind of transformations may be necessary.

Master data structure analysis

As noted earlier if the initiative we are undertaking is a candidate for a master data repository then we would recommend undertaking the following analysis. This will help determine what data is candidate for master data and what structure that data may have to take.

This would be the analysis that defines an initial structure and possible candidates for the master data repository. Its goals is to try and determine what terms and relationships should be considered for candidates for master data and what kind of rules will drive those decisions..

4.6 Alignment and Harmonization

What does alignment and harmonization really mean? We define alignment as the method by which we align source data to target data, whether that is through direct 1:1 mapping or achievable through transformations. What kind of transformations are necessary, what kinds of mappings will be required?

We define harmonization as the removal of duplicate data or overlapping definitions. If we are dealing with multiple sources then we “harmonize” until we have a single definition or survivable data entity to map to.

As part of our alignment and harmonization analysis we will have to determine how to standardize records to a certain format, and how to match data. We want to ensure that we are always dealing with the right record, given multiple copies of possible the same information. If we are dealing with multiple records then we need to know which source to go to for the “right” data. This is what we term as survivability (which record survives). It is also possible that we may have to build a composite record based on different attributes from different sources. This is where data integration becomes quite difficult.

During this process we will typically perform the following analysis.

- Standardization Analysis
- Matching Remediation Analysis
- Attribute Alignment Analysis

Standardization analysis

This is the analysis of to understand the operations necessary on source data to standardize it to the correct format. This is typically most prevalent with terms like addresses and names but can also apply to other business terms.

Matching analysis

This is the analysis of what transformations have to be specified to match and survive data in order to create the correct record – the single version of the truth – when integrating and transforming the data.

Attribute alignment analysis

This is to determine what kinds of transformations will have to be performed to map a source term or entity to a target term or entity.

Below are a few example reports of what the user will see in the alignment & harmonization analysis phase of a data assessment.

The first screenshot is an example standardization report. The report shows the percentage of data that was not completely standardized. During a preliminary data assessment we will highlight the data that has invalid contents. This particular graph shows that close to 15% of the data is not standardized.

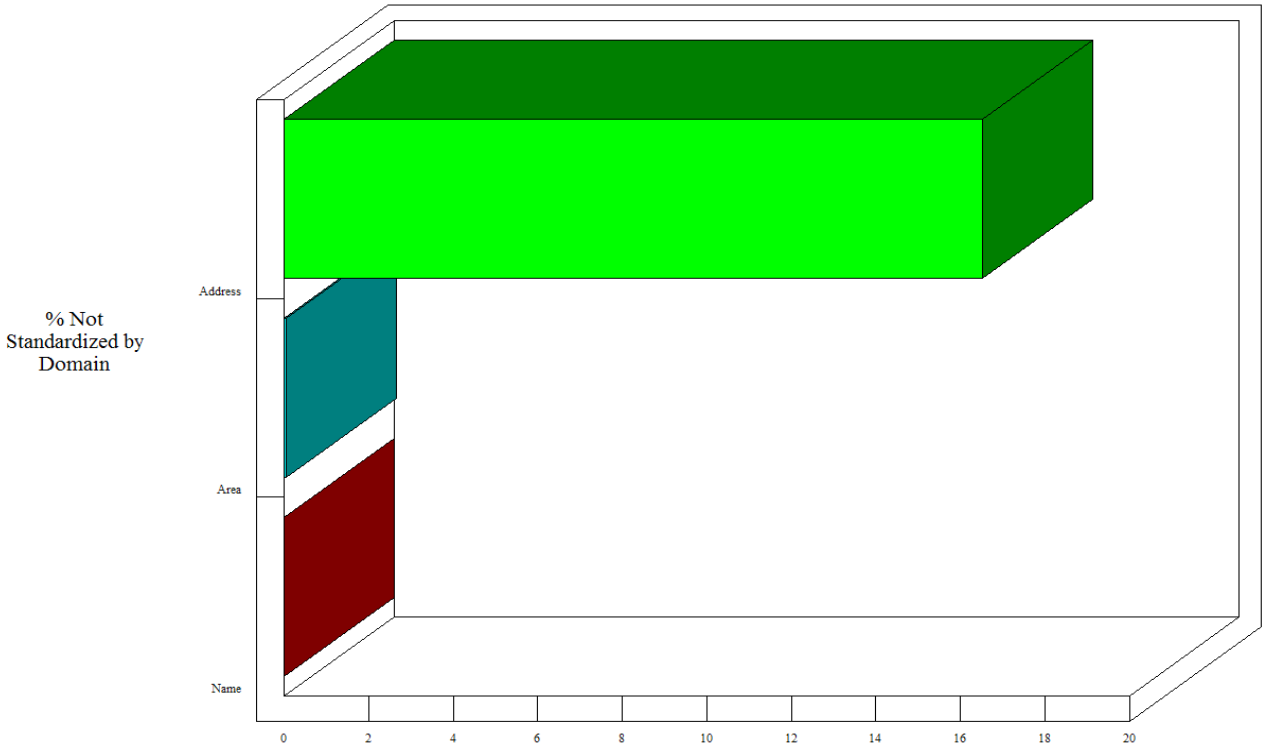


Figure 26: Standardization Report

The next report contains valuable information about embedded data that may not be understood. It is based on postal address matching and standardization as well as uncovering embedded logic or information in legacy data. Such as strings saying “Do Not Use” or “Credit Card Orders Only” things like that.

- Hidden business rules
- Amount of effort required to modify standardization rules
- Embedded information that belongs in other fields
- Misuse of input fields

Post Address
Standardization Result
(Data Not Standardized)

DOMAIN	STANDARDIZED	FREQUENCY	PERCENTAGE	EXAMPLETEXT
Address		60	0.572	ACCOUNTS PAYABLE ACCOUNTS PAYABLE
Address		23	0.219	RECEIVING DEPT RECEIVING DEPT.
Address		13	0.124	CB CB# 7400
Address		11	0.105	BUSINESS BUSINESS OFFICE
Address		11	0.105	VOID VOID
Address		9	0.086	CAMPUS CAMPUS BOX 7247
Address		9	0.086	DEPT DEPT:
Address		9	0.086	TULLASTRASSE 4 TULLASTRASSE 4
Address		9	0.086	UMHC BOX 494 UMHC
Address		7	0.067	CREDIT CARD ORDERS ONLY ***CREDIT CARD ORDERS ONLY***
Address		7	0.067	JHMHC BOX 100245, JHMHC
Address		7	0.067	SCHOOL OF MEDICINE SCHOOL OF MEDICINE
Address		6	0.057	ATTN ATTN:
Address		6	0.057	BIOLOGY DEPT BIOLOGY DEPT.
Address		6	0.057	RM BLDG. 19, RM._____
Address		6	0.057	WAREHOUSE WAREHOUSE BLDG 50
Address		5	0.048	2-1-9 SHINKAWA 2-1-9 SHINKAWA
Address		5	0.048	ACCOUNTS PAYABLE DEPT ACCOUNTS PAYABLE DEPT
Address		5	0.048	CENTRAL RECEIVING CENTRAL RECEIVING
Address		5	0.048	PURCHASE ORDER PURCHASE ORDER #_____
Address		4	0.038	34TH & CIVIC CENTER BLVD 34TH & CIVIC CENTER BLVD
Address		4	0.038	BLDG ONE KENDALL SQUARE, BLDG.#200
Address		4	0.038	DEPT OF PATHOLOGY DEPT. OF PATHOLOGY
Address		4	0.038	MAIL MAIL STATION 201-6
Address		4	0.038	NAVAL STATION BLDG. 204 NAVAL STATION
Address		4	0.038	P & S BLDG P & S BLDG./RM.17-501
Address		4	0.038	RECEIVING RECEIVING
Address		4	0.038	STE CATHERINE 3755 CH COTE STE-CATHERINE
Address		3	0.029	104 25 STOCKHOLM 104 25 STOCKHOLM
Address		3	0.029	B 8545 ARJONS DRIVE "B"
Address		3	0.029	BLDG RM BLDG: RM:
Address		3	0.029	CHEMISTRY CHEMISTRY DEPT., MH-227
Address		3	0.029	DEPT OF BIOLOGY DEPT. OF BIOLOGY
Address		3	0.029	DO NOT USE ***** DO NOT USE ****

Figure 27: Embedded Logic Report

This next textual report is an example of a simple match analysis, referencing addresses and some of the valuable information that can be found, such as embedded business rules or records matched across sources.

```
Duplicate NA Div    0202-01 UNITED HOSPITAL                                    333 N SMITH AVE     SAINT PAUL MN 55102
Duplicate NA Div    55101   UNITED HOSPITAL/CHILDREN HOSP ***PLEASE USE ACCT# 3401***  333 N SMITH AVENUE  SAINT PAUL MN 55102
```

Embedded Business Rule In One Duplicate

```
Duplicate  NA Div     177011       TRUMAN MEDICAL CENTER WEST       2301 HOLMES ST    KANSAS CITY   MO 64108
Duplicate  PubSector  21373000     TRUMAN MED.CTR. WEST             2301 HOLMES ST.   KANSAS CITY   MO 64103
Duplicate  NA Div     004600-20    TRUMAN **DO NOT USE ACCOUNT**    2301 HOLMES       KANSAS CITY   MO 64108
```

Records Matched Across Source Systems

```
  Duplicate   NA Div    323271      TEXAS CHILDRENS HOSPITAL      6621 FANNIN ST     HOUSTON  TX 77030
      ?       NA Div    140200-01   TEXAS CHILDRENS HOSPITAL      6519 FANNIN ST     HOUSTON  TX 77030
  Duplicate   NA Div    24221       TEXAS CHILDRNS HSP CLIN CRE   6621 FANNIN ST     HOUSTON  TX 77030
```

Figure 28: Example of a Simple Match Analysis

There are many examples of close matches that may require user input. Is this a match? Maybe, it depends on the role (partner type) of the customer – bill to, ship to, etc.

5. Rational Data Architect

Rational Data Architect (RDA) is IBM's strategic tool to address various modeling and design requirements within the information management domain. It is built on the Rational Software Development platform (and implicitly on Eclipse). This gives the user the option to focus on one individual activity and only use a particular tool such as RDA. It also gives the user the option of having the tools tightly integrated as if it was a single tool, e.g. using RequisitePro for requirements, Rational Software Architect for software design, and RDA for data modeling.

The following diagram shows an overview of the major capabilities of RDA.

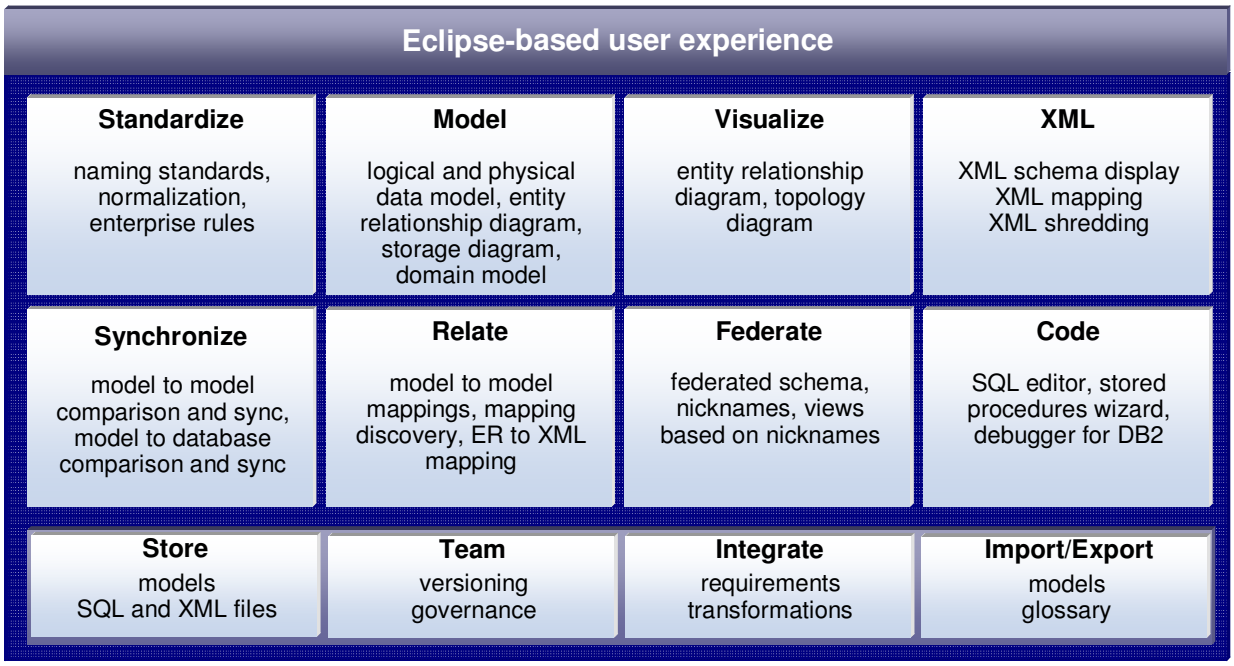


Figure 29: RDA Capability Overview

As shown in the product overview diagram above, RDA has a set of core functions that go beyond just data modeling. We will first introduce all the major capabilities briefly. We will highlight RDA functions that are relevant for the SOMA 3.1 Information Discipline and describe them in more detail in following sub sections.

- **Standardize – Supporting the Business Glossary**

  Data naming standards promote a common understanding of data terms, sharing of those defined terms across organizational boundaries, and reduction of data redundancy through the consolidation of synonymous and overlapping data terms.

  In Sect. 5.2, we will describe the Glossary Model in RDA in more detail and how it supports to create and enforce a Business Glossary as introduced in the SOMA 3.1 Information Discipline.

- **Model – Supporting Conceptual and Logical Data Modeling**

  RDA provides extensive capabilities to define data models on a logical and physical level and to transform / convert between them. RDA distinguishes between the following models:

  – **Glossary Model**
  A glossary model is a model that describes the names and abbreviations that an organization allows for data objects.

- **Logical Data Model**
  A logical data model is a model that is not specific to a database. It describes entities about which an organization wants to collect data and the relationships among these entities.
- **Domain Model**
  A domain model describes the domain types that an organization allows and their constraints. Atomic domains can be stored in a domain model or as part of a logical data model.
- **Physical Data Model**
  A physical data model is a database-specific model that represents relational data objects (for example, tables, columns, primary keys, and foreign keys) and their relationships. A physical data model can be used to generate DDL statements which can then be deployed to a database server. For some database targets, you can also add storage objects to your physical data model such as table spaces and buffer pools.

As mentioned above, we will describe in Sect. 5.2 how RDA supports the specification of a Business Glossary. In Sect. 5.3, we discuss how you can specify conceptual data models in RDA, even though RDA does not have a dedicated conceptual data model artifact. Developing and maintaining logical data models follows in Sect. 5.4.

One of the major advantages of a data modeling tool and in particular RDA is that you can transform from a logical data model to a physical data model and vice versa. This allows you to model your data on different levels of abstraction and easily switch between them and keep them consistent. In the traditional data context (such as data warehousing, database design, etc.), the physical data model is probably the most important model as it defines the actual structure of your persisted database. Its design has direct impact on many important non-functional requirements such as performance, scalability, etc. However, in the SOA context and in particular in the SOMA 3.1 Information Discipline, we concentrate more on the canonical (conceptual and logical) design aspects. The primary goal of data modeling is to ensure alignment and consistency between the logical data model and the message model. We will therefore not describe the RDA capabilities to support the physical data modeling in more detail.

- **Visualize – Supporting Conceptual and Logical Data Modeling**

  Data models become quickly very large with many entities and relationships between them. RDA visualizes data modeling related artifacts in various types of diagrams.
  - Data Model Diagrams display logical and physical data models
  - Storage Diagrams visualize storage related artifacts of a physical data model
  - Topology Diagrams allow to illustrate the database topology

  Since the SOMA 3.1 Information Discipline focuses mainly on the concpetual and logical data modeling, we will describe in Sect. 5.3 and Sect. 5.4 the various visualization options for those models.

  An important capabilty of RDA is that you can connect to a live database from RDA and visualize the data model and reverse engineer a logical model from the given physical model.

- **XML**

  XML is the standard language to define a service specification (in WSDL) and its related messages (in XML schemas). Some of the data that is being processed, shared, and exchanged by services will need to be persisted in a database. RDA allows to visualize an XML schema definition and to map the XML schema to data model in which the XML may need to be persisted. If required, RDA assists in shredding the XML data into a relational form.

- **Synchronize**

  The IT infrastructure of an organization is dynamic and changes over time. Previously separated databases may need to be merged due to mergers and acquisitions, the replacement of multiple systems by a new system, or the adoption of an industry standard for a particular scope, etc. In all

those cases, we need to combine data models into a single model, compare the models and then synchronize them. We will describe some of the aspects of this approach in Sect. 5.5.

- **Relate**

  The SOMA 3.1 Information Discipline includes a top-down path to desing services based on business needs, process decomposition etc. as well as a bottom-up path to leverage existing assets. Both paths meet in the middle when specify the services and in particular when we make the realization decisions. The specification of the canonical data model and message model should not be constrained by the design of current legacy database since it represent the future and longer-term representation of entities and services that may not map exactly to the current enviornment. However, the services will need to be realized in large by the given infrastructure. That requires a mapping of the services, their message models, and in many cases the corresponding logical data model to the underlying sources as defined in the SOMA 3.1 Information Disicpline. We will describe in Sect. 5.6 how you can map various data models in RDA.

- **Federate**

  As described above, RDA allows to specify a mapping between a canonical data model – which is independent of one particular source – and one or more source models. The purpose of this model can be to join the information that may reside in multiple sources together in a virutal federated view and to expose it as a service. In Sect. 5.6, we illustrate how you can define a federated view in RDA and generate the necessary statements fhat you can deploy on WebSphere Federation Server or DB2.

- **Code**

  Once a data model is specified, RDA allows to create valid SQL code for DB2 so that you can deploy the data model on a database. RDA also allows you to develop valid code for java applications or to develop stored procedures.

- **Store**

  Artifacts (models) that are created in RDA can be stored within the RDA tool and as part of the Rational Software Development Platform or also in the Unified Metadata Platform of IBM Information Server (see Sect. 2.2). If you are using a versioning system – such as Rational ClearCase – then you can also store your models there to keep track of who is doing what (see also next bullet).

- **Team**

  When you collaborate in a team with other developers and architects, you may need to share your models and access the models of other team members. RDA integrates with the Rational ClearCase to facilitate this team collaboration. RDA allos to track your changes and can be integrated with Rational ClearQuest.

- **Integrate**

  In Sect. 5.7, we will illustrate how RDA integrates with various tools in the traditional database and data modeling context as well as in the SOA context. RDA also integrates with tools that we do not mention in Sect. 5.7 such as DB2, DB2 DataWarehouse Edition, WebSphere Federation Server, IBM Change Management Expert.

- **Import/Export**

  RDA can import and export artifacts from a variety of systems, ERWin being one of the primary tools that RDA can exchange data models with.

One of the primary functions of RDA is clearly to assist the data architect with data modeling. Data models define how real-world entities are represented in an IT environment. RDA provides extensive data modeling

capabilities and specifically focuses on data models that may be persisted in a (relational) database. Data models can be defined on different levels of abstraction. RDA supports the specification of logical data models – which we have categorized further in the SOMA 3.1 Information Discipline White Paper into a conceptual data model and logical data model – and physical data models and the relationship / transformation between them. RDA can be used to develop data models in a top-down manner as proposed in the SOMA 3.1 Information Discipline White Paper or bottom up from existing databases. RDA allows you to directly look at your existing databases and develop your data model by reverse engineering your existing data assets.

In the following sections, we will describe the core capabilities from RDA that are relevant for the SOMA 3.1 Information Discipline in more detail.

5.1 Getting Started

After you have installed RDA, you will first see a welcome screen.

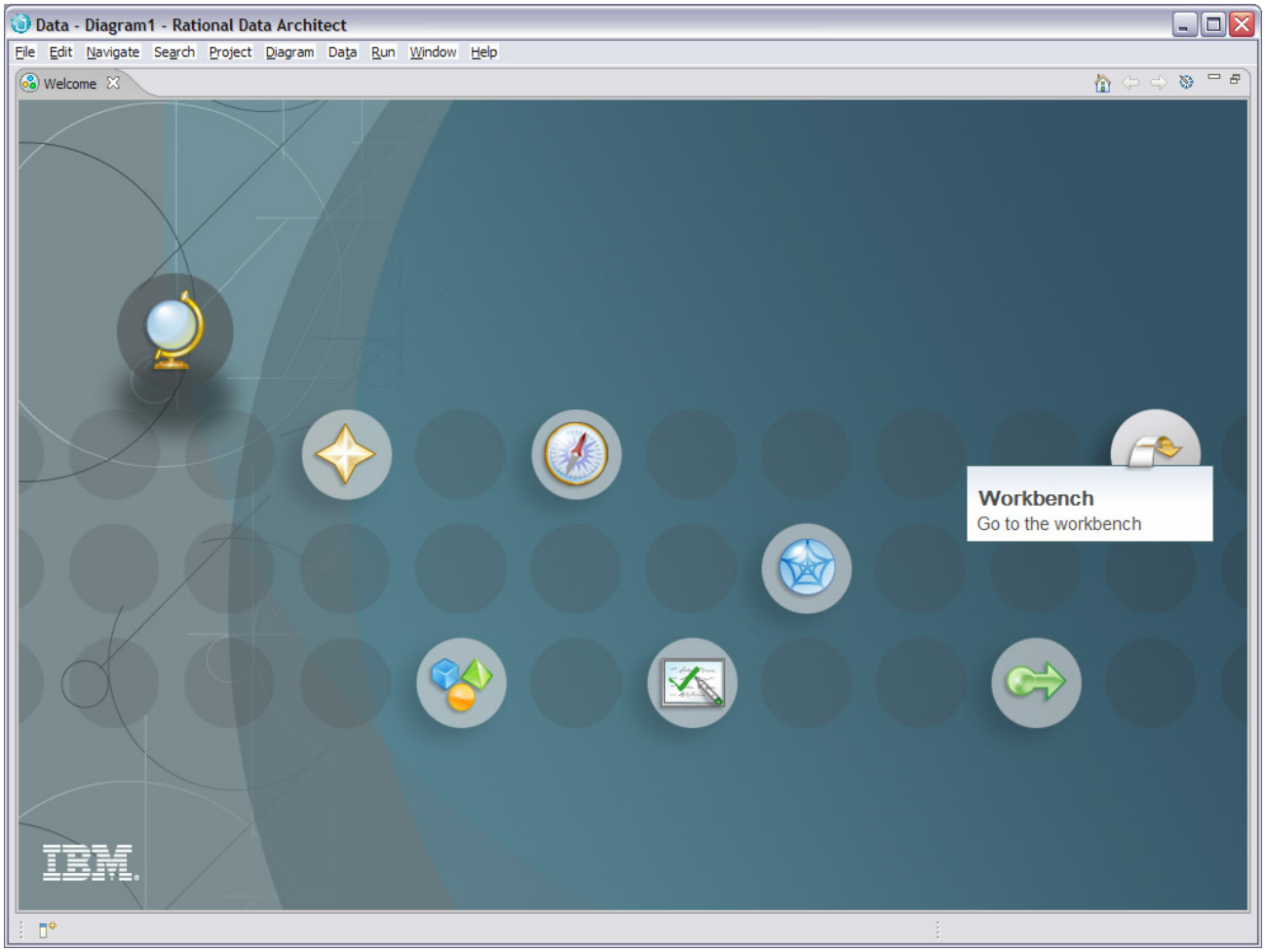


Figure 30: RDA Initial Welcome Screen (go to: Help → Welcome)

You have various topics to select from such as overview, samples, tutorials, etc. Depending on which topic you select, you will get to a page for a specific topic. The screenshot below shows the page when you select to see the overview.

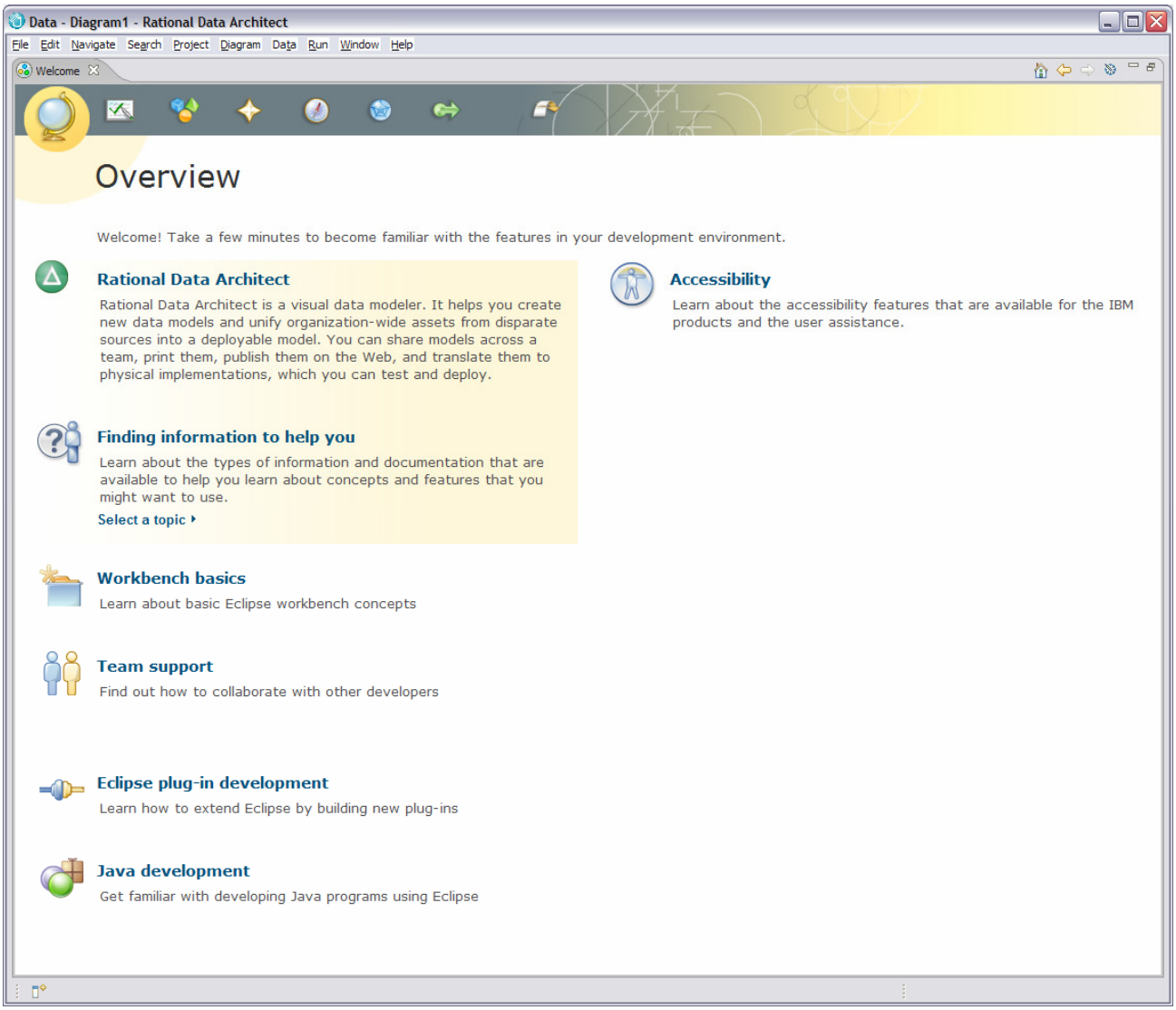


Figure 31: RDA Overview Help Screen (go to: Help → Welcome verview)

If you follow the link 'Finding information to help you', you will see instructions on how to access details help, which is basically when you go to: Help → Help Contents. In this paper, we will now show you detailed instruction for how to use RDA but focus on some major RDA capabilities that are relevant for using the product to support the SOMA 3.1 Information Discipline activities.

After you may have read some tutorials on those welcome screens, you can access the RDA workbench by selecting that menu item in the welcome screen. RDA is based on the Rational Software Developer Platform and you will see the familiar Eclipse-based workbench as shown below.

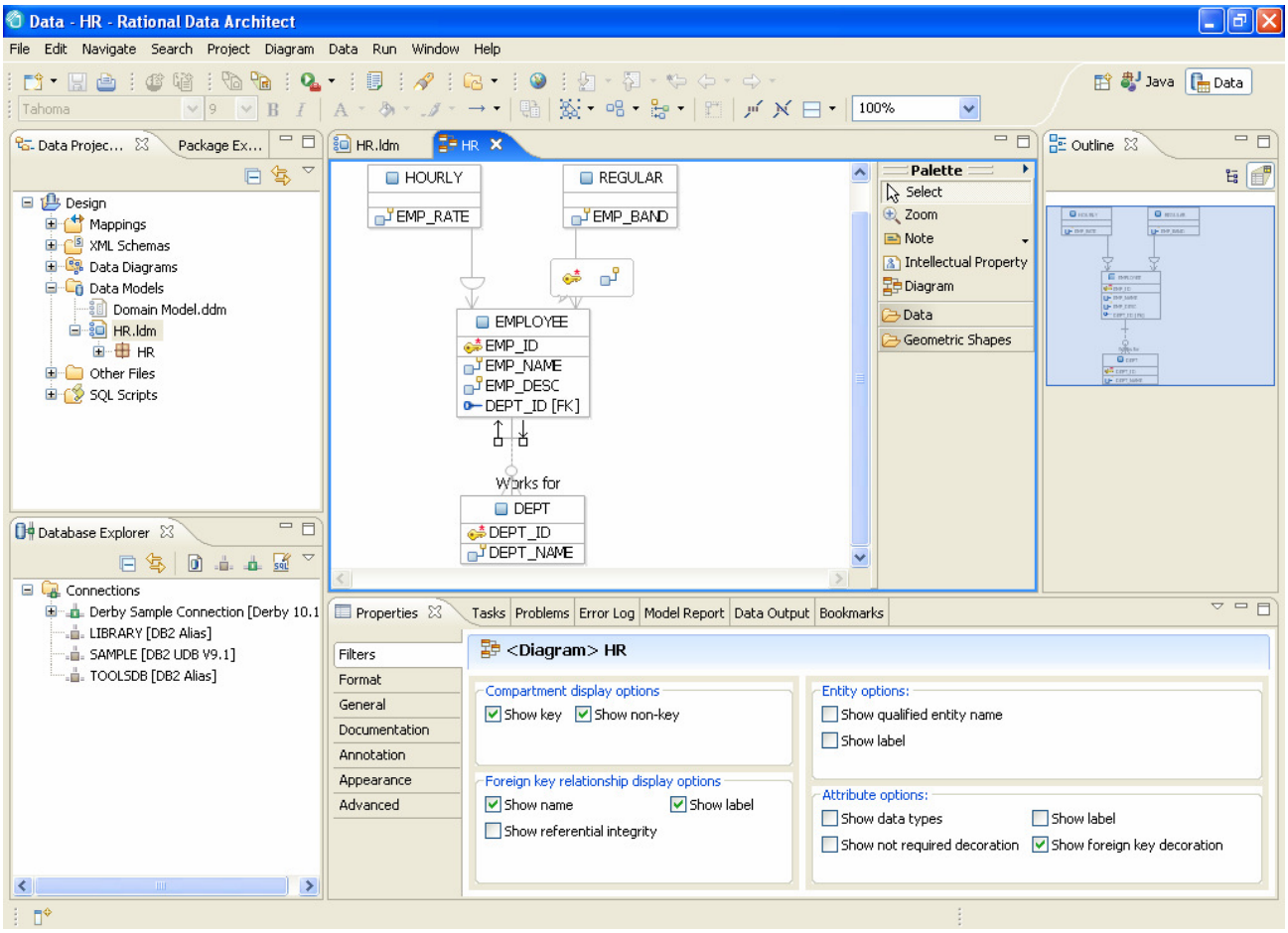


Figure 32: RDA Workbench Standard View in the Data Perspective

In order to get started, you can either import an existing example from the initial help screen or just start a new project. The primary perspective in RDA for a data architect is the data perspective (Window → Open Perspective → Data).

5.2 Glossary Modeling

The SOMA 3.1 Information Discipline White Paper [1] articulates the importance of a consistent business terminology that conforms with your overall SOA project terminology. We have positioned WebSphere Business Glossary (see Sect. 3) as the strategic software solution from IBM to define the business terminology in SOA and to share it broadly across various tools. One of the primary tools that can be integrated with WebSphere Business Glossary is RDA. During the data modeling activity, an architect should follow the business terminology as defined in the glossary.

RDA can import business terminology from WebSphere Business Glossary through the unified metadata management platform from IBM Information Server (see Sect. 2.2). Alternatively, the business terminology can be defined in RDA manually. In both cases, the information will be captured in a glossary model.

The RDA help content includes details on how to create and maintain a glossary at: Help → Help Contents; navigate in the contents tree to: 🗁 Rational Data Architect ↳ Overview of data modeling ↳ Modeling naming standards ↳ Glossary models:

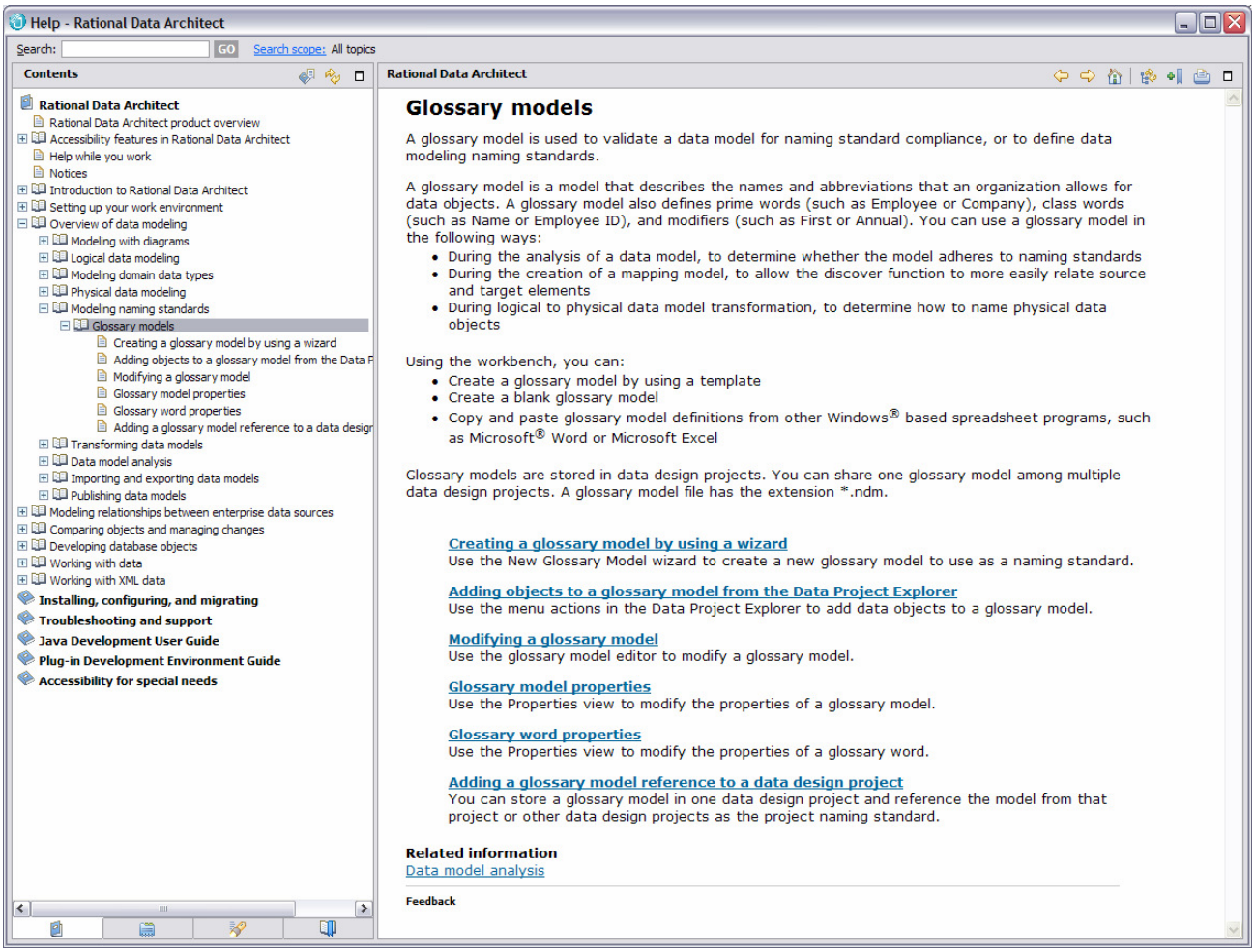
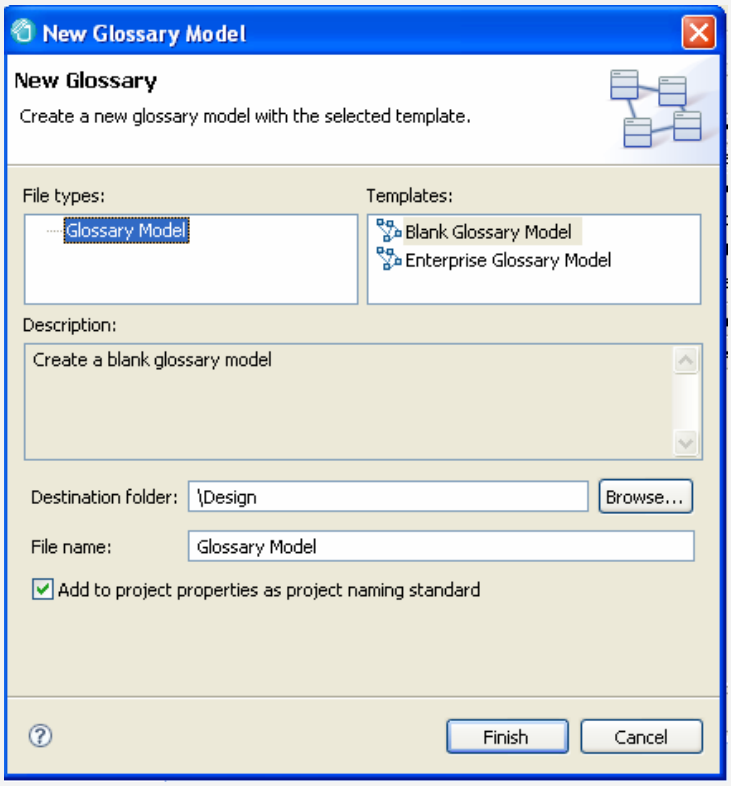


Figure 33: Help Contents Screen for the Glossary Model

No matter if you import an existing glossary definition or if you want to define a glossary in RDA, the designated area in RDA to document a glossary is in the so-called Glossary Model.

Create a Glossary in RDA: File → New → Glossary Model



Besides the general advantages of defining a business glossary as motivated in the SOMA 3.1 Information Discipline White Paper [1], it is in particularly valuable to incorporate the business glossary in your data model. The RDA Glossary Model allows to define and implement naming standards and use them directly with your various data models (conceptual, logical, physical models). The data architects has direct and easy access to all the terms when creating new entities and attributes and can ensure the consistency of naming standards between the Glossary Model and the data models.

After the Glossary Model is created for a particular project, you can then create one to many glossaries which can be arranged in a hierarchical form. Each glossary contains a list of so-called Words (that define terms). A Word can be defined by a set of attributes such as

- General information: name, abbreviation(s), type, modifier, status, abstract
- Related words
- Synonyms
- Description
- Documentation
- Annotation

The end result is that your glossary gets defined directly in RDA and can be used when creating your models.

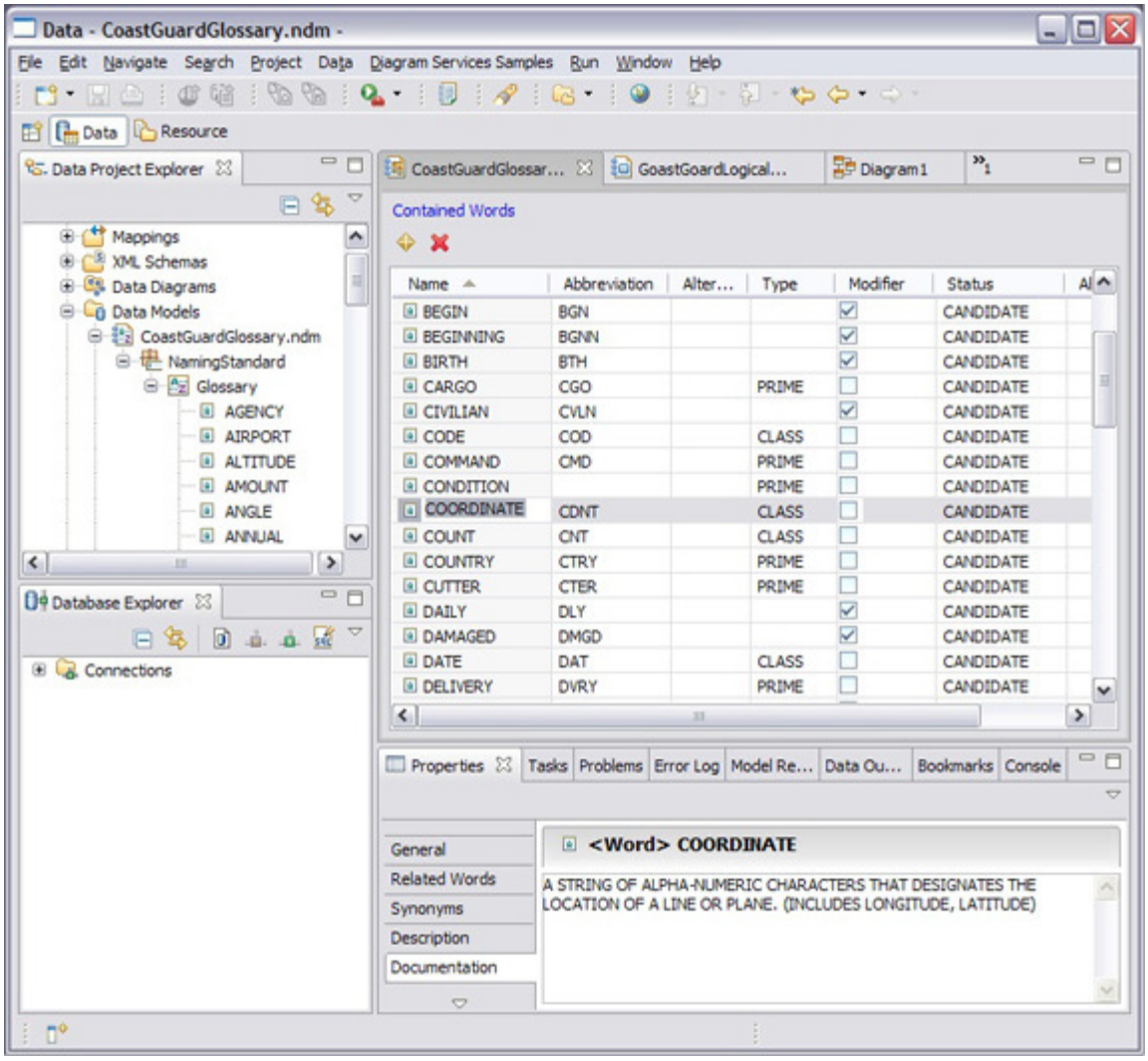
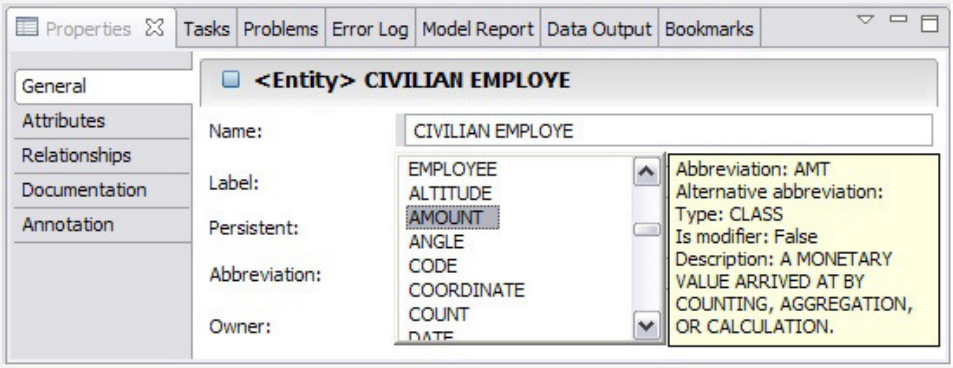


Figure 34: Screenshot of a Glossary Model

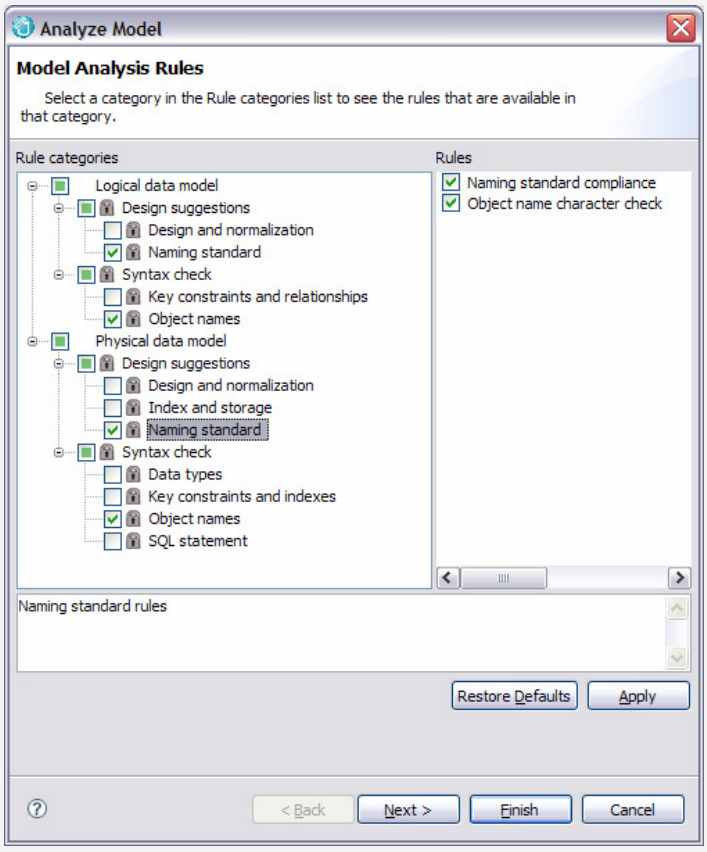
Once you have defined your Glossary Model, you can easily follow the naming standards when you define your data model elements such as entities and attributes. RDA's content assist (ctrl+space) key displays the defined Words when you select an entity or attribute name as shown below.

Glossary Content Assist in RDA: *(for selected name)* CTRL+SPACE key



After the data model is defined, an architect or database administrator can analyze the compliance of the various models based on the definitions in the Glossary Model as shown below.

Analyze Glossary Compliance in RDA: *(for select model)* Data → Analyze Model



When performing this analysis, RDA compares the selected data model with the Glossary Model and indicates violations to the definitions that an architect can then resolve.

5.3 Conceptual Data Modeling

As defined in the SOMA 3.1 Information Discipline White Paper [1], the conceptual data model defines the major business entities, generalizations of those entities, and relationships between the entities. It may include the specification of some attributes of the entities but mainly to define the scope and not to provide a comprehensive or complete attribute definition. In order to decompose a possible large enterprise data model, it may be necessary to define subject areas, possibly even decomposed on two levels.

In some cases, architects use drawing tools such as Microsoft Visio to design conceptual data models. The advantage of drawing tools is that you can easily and quickly create a diagram. The disadvantage is that because such tools do not enforce any rules, your drawing may look nice but may not be consistent with the technical artifacts in your enterprise. Drawings cannot always be converted to technical artifacts, in particular when an entity is just represented as a box and not as an entity. The ease of use of drawing tools comes also with the price of ambiguity. For example, is a box an entity, a subject area, or just a drawing object to indicate an organizational unit?

The use of modeling tools, and in particular RDA, allow to specify a conceptual model and then to take this model and then transform it and refine it further consistently on a logical and possibly even physical data model level.

To start defining a conceptual data model in RDA, you will need to create a project – or use an existing project – and to create or use a logical data model. RDA only differentiates between two types of models: a logical data model and a physical data model. Since a conceptual data model is a specific type of a logical model, we will use the logical data model definition in RDA to define a conceptual data model.

Before we describe how you can use RDA to support the activities as defined in the SOMA 3.1 Information Discipline, you can access more detailed information on how to use the RDA tool when you go to: Help → Help Contents; navigate in the contents tree to: 🗁 Rational Data Architect ↳ Overview of data modeling (↳ Logical data modeling) as shown in the screenshot below:

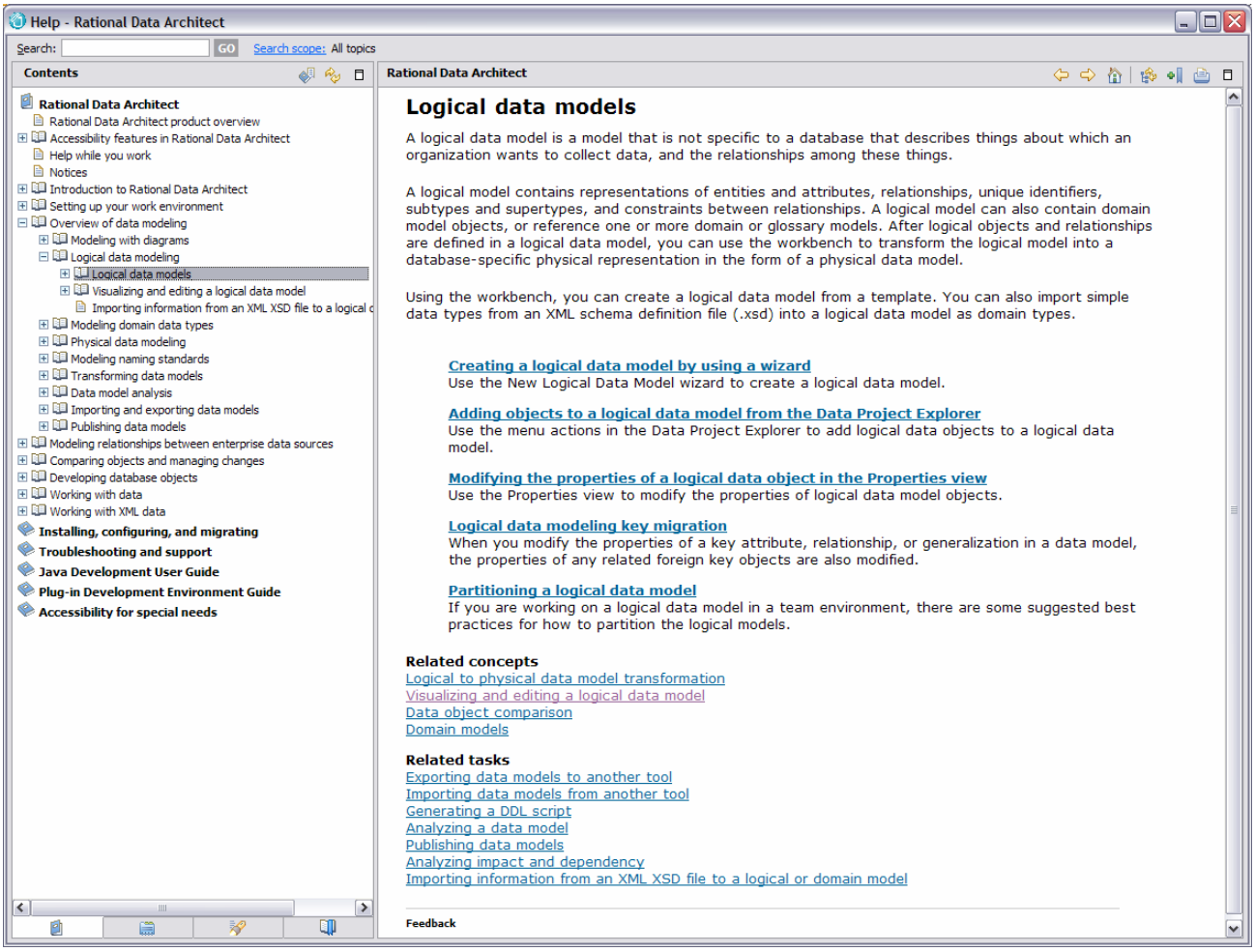


Figure 35: Conceptual Data Modeling in RDA (by Using the Logical Data Model Support)

As described in the SOMA 3.1 Information Discipline White Paper, you may want to start the conceptual data model by defining subject areas in which you group your entities. As shown in the screenshot below, you can define subject areas in RDA which are called diagrams. You can define relationships between diagrams.

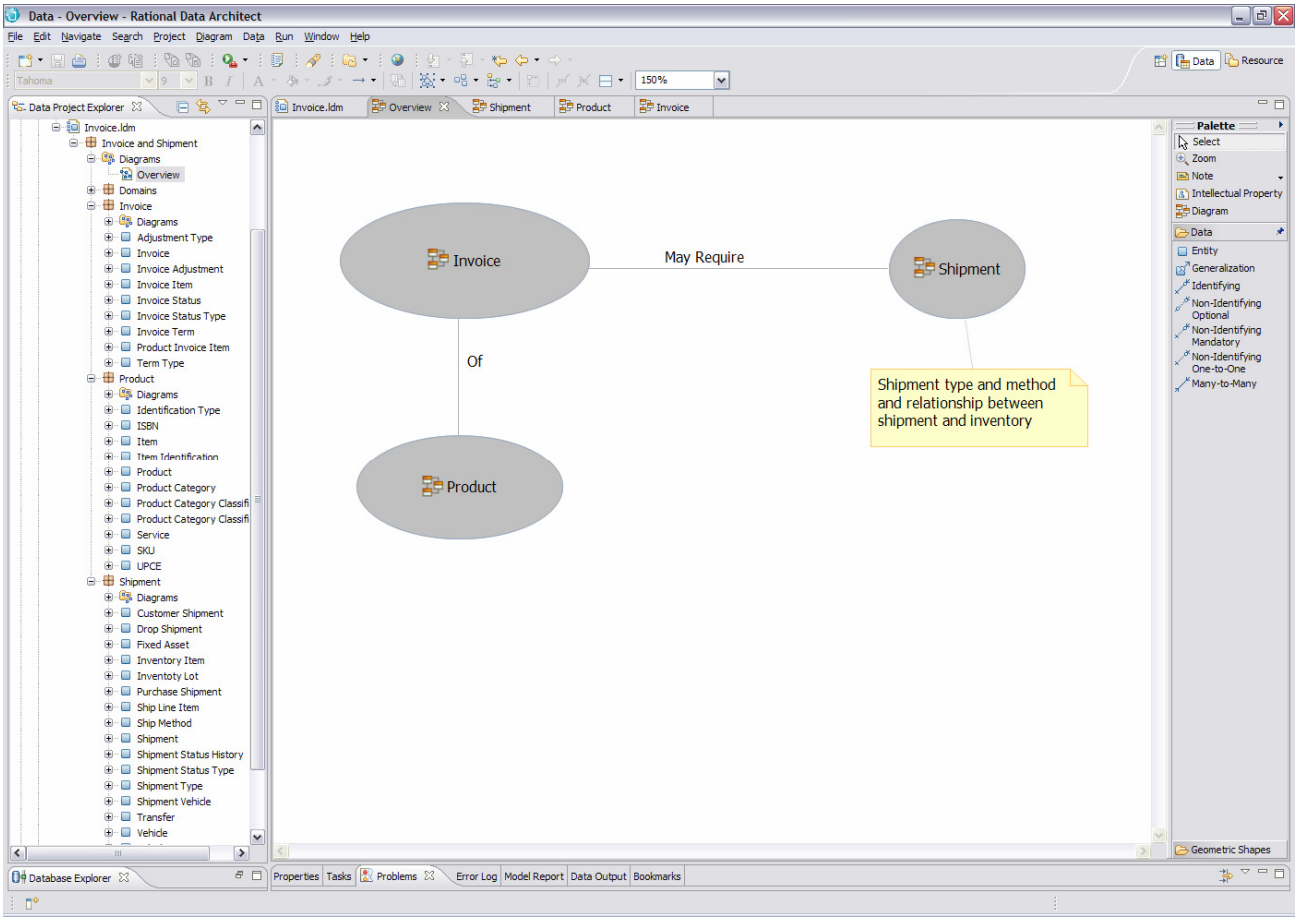


Figure 36: Subject Areas / Diagrams in RDA

You can define a diagram in RDA by simply using the palette which is in the screenshot above (Figure 36) on the right side next to the display of the diagram overview. When you double click on a diagram in RDA, it takes you to its definition. For example, if you double click on the diagram 'Invoice', it takes you its layout definition as shown in the screenshot below.

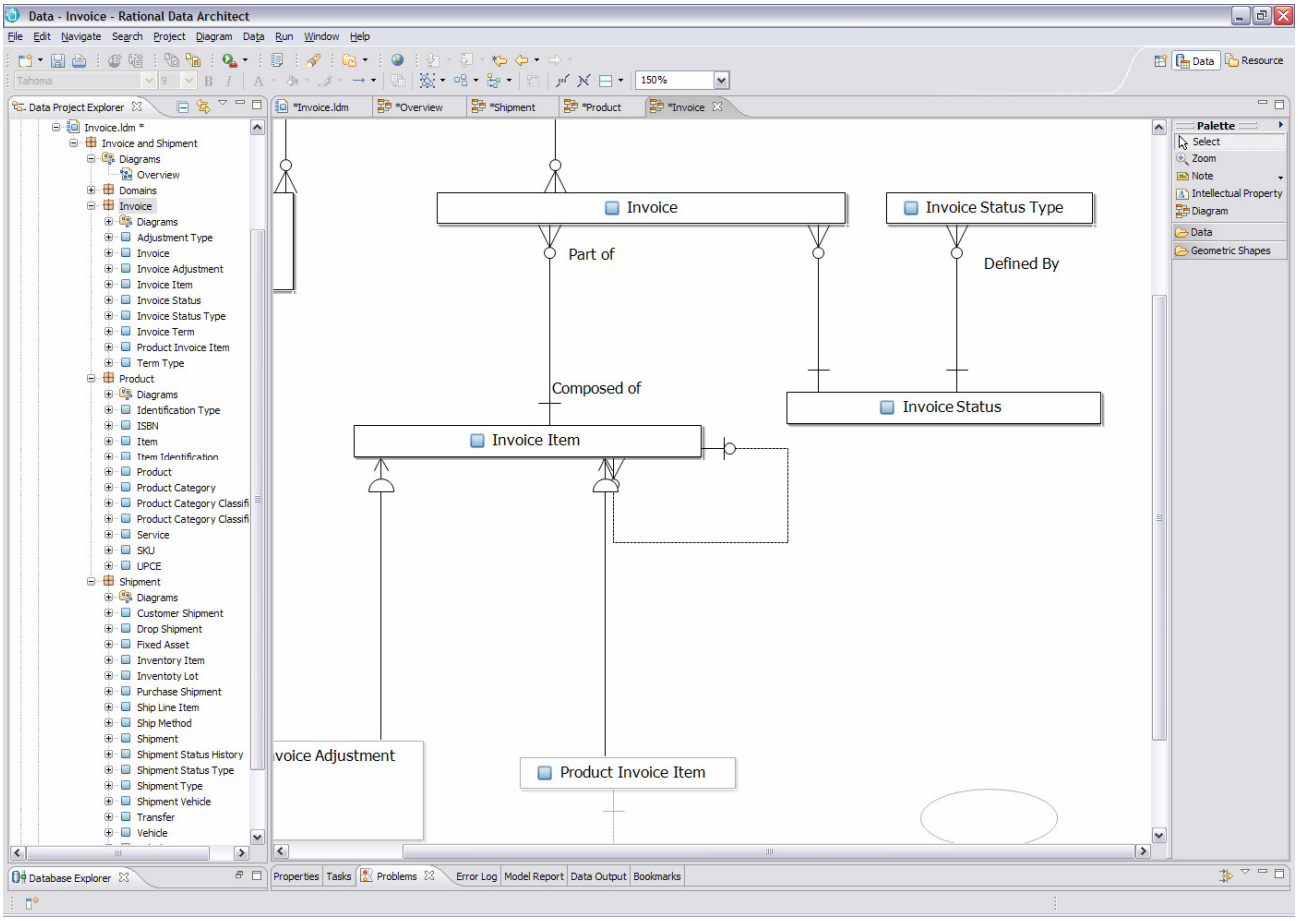


Figure 37: Example: Invoice Diagram

There can be a single diagram for a conceptual model which visualizes all or some of the entities defined or multiple diagrams that are arranged in a hierarchical form in the sense that one diagram can include a list of diagrams which include entities for further diagrams.

In each project you can then define a set of entities[1]. You can create entities either graphically in a diagram by using the palette or by adding an entity object for a selected package. It is important to understand that a diagram simply visualizes the entity definitions. It is not the scope to define the entities and relationships. That means, you can define an entity in a package and then drag it onto one or more diagrams. If you drag an entity onto the diagram that has a relationship to another entity that is already on the diagram, RDA will automatically draw that relationship.

For each entity that you create, you can specify attributes and relationships. RDA does not enforce the specification of attributes for logical data models. You can also define generalization relationships between two entities. This will then make one entity the super-type (e.g. party) and the other entity the sub-type (e.g. customer).

As shown in the screenshot above (Figure 37), you can either show or hide attribute definitions.

[1] An entity is actually defined within a package. A project can have one or more packages.

5.4 Logical Data Modeling

Logical data models in RDA are not related to any specific database (vendor, version, etc.) but rather give you a representation of your business entities and how they are all related to each other. A logical data model contains representations of entities and attributes, relationships, unique identifiers, sub-types and super-types, and constraints between relationships. A logical model can also contain domain model objects, or reference one or more domain or glossary models.

Following the top-down approach as described in the SOMA 3.1 Information Discipline White Paper, you develop the logical data model based on the SOA project requirements and by refining the conceptual data model. RDA allows you to add attributes to the entities by following the naming standards as defined in the glossary model. When you define data types in your logical data model, you can use consistent definition by creating a domain model. The domain model allows to specify data types and associated constraints that you can then use in attribute definitions. For example, you can specify an atomic domain (i.e. a data type) 'ID' which may be of the base type DECIMAL with precision 7 and scale 0. Another atomic domain might be 'Description' that you will abbreviate 'Descr' of the base type VARCHAR with the length 256. When you define an attribute, e.g. the attribute Cust_ID in the entity Customer you can select the predefined type ID. You can leverage the same data type definition for another attribute in another entity.

RDA supports the definition of primary keys and foreign keys. Once you start adding an attribute to a primary key (or create a primary key), RDA ensures the key migration to the depend entities. For example, if you add a primary key to a parent, the child in the identifying relationship to this parent will receive the corresponding foreign key with the required naming convention. The key migration is supported for a wide range of scenarios as defined in the help contents of RDA.

After entities and relationships are defined in a logical data model, you can use the RDA workbench to transform the logical model into a database-specific physical representation in the form of a physical data model. Physical data models are database-specific models that represents relational data objects (for example, tables, columns, primary and foreign keys) and their relationships. A physical data model can be used to generate data definition language (DDL) statements which can then be deployed to a database server. RDA allows you to make changes to the model, view the DDL that would negotiate this change and then run the SQL queries directly on the server or save it as a file to give to someone else in your organization to run.

5.5 Compare and Synchronize Data Models

You can select a model in RDA and compare it with its original source in order to determine any changes or to another model. This functionality is helpful for models that are relatively close but not identical. RDA allows you to quickly see the difference and to synchronize the model by applying the definition in one model to the other.

For more detailed information on this topic, go to: Help → Help Contents; navigate in the contents tree to: 🗁 Rational Data Architect ↳ Comparing objects and managing changes.

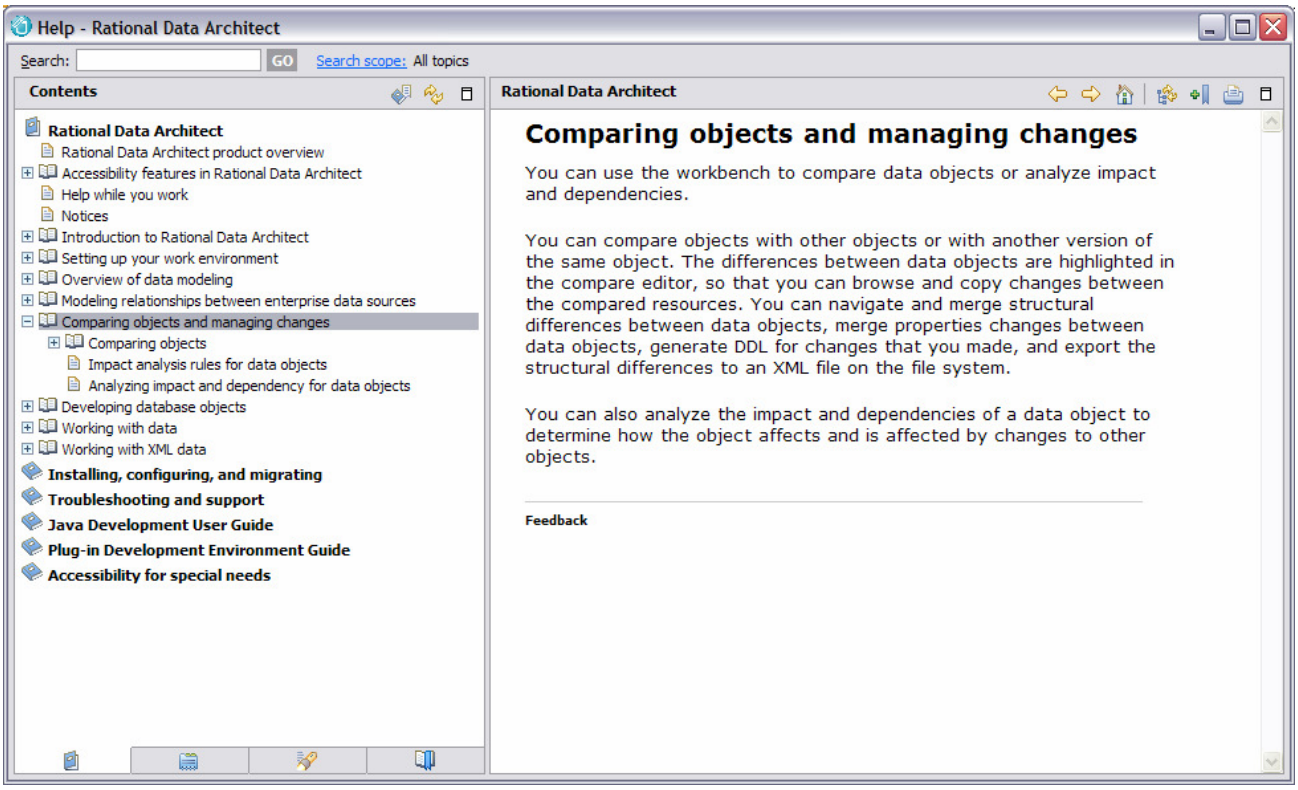


Figure 38: Compare and Synchronize Data Models (and Objects)

5.6 Relate and Map Data Models

The RDA tooling can assist you with relating and mapping one or more source models to one or more target models. You may have multiple schemas and may want to specify how they are related. For example, you may have developed a canonical data model on a logical level and may now want to map this canonical model to the data model of a specific system. Or you may have multiple data models of existing databases and want to integrate them in a federated data model that you have specified. Maybe you have an XML schema that you want to map to a given relational schema. You may or may not know exactly how to map between the various models.

The RDA tooling can assist you in this effort. As a first step, you will need to create a mapping model in RDA. When you follow the wizard, you will be able to enter various source and target data models. The source models are then listed on one side of a mapping window and the target models on the right side. RDA supports a mapping discovery tool that offers multiple algorithms on how to discover the mapping between one or more source models and one or more target models.

Creating the mapping file is as easy as right clicking on ‘Mappings’ which are part of your project, New → Mapping Model. Then you can drag and drop the object that you want into the source window and the object that you want into the target window and perform your mapping.

The result of the discovery of relationships may look similar to the following screenshot.

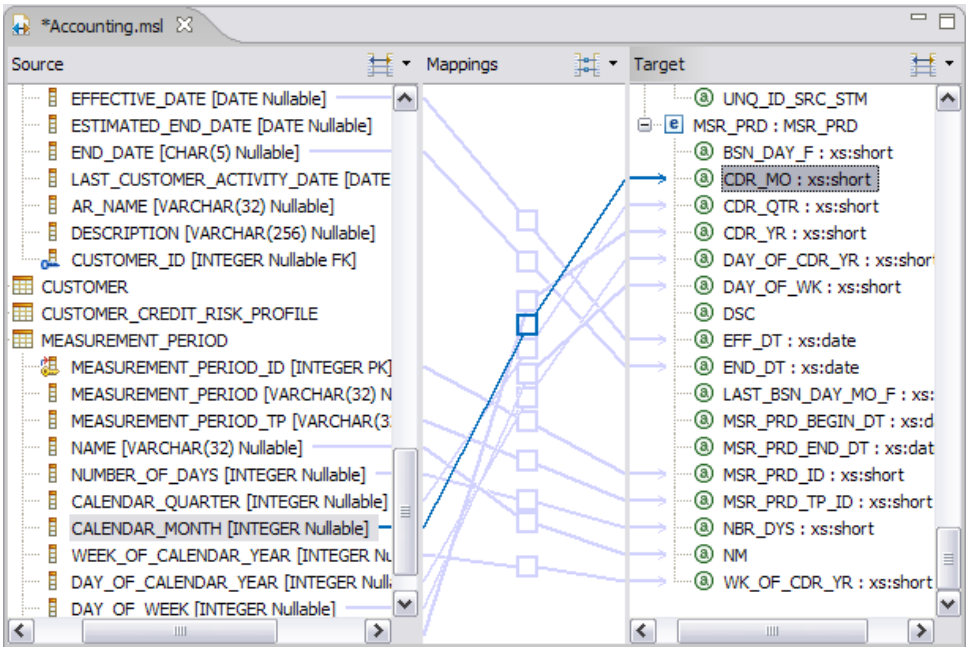


Figure 39: Mapping of an XML Schema with a Relational Model

You can either explicitly specify a mapping between to related elements or let RDA discover the mapping for you, based on various algorithms that you can select. For relationships that RDA has discovered, you can then either accept or reject the discovered and proposed mapping. You can specify the mapping in more detail by using an expression builder which helps you to build SQL fragments.

Once the mapping is completed, you can save it as is just as a mapping artifact. In case you have mapped multiple physical models of source databases to the model that you want to use as a federated data model, RDA can generate the necessary SQL scripts to map the source models to the federated model.

For more detailed information on this topic, go to: Help → Help Contents; navigate in the contents tree to: 🗁 Rational Data Architect ↳ Comparing objects and managing changes.

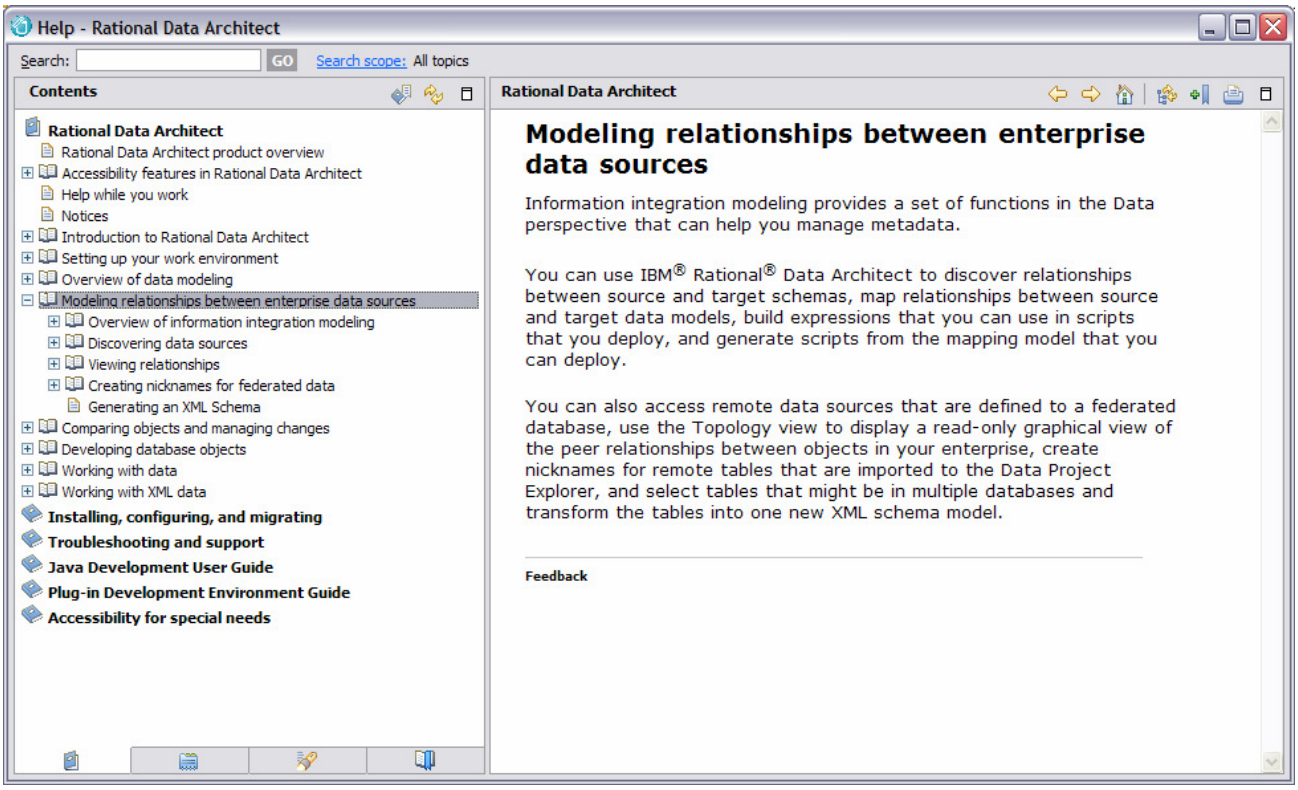


Figure 40: Relate and Map Data Models Support in the Help Contents

5.7 Integration of Rational Data Architect With SOA

5.7.1 WebSphere Business Glossary

As described in Sect. 5.2, RDA can import glossary definitions from WebSphere Business Glossary. As a prerequisite, you RDA version 6 Fixpack 1 or higher must be installed on the same client Windows® machine on which the IBM Rational Data Architect MetaBroker is installed. Then simply import the glossary definitions by following the steps in the import wizard.

5.7.2 WebSphere Information Analyzer

Data models that have been defined in RDA can be stored in the unified metadata management platform of IBM Information Server and then be made available for other tools such as WebSphere Information Analyzer to analyze the model.

5.7.3 WebSphere DataStage and WebSphere Federation Server

RDA can define the physical data models of source and target databases for WebSphere DataStage and make them available via the unified metadata management platform of IBM Information Server.

As described above in Sect. 5.6, RDA can generate SQL statements to build a federated view over existing database. These SQL statements can then be deployed on WebSphere Federation Server

5.7.4 Rational Software Architect

Through the RSA to RDA integration, you can export UML models from RSA into RDA and use them as logical data models. The opposite direction works also. You can export a logical data model from RDA as a UML model into RSA.

5.7.5 Integration with ERWin

RDA gives you also the ability to import models from other popular data modeling software programs like ERWin.

5.8 Summary

We have seen that Rational Data Architect gives you various tools to aid with your data modeling and design of your information systems. Logical and Physical data modeling gives you the ability to plan for future database design and even create the necessary SQL code needed to make those changes a reality. Capabilities like mapping will let you compare data objects from one table to another and adjust and plan before you actually make any changes. Being able to optimize business and technology terms can go a long way in enabling your organization to transform what business users want to see into real life data in your database.

Rational Data Architect gives you the tools to transform real business user requirements into data that is directly represented in your database. It can really make a difference on how efficient your organization can be.

6. Appendix

6.1 References

[1] IBM – "Service-Oriented Modeling and Architecture" – White Paper – Version 3.0

[2] Brian Byrne's "Applying SOMA 3 to IBM's Industry models", from Brian Bryne

6.2 List of Figures

