IBM® Information Management Software

A Practical Guide to Managing Reference Data with IBM InfoSphere Master Data Management Reference Data Management Hub

- Gain insight into governance, process, and security of managing reference data
- Understand the RDM reference solution architectures
- Learn RDM solution implementation

Whei-Jen Chen
John Baldwin
Thomas Dunn
Mike Grasselt
Shabbar Hussain
Dan Mandelstein
Ivan Milman
Erik A O'Neill
Sushain Pandit
Ralph Tamlyn
Fenglian Xu

Redbooks

ibm.com/redbooks

International Technical Support Organization

A Practical Guide to Managing Reference Data with InfoSphere MDM Ref DM Hub

May 2013

SG24-8084-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (May 2013)

This edition applies to IBM InfoSphere Master Data Management Reference Data Management Hub Version 10.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DataStage®	IBM®	Redbooks (logo) ®
DB2®	InfoSphere®	Tivoli®
developerWorks®	Passport Advantage®	WebSphere®
Global Business Services®	Rational®	
IA®	Redbooks®	

The following terms are trademarks of other companies:

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Reference data is a key aspect of any application integration. Today, many enterprises have no centralized enterprise governance and management over reference data. Reference data variations and inconsistencies can be a major source of data quality issues within the enterprise and cause business losses through system downtime, incorrect transactions, and incorrect reports. IBM® InfoSphere® Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) is designed as a ready-to-run application that provides the governance, process, security and audit control for managing reference data as an enterprise standard, resulting in fewer errors, reduced business risk and cost savings.

This IBM Redbooks® publication describes where InfoSphere MDM Ref DM Hub fits into information management reference architecture. It explains the end-to-end process of an InfoSphere MDM Ref DM Hub implementation including the considerations of planning a reference data management project, requirements gathering and analysis, model design details, and integration considerations and scenarios. It shows implementation examples and the ongoing administration tasks.

This book can help IT professionals who are interested or have a need to manage reference data efficiently and implement an InfoSphere MDM Ref DM Hub solution with ease.

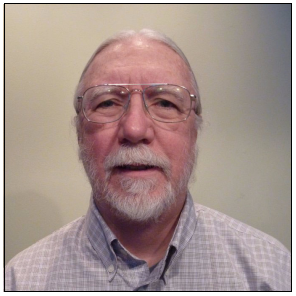
The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Whei-Jen Chen is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and IBM DB2® system administration. Whei-Jen is an IBM Certified Solutions Expert in Database Administration and Application Development, and an IBM Certified IT Specialist.

John Baldwin has over 20 years of experience in data modelling and database design. Since 2007, John has worked for the IBM Software Group in the UK and in the worldwide MDM Centre of Excellence. As a Solution Architect, he has implemented many Master Data Management and Reference Data Management solutions, principally in the banking and insurance sectors. He has created and contributed to best practices pertaining to MDM and to creating an MDM implementation methodology within IBM frameworks.

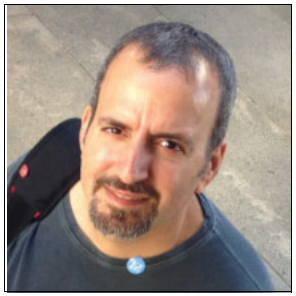
Before joining IBM, he worked in the Transportation and Logistics sector designing and managing real-time messaging systems and tracking systems; in the Broadcast Media sector building systems for capturing programming metadata; and in the Public sector designing and building screen-based examination systems and configuring and implementing accounting and payroll packages.



Thomas Dunn is an IT Architect in the IBM CIO office focused on enterprise content and taxonomy management solutions. He has been with IBM for 17 years. Most of that time has been spent designing and developing web content management systems and supporting product master data management tools. These systems deliver much of www.ibm.com marketing content and the master data is used by product development, sales, and support sites. For the last two years, he focused on developing processes and tools that improve management of taxonomies that are used to tag content. This solution has InfoSphere MDM Ref DM Hub as a core component and uses other software in the InfoSphere MDM family. Tom graduated from Rutgers University with a degree in technical theater.

Mike Grasselt is a Senior Software Engineer in the IBM Information Management Development organization in Germany. Mike is currently the Architect and Team Lead for various assets that support data integration tasks for large SAP migrations and consolidation projects. Prior to this role, he was responsible for integrating IBM Master Data Management products into SAP system landscapes and for integrating InfoSphere Information Server with IBM WebSphere® Portal, IBM Business Process Manager, and InfoSphere MDM Server in the context of a broader service-oriented architecture effort. Mike also developed an SQL extension for the Business Processes Execution Language (BPEL) and published several papers about data-aware business processes and web services. He has integrated text analysis technology with content management systems and was an author of an IBM Redbooks publication on this topic. Mike collaborates with several universities in Germany to define integration architectures, and gives lectures on IBM mainframe concepts. In addition to his technical responsibilities, he is a mentor in the IBM technical community.

Shabbar Hussain is an MDM Business Architect with IBM Software Lab Services, based in Toronto, Canada and member of a worldwide team that provides subject matter expertise, best practices, solution architecture, and design during client engagements. He has helped implement MDM solutions across various industry verticals which involved customer, product, and reference data domains. His expertise includes business intelligence and data warehousing. Shabbar holds a Master in Computer Applications degree from University of Pune, India.



Dan Mandelstein is an IBM Certified IT Architect, with over 20 years of working in the IT industry. Currently, he is working in the Master Data Management portfolio and is the Architect and Development Manager for IBM InfoSphere Master Data Management Reference Data Management Hub. He has a Bachelor of Science degree in computer science from the University of Texas and a Master degree in computer science from the University of Texas at Dallas.

Ivan Milman is a Senior Technical Staff Member at IBM, working as a Security and Governance Architect for IBM Master Data Management (MDM) and InfoSphere product groups. Ivan co-authored the leading book on MDM: *Enterprise Master Data Management: An SOA Approach to Managing Core Information*. Ivan was on the original team that created the IBM InfoSphere Master Data Management Reference Data Management (RDM) product, and Ivan is working on integrating RDM with other IBM products.

During his career, Ivan has worked on a variety of distributed systems and security technology, including OS/2 Networking, DCE, IBM Global Sign-On, and Tivoli® Access Manager. Ivan has also represented IBM to standards bodies, including The Open Group and IETF. Prior to his current position, Ivan was the Lead Architect for the IBM Tivoli Access Manager family of security products. Ivan is a member of the IBM Security Architecture Board and the IBM Data Governance Council. Ivan is a Certified Information Systems Security Professional and a Master Inventor at IBM, and has been granted 14 US patents.

Erik A O'Neill is the Product Manager responsible for the InfoSphere MDM Ref DM Hub and an evangelist on the merits of a master data management approach for the governance and stewardship of enterprise reference data. Erik helped grow the InfoSphere MDM Ref DM Hub from being a concept to being a fully supported product within the InfoSphere MDM product family.

Erik has been with IBM in senior product management roles for nine years, and worked on the Master Data Management team for most of that time. Prior to IBM, Erik worked in sales and product management roles for both startups and established enterprise software companies in US, Europe and SE Asia.

Erik graduated from Trinity College Dublin, Ireland, with MA and BA degrees in Experimental Physics.

Sushain Pandit is a Lead Engineer in the InfoSphere segment of the Information Management division in the IBM Software Group. His background is primarily in enterprise information management. His recent experience includes key contributions to development, architecture, and technical leadership of InfoSphere MDM Ref DM Hub. His experience also includes core feature development contributions to InfoSphere MDM Server and several advanced platform integration projects in the InfoSphere Information Server suite in the areas of semantics, federation, and linking. He is an IBM Master Inventor with approximately 20 US patent applications, has written several published papers, presented at conferences such as Americas Conference on Information Systems (AMCIS), International Conference on Tools with Artificial Intelligence (ICTAI) and IBM Information On Demand (IOD). Sushain holds a Master degree in Computer Science.

Ralph Tamlyn is a Consulting Senior It Architect at IBM. Ralph has worked with content management and delivery systems for most of the last 15 years. His focus has been the taxonomies and user-facing hierarchies to classify, manage, organize, integrate, and deliver that content efficiently and seamlessly. This focus has led him to develop a set of best practices for managing taxonomies and a complementary set of best practices for the classification metadata in which the taxonomies are used. Ralph is currently Lead Taxonomist for the IBM CIO Enterprise Solutions and Web Enablement team. In that role, he is responsible for unifying IBM taxonomies for all internal and external web content. He and his teammates are responsible for establishing the larger strategy to develop processes, governance, and tools to reach that unification goal.

Fenglian Xu is a Lead Software Engineer and a Senior Inventor in InfoSphere Reference Data Management at the IBM UK Hursley lab. She has 16 years of experience in IT research and development field. She worked as a lead developer in IBM WebSphere Service Registry and Repository, WebSphere Enterprise Service Bus, and WebSphere Process Server for the last seven years. Prior to joining IBM, she worked in various IT industries such as grid computing and information management, and did research in e-Science and image processing for large stack of neuron images.

Fenglian holds a PhD degree in Computer Science from University of Southampton. Her areas of expertise include information management, service-oriented architecture (SOA), and cloud computing. She also has expertise in service registry and management, Java Messaging Service (JMS), Service Component Architecture (SCA), web services, and XML technologies.

She contributes to IBM developerWorks®, to which she was invited twice to be a featured author. She has over 20 publications about IBM products and her research in IBM developerWorks, International conference proceedings, and journals.

Acknowledgement

Thanks to the following people for their contributions to this project.

From IBM locations worldwide:

- **Subramanian Palaniappan** IBM Software Developer, IBM Software Group
- **Dan Wolfson** IBM Distinguished Engineer, Chief Architect, InfoSphere and Information Management Integration and Solutions of IBM Software Group.
- **Conor P Beverland** Technical Sales Manager, MDM, North America, IBM

From the International Technical Support Organization, San Jose Center:

Mary Comianos: Operations and Communications
Tamikia Barrow: Residency Administration
Emma Jacobs: Graphics

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Reference data management

Reference data refers to data that is used to categorize other data within enterprise applications and databases. Reference data includes the lookup table and code table data that is found in virtually every enterprise application: data such as country codes, currency codes, and industry codes.

Reference data is distinct from transactional data and master data. *Transactional data* is the data that is produced by transactions within applications; *master data* is the data that represents the key business entities that participate within transactions. Reference data is also distinct from *metadata*, which describes the structure of an entity. Transactional data, master data, and reference data when combined, comprise the key business data within an enterprise. Reference data is a part of enterprise applications from the beginning of the modern computing era. However, despite this fact and the fact that it constitutes a fundamental class of enterprise data, there is relatively little focus on reference data and its importance as an enterprise data asset.

Most enterprise applications contain reference data, built into code tables, to classify and categorize product information, customer information, and transaction data. Reference data changes relatively infrequently, but it does change over time, and given its ubiquity, synchronizing reference data values and managing changes across the enterprise is a major challenge.

Ad-hoc management of reference data without a formal governance policy can create significant operational risk. For many enterprises, reference data is a major contributor to enterprise data quality problems and has a high support

cost. The demands of complying with national and international industry regulations are causing industry to rethink reference data management, and compelling the enterprises to manage and control their reference data by using sound data governance principles.

This chapter provides a definition of reference data, describes the problems that are associated with managing reference data, and provides an introduction and functional overview of the InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub). The InfoSphere MDM Ref DM Hub is designed specifically to support centralized management and governance of enterprise reference data.

1.1 What is reference data

A simple definition of reference data is that *reference data is data that is used to categorize other data within enterprise applications and databases*. The database tables that store reference data within enterprise applications are usually referred to as lookup, code, check, or domain tables. Reference data is typically defined with a code and a description, and has a set of domain values, that is, a list of allowed values. Reference data is read-only data that is used by transactions but not changed or modified by those transactions.

Reference data can take the form of a flat list, for example, the list of US states, or can have a hierarchical structure over the code values, for example, a geographic hierarchy that includes country, state, and city. Reference data is used to classify and categorize transaction and master data.

Reference data is widely used within enterprise applications. Typically, both transaction data and master data have many types of reference data associated with them. Business application users see reference data populating the drop-down menus and selection lists within software application user interfaces. These selection lists constrain the user's choice to one item from a list of allowed values, thus speeding data entry and reducing errors.

Reference data can range from general to being specific to industry, company, department, or even application as in the following examples:

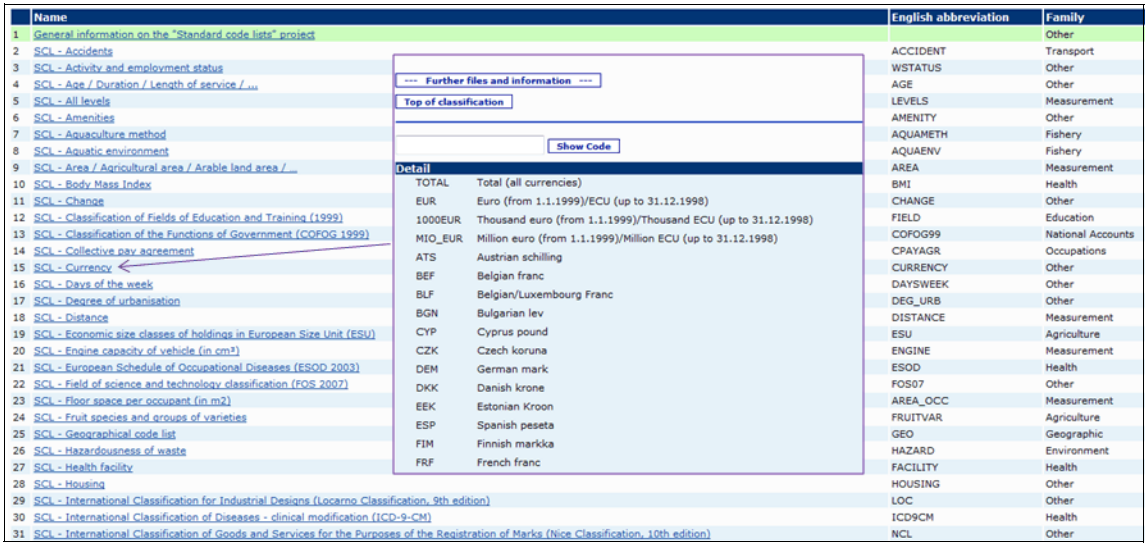
- ISO 3166-1 Country Code has applications across many industries.
- ICD-10, the international standard for classification of diseases, is related to healthcare.
- "*Fictional IBM Redbooks Company* employee expense codes" might be specific to enterprise or application.

Many reference data standards are established to support interoperability between applications and organizations, for general commerce, and to support statistical analysis of data across organizations. A wide range of organizations maintain various reference data standards and publish standard code sets and classifications, and the updates to them, in a range of formats.

The standard code tables that are published by the European Union for statistical reporting and hosted on the RAMON Metadata Server illustrate the range of common types of reference data that are used for government and statistical reporting in Europe. See the standard code tables:

http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM&StrGroupCode=SCL&StrLanguageCode=EN

Figure 1-1 shows an example of the standard code lists that are hosted on Eurostat Metadata Server, RAMON.



	Name	English abbreviation	Family
1	General information on the "Standard code lists" project		Other
2	SCL - Accidents	ACCIDENT	Transport
3	SCL - Activity and employment status	WSTATUS	Other
4	SCL - Age / Duration / Length of service / ...	AGE	Other
5	SCL - All levels	LEVELS	Measurement
6	SCL - Amenities	AMENITY	Other
7	SCL - Aquaculture method	AQUAMETH	Fishery
8	SCL - Aquatic environment	AQUAENV	Fishery
9	SCL - Area / Agricultural area / Arable land area / ...	AREA	Measurement
10	SCL - Body Mass Index	BMI	Health
11	SCL - Change	CHANGE	Other
12	SCL - Classification of Fields of Education and Training (1999)	FIELD	Education
13	SCL - Classification of the Functions of Government (COFOG 1999)	COFOG99	National Accounts
14	SCL - Collective pay agreement	CPAYAGR	Occupations
15	SCL - Currency	CURRENCY	Other
16	SCL - Days of the week	DAYSWEEK	Other
17	SCL - Degree of urbanisation	DEG_URB	Other
18	SCL - Distance	DISTANCE	Measurement
19	SCL - Economic size classes of holdings in European Size Unit (ESU)	ESU	Agriculture
20	SCL - Engine capacity of vehicle (in cm³)	ENGINE	Measurement
21	SCL - European Schedule of Occupational Diseases (ESOD 2003)	ESOD	Health
22	SCL - Field of science and technology classification (FOS 2007)	FOS07	Other
23	SCL - Floor space per occupant (in m2)	AREA_OCC	Measurement
24	SCL - Fruit species and groups of varieties	FRUITVAR	Agriculture
25	SCL - Geographical code list	GEO	Geographic
26	SCL - Hazardousness of waste	HAZARD	Environment
27	SCL - Health facility	FACILITY	Health
28	SCL - Housing	HOUSING	Other
29	SCL - International Classification for Industrial Designs (Locarno Classification, 9th edition)	LOC	Other
30	SCL - International Classification of Diseases - clinical modification (ICD-9-CM)	ICD9CM	Health
31	SCL - International Classification of Goods and Services for the Purposes of the Registration of Marks (Nice Classification, 10th edition)	NCL	Other

--- Further files and information ---

Top of classification

Show Code

Detail

TOTAL	Total (all currencies)
EUR	Euro (from 1.1.1999)/ECU (up to 31.12.1998)
1000EUR	Thousand euro (from 1.1.1999)/Thousand ECU (up to 31.12.1998)
MIO_EUR	Million euro (from 1.1.1999)/Million ECU (up to 31.12.1998)
ATS	Austrian schilling
BEF	Belgian franc
BLF	Belgian/Luxembourg Franc
BGN	Bulgarian lev
CYP	Cyprus pound
CZK	Czech koruna
DEM	German mark
DKK	Danish krone
EEK	Estonian Kroon
ESP	Spanish peseta
FIM	Finnish markka
FRF	French franc

Figure 1-1 Standard code lists (partial) hosted on Eurostat Metadata Server - RAMON

The standard code lists span a range of reference data domains, from the general, such as SCL-Languages, SCL-Currency, SCL-Days of the week, to industry-specific, such as SCL - Loading status [Transport-specific]; ICD -10 2007 [Health-specific]. Some of these lists are flat lists; others have hierarchical relationships between the codes. For example, SCL- Classification of Fields of Education and Training (1999) is hierarchical.

Standard classifications and nomenclatures such as NACE, SIC, NAICS, and many others are also listed on RAMON and are also reference data under the definition that was provided previously: *reference data is data used to categorize other data within enterprise applications and databases*.

A general characteristic of reference data is that it changes slowly relative to master data or transaction data. The list of countries in the ISO 3166-1 country code list does not change often. Changes to larger more complex sets such as ICD-10-CM are published on annual basis so they can be easily consumable. The relatively static and unchanging nature of reference data is one of the reasons that formal governance over this class of data within the enterprise is so often neglected.

A set of common-sense questions can be used to determine whether something should be treated as reference data, from a data governance perspective:

- Does it categorize other data?
- Is there a well-defined list of allowed values?
- Is it unchanged by the transactions that use it?
- Is it relatively static and slow changing?
- Is there a requirement to manage a lot of additional properties along with the value?

For reference data, the answer to the first four bullets is *yes*; for the fifth bullet, that answer is typically *no*. Although data that does not meet these criteria might also benefit from centralized governance and stewardship, such data usually has governance concerns and characteristics that differ from those compared to reference data, and usually requires separate processes, policies, and tooling for its governance.

In fact, the answers to these questions are the key characteristics that differentiate *reference data* from *master data* under the usual definition of the terms.

Figure 1-2 shows the key differences between master data and reference data.

Reference Data	Master Data
Categorizes other data. Reference values qualify/classify master data and transaction data	Key business entities that participate within transactions – the who, what and where of business transactions: Party (eg customer, business partner, employee) Product, Account, Location
Well defined set of allowed values	No predefined set of allowed values.
Unchanged by the transactions and application processes that use it. Used in read only mode	Can be created, changed and modified within business applications – eg customer and product updates
Relatively static and slow changing data	Can change frequently reflecting the fact that master data evolves and changes as a normal part of application and business processes
Relatively flat data structure and does not include many associated properties (attributes)	Can have complex data structure and many properties

Figure 1-2 The key differences between master data and reference data

Variables are one type of data that falls outside of our definition of reference data and are used across the organization, such as tax rates and daily currency exchange rates. Such *reference variables* do not categorize other data and do not have a predefined list of allowed values. However, similar to reference data, they are used by transactional applications in a read-only fashion. Reference variables are typically authored and defined externally to the enterprise and might be used across many enterprise applications. Managing reference variables and reference data presents similar problems. In designing a governance program for reference data, including reference variables as a type of reference data usually makes sense. However, reference variables have slightly different characteristics compared to reference data, and can change more frequently. For example, currency exchange rates can change daily or more frequently.

1.2 Structure in reference data: Hierarchies and relationships

Reference data often has a hierarchical structure that is defined over the reference data values. Hierarchies are created by defining relationships between the values within an individual set. You can create a hierarchy by defining relationships over the nodes of an individual set, or you can create hierarchies by

defining relationships between the values in different reference data sets. The topology of these hierarchies has different implications for the data stewardship processes.

1.2.1 Tree hierarchies

A tree hierarchy is a hierarchical structure over some or all reference data values within a set. A tree hierarchy has a parent-child relationship between the values. where each value is a node within the hierarchy. For example, within the NACE Industry codes, you see a hierarchy structure implicit within the code values themselves, representing the subcategorization of industry activities, as shown in Figure 1-3.

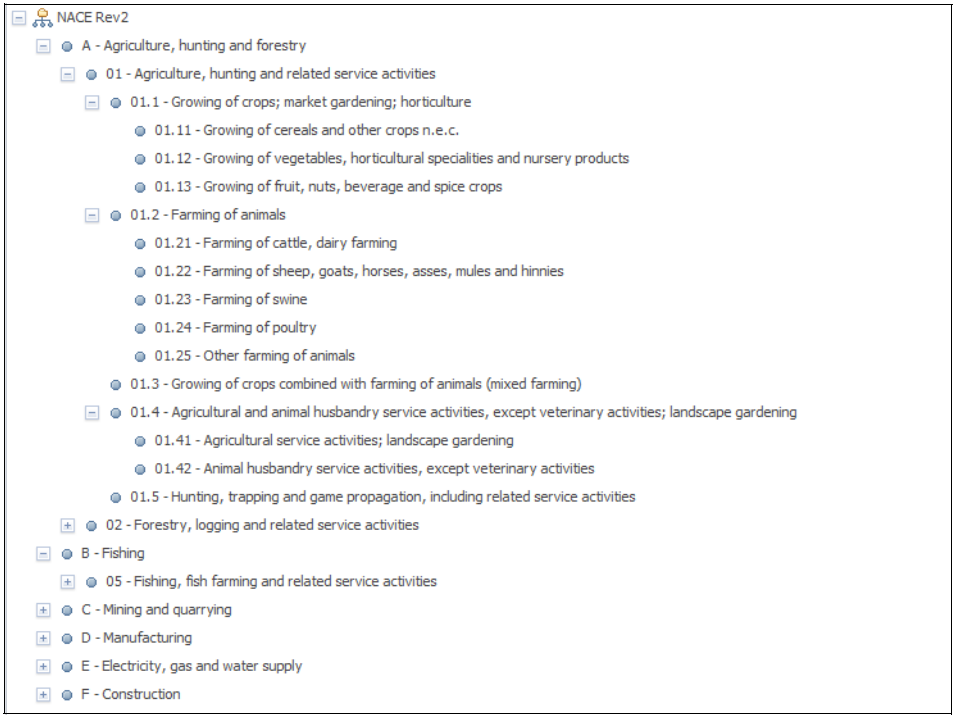


Figure 1-3 NACE Rev 2 hierarchy (partial)

1.2.2 Level-based hierarchies

With InfoSphere MDM Ref DM Hub, you can create hierarchies by defining relationships across various reference sets. Each level of the hierarchy is represented by a separately managed reference data set.

These hierarchies are referred to as *level-based* hierarchies. Level-based hierarchies are similar to tree hierarchies except that each level within the hierarchy is managed and defined as a reference data table or set in its own right. For example, Country → State is a geographic hierarchy where Country and State are usually managed as independent reference data sets.

Figure 1-4 shows the independent set structure of a level-based hierarch.

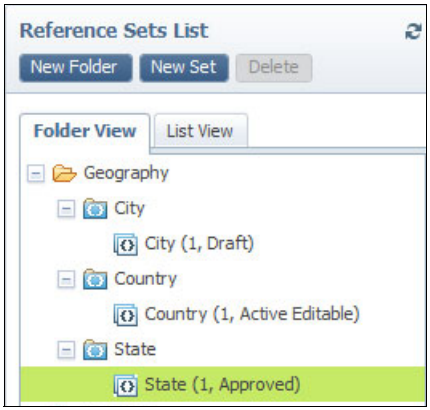


Figure 1-4 Independent sets structure of a level-based hierarchy

From a structure perspective, both tree and level-based hierarchies are the same. However, from a management and governance perspective, the two types of hierarchies have governance considerations that differ slightly. With a level-based hierarchy, changes can be made independently to the sets at each level as each set goes through its own change lifecycle. There might even be separate stewardship teams managing separate sets at each level. For level-based hierarchies, the stewardship process must take into account how changes to the underlying sets at each level are reflected in the hierarchies based on the values of the sets at each level.

A geographic hierarchy can be represented within InfoSphere MDM Ref DM Hub application as either a tree hierarchy or a level-based hierarchy. The distinction between them is how the relationships are defined between the terms and sets. A hybrid hierarchy structure can also be created where the hierarchy is defined using parent-child relationships within the values of a set and also between values of separate sets as in a level-based hierarchy. For example, a client might want to manage the nodes of the first three levels of a five-level hierarchy as a single set and use individual sets for the subsequent fourth and fifth levels.

1.2.3 Poly hierarchies

In some cases, a requirement might exist to create more complex relationships between reference sets and the values within those sets, to represent an ontology for example. Within IBM, the IBM Office of Chief Information Officer (CIO) uses the IBM Reference Data Management Hub to manage taxonomies of standard terms used within IBM internal and external facing applications. There is a business need to define relationships between the individual reference sets and values within those sets. The relationships are not always parent-child relationships, rather, they define a complex network, or poly-hierarchy structure. The InfoSphere MDM Ref DM Hub mapping capability supports creating these more complex cross-set relationships.

1.3 Challenges of managing reference data

Reference data is relatively static and managing the reference data in a single table over time might not seem to be a lot of work. This section outlines several reasons why the sum of managing reference data across all applications in the enterprise and coordinating changes and mappings across these applications is a major challenge.

Multiplicity of code tables and code table variations

Many of the same types of reference data are used across industries and applications and the reference data is commonly used and often defined in the limited context of a particular application. The result is that, within an enterprise, there are many different code set variations describing the same domain.

The differences between reference data sets across various systems can include semantic differences, coding scheme differences, format differences, and value differences. Even within well-established standards, there might be different representations for the same values. For example, the ISO 3166-1 standard defines three sets of country codes:

- ISO 3166-1 alpha-2: two-letter country codes
- ISO 3166-1 alpha-3: three-letter country codes, which allow a better visual association between the codes and country names
- ISO 3166-1 numeric: three-digit country codes, which are identical to those developed and maintained by the United Nations Statistics Division

In addition to ISO 3166-1, many other country code lists are defined and used by other international organizations. Some differ completely from the ISO standard

both in how the values are coded and the list of country code values contained in the set.

Another example of variability within a standard code set is demonstrated by NACE codes. The Statistical Classification of Economic Activities in the European Community, commonly referred to as NACE (acronym derived from its French name), is a European industry standard classification system consisting of a six-digit code. It is similar in function to the Standard Industry Classification (SIC) and North American Industry Classification System (NAICS) codes that are used in the US. The codes enable standard statistical reporting across industries and use of the standard is required in regulatory reporting. The first four digits of the NACE code, which is the first four levels of the classification system, are the same in all European countries. The fifth digit might vary from country to country and further digits are sometimes placed by suppliers of databases. Enterprises with operations in multiple countries often have to manage the different variations of the codes by creating reports within each country using the country specific standard version and then rolling up results to a single corporate version of NACE for its corporate level reporting. In addition, various versions of the NACE standard exist, and older data within a data warehouse might be classified according to earlier schemes:

- NACE Revision 1, the first revision of the original NACE (1970)
- NACE Revision 1.1, a minor revision of NACE Rev. 1
- NACE Revision 2, adopted end 2006

Similarly in the US, NAICS is a different but analogous industry classification scheme that superseded the earlier SIC standard in 1997. However, some types of business are still required to provide regulatory reports that use the older SIC scheme so both SIC and NAICS are actively used in the US. A global organization with operations in both US and Europe might need to translate between NACE, NAICS, and SIC codes to meet various regulatory reporting requirements.

In summary, even for something as commonly used as country codes or industry classification codes, where well established standards are available, the variability among the standards and versions of the standards means that an organization might need to handle and reconcile data across multiple versions and representations as a matter of course. Moreover, the mapping between the codes in separate code-set variations are often not one-to-one. NACE and NAICS represent completely separate code schemes. Although the end purpose is similar, the industry categories within NACE do not correspond one-to-one with the industry codes in NAICS. The categorization structure differs, the coding structure differs, and the number of values within the code sets also differs. One NACE code might map to multiple NAICS categorizations and vice versa, making it difficult to determine how a piece of transaction data categorized in NACE should be categorized against NAICS.

To accurately map data across dissimilar code sets on a one to one basis, it might be necessary to use rules and additional information beyond the source code value itself to determine how to map the value.

Just as the reference data sets are published as standards, so the mapping tables that map between similar sets are also often published as related standards by the standards organizations. For example, the EU RAMON Server lists a large number of correspondence tables (that is, mapping tables) between various versions of code lists and classifications that are used in European statistical reporting. See the tables at the following website:

http://ec.europa.eu/eurostat/ramon/relations/index.cfm?TargetUrl=LST_REL&StrLanguageCode=EN&IntCurrentPage=1

Although variations exist in and within industry standard code sets, the greatest source of code table variation within the enterprise is the enterprise applications where the code tables are defined and used. Reference data is typically created and managed in a siloed fashion within each individual IT application. In many cases, application developers implement code tables within an application with purely the local processing needs of that application in mind. Thus a hundred enterprise applications might have a hundred ways of representing a country code list, with various coding schemes and even various sets of country values within each.

Because each application typically has many code tables, the larger the enterprise, the greater the number of code table variations.

Maintaining mappings between reference data representations

If data and reference data were confined to individual applications, the problem of managing reference data within the enterprise would simply be the problem of how to manage the changes to reference data over time within each individual application. In reality, few applications work in isolation and data must cross application boundaries. Data is consolidated in master data hubs and data warehouses. Data passes from application to application in cross-business processes. Data that is entered in a web application by a client might result in transactions and processes in multiple back-end applications. Data with coded reference data values is received from suppliers, business partners, and customers as part of business transactions. Wherever transaction and master data flows between applications, so must the related reference data. Code tables that are used to categorize transaction data might have different formats and content within source and target applications for data movement. When transaction data flows from a source application to a target application, the code values of the source must be mapped to the corresponding code table values of the target if the categorization information that is associated with the reference data will be correctly carried also.

Typical use cases, where reference data mappings are key to the business, include data warehouse data load and master data hub data load.

Data warehouse load

Transaction and master data are consolidated and sent from back-end applications to the data warehouse. The data warehouse data is the source of business intelligence (BI) reports, financial reports, statutory reports, and so on. The data warehouse dimensions contain reference data and the reference data that is related to incoming fact data supports statistical analysis of the fact data. A common approach is to run a daily batch load job by using standard extract, transform, and load (ETL) tools, which use mapping tables to map the reference data codes from the source application format to the corresponding data warehouse format, so that the transaction data from back-end applications can be categorized correctly. The accuracy and completeness of the mapping tables might directly affect the accuracy of how fact data is mapped to dimensional data.

Without centralized management and coordination of reference data changes, the code table maps between the source application and data warehouse reference data representations are difficult to maintain in an ongoing basis and quickly become outdated. When reference data maps are incorrect or incomplete, the transaction data might not be posted correctly, or might not be posted at all. A common pattern for managing the case where a source reference code is not found in the map by an ETL job is to map the transaction data to a default “unknown” code for the particular dimension within the data warehouse. With this approach the quality of the categorization within the data warehouse declines over time, because the percentage of data that is correctly mapped to the respective dimensions declines.

Master data hub load and transactions

Master data encompasses the key business entities that participate in business transactions and operations including customers, products, and accounts. A master data entity, such as a customer, has reference data associated with it, and a typical master data management system has hundreds of code tables associated with the master data. A master data hub can be both a consolidation point and a distribution point for data relating to master data entities. Just as for the data warehouse, map tables allow reference data defined in external applications to be mapped to the code tables within the MDM hub so that master data that is sourced from those external applications can be loaded correctly into the MDM application. In addition, there might be a need to distribute data from the master data hub to other applications that require a reverse mapping.

The difficulty of managing reference data change

An enterprise has many variations of the same code set within the various enterprise applications and many code tables to manage. Although reference data is relatively static, it is not completely static, and managing changes to code tables across the enterprise can be a significant challenge. Many enterprise applications were never designed to accommodate changes to the code tables over time, and changing the tables will often require application development and testing.

Unlike master data management, reference data management is not an established discipline; many organizations have no central management of reference data.

Separate lines of business, application owners, and departments manage their own sets of reference data; there is often little communication or coordination between the various business areas. Rolling out a change across the enterprise is difficult. Also difficult is to understand or manage what individual changes are being made by the different reference data owners to their code tables.

Examples of changes to reference data are described next.

Changes to a standard

An example of changes to a standard is that a new country code is added to the ISO 3166-1 standard. When a standard is changed, an enterprise has to assess the following questions:

- What applications does this change affect?
- What will be the impact of adding the new code to each application and how will this be effected and coordinated?
- What mappings are affected and need to be updated?
- What is the contingency for managing data from applications that cannot be changed or that need to be changed on a staggered timeline?
- How will this change be reconciled with data recorded prior to the change?
- How will adoption of the changed standard by application owners be tracked and managed over time?

The impending healthcare industry change from ICD-9 to ICD-10 codes in the US is an extreme case of a code standard change. Healthcare organizations are currently engaged in major projects to ensure that their systems are ready for the change in October 2014.

However, even relatively small changes to reference data can cause problems because few organizations have a good understanding of how reference data is

being used in each application and what the inter-dependencies are. The impact of making a code change in one or more applications can be difficult to assess.

Changes to an application-specific code set

When an application-specific code set is changed, an enterprise has to assess the following questions:

- What other applications does this change affect?
- What external mappings are affected and need to be updated?
- How is this change coordinated across other application owners

Changes to mappings between sets

As an enterprise makes changes to reference data sets that are used within the application ecosystem, it must also update the cross-application reference data maps that are affected by a reference data change. These maps are used to drive mapping between one reference data representation and another and basically support interoperability between applications. Without governance over who is making changes to what reference data, coordinating changes to the related maps is difficult. As a result, the reference data maps that are used in ETL jobs and so on often become outdated.

Today, enterprises often catalog their reference data and the mappings between them by using Microsoft Excel spreadsheets and similar manual methods. Using a manual spreadsheet approach is error-prone and enforcing good data governance practices and policy around security, audit, and history is difficult.

1.4 The cost of unmanaged reference data

Today, many enterprises have no centralized enterprise governance over reference data; critical reference data is managed using spreadsheets and manual ad-hoc methods. The difficulty of managing change across the complex web of reference data variations is not systematically addressed; errors in reference data mappings and inconsistencies are accepted and tolerated as an everyday reality. Reference data variations and inconsistencies can be a major source of data quality issues within the enterprise and cause business losses through system downtime, incorrect transactions, and incorrect reports.

1.4.1 Costs related to business risk

The following business risks result from unmanaged reference data:

- Operational risk

  If incorrect reference data is used, transactions or applications might not function correctly and the outputs and results might not be as expected. Reference data often plays a key role in transactional applications and business processes so the potential business impact can be major.

- Compliance risk

  Regulatory compliance is increasingly a driver for adopting new approaches to managing reference data. More than ever, enterprises must be able to prove through internal and external audits about how their business processes work; that they work appropriately; that the data they incorporate is accurate; and that only authorized employees had access to the data. There has to be a clear audit trail and provenance for data used in regulatory reports.

  Frequently, enterprises have only the most rudimentary ways of handling governance over reference data and the common approach of tracking reference data in spreadsheets and doing manual reconciliation is both time-consuming and prone to human error. Because the data stored within data warehouse and financial systems is used to generate financial and regulatory reports the accuracy of this data is a critical requirement. Incorrectly mapped reference data can have a direct impact on the quality and integrity of data that is stored within the data warehouse

- Analytical risk

  In addition to regulatory reporting, the data warehouse is used to drive business intelligence analysis and reporting. Reference data is fundamental to reporting because it defines the dimensions underpinning BI analysis. As BI reports are increasingly used to drive business decisions, business decisions might be made based on faulty analysis if transaction data that is mapping to dimensions is incomplete or incorrect.

- Distribution risk

  Reference data changes might not be correlated across related applications, causing data transfer problems, and reporting inconsistencies between applications.

1.4.2 Increased IT cost

Usually IT has the task of managing the reference data mappings, and there is little involvement from the business. Because reference data might be independently changed within source applications, data transfer mappings that are used to map reference data from one application to another become inaccurate and incomplete over time, potentially resulting in failed transactions, missing data, or incorrectly categorized transactions. IT often remains in a reactive mode, only recognizing and reacting to changes in back-end application reference data by the failures caused mapping and transaction problems.

Separate application owners make their own changes without addressing the implications of those changes on business processes or other parts of the organization.

1.4.3 Cost of business inflexibility

Reference data is a key aspect of any application integration, and the speed with which an enterprise can integrate the reference data from new applications and correctly map that new reference data to existing reference data standards has a bearing on business flexibility. Whether new applications are introduced through ongoing application acquisition or organizational mergers, the ability to accommodate new reference data can directly affect the speed with which the IT infrastructure of an organization can change. Without a formal governance program and the support of a reference data management application integration, reconciliation and ongoing management of new data and reference data sources is slowed.

1.5 Reference data governance with master data management approach

Master data management has become the best practice approach to address the problems that are caused by master data being defined and managed in many separate application and data silos within an enterprise. The goal of master data management applications is to support identification and reconciliation of master data records across an organization, and to provide well defined processes for managing the stewardship of the master data over time.

Master data management systems support identification and maintenance of a gold record for master data entities within the enterprise. Reference data by definition is not master data but it presents a similar problem: Reference data is defined and used independently across many applications and tends to be

maintained in a siloed fashion on an application-by-application basis. Just as master data management systems can support formal, centralized governance and management of master data at an enterprise level, the same approach can be used to manage reference data. The emerging best practice is to treat reference data like a master domain in its own right and to provide centralized management and stewardship of reference data using a standard set of processes, policies and tools.

Master data management software provides the tooling foundation for building a specialized reference data management governance solution. IBM has designed a dedicated stewardship and hub application for the centralized management of enterprise reference data using a master data management approach. The IBM InfoSphere Master Data Management Reference Data Management Hub, first launched as a component of IBM InfoSphere Master Data Management Platform in July 2012, provides the repository, services, and stewardship user interface to support a complete governance program for enterprise reference data.

Managing reference data is a key aspect of an enterprise data governance program. The industry best practice for enterprise data governance in general is to establish a Project Management Office (PMO) with oversight of data governance projects. If a data governance PMO already exists, then reference data management project might fall under the auspices of that PMO. If not, then a reference data management project can be a good starting point for implementing a broader master data management initiative and data governance structure. IBM Lab Services and IBM Global Business Services® have plans and details for how to establish a data governance program so only certain key elements are covered here.

A data governance initiative for managing reference data should address four key elements:

- *People*: Who owns the reference data and who is responsible for the data, both on the business side and within IT? The answer to this question will establish liaison relationships and simplify communication regarding questions and updates. Separate lines of business might have separate reference data requirements and typically, there will be one or more stewardship teams managing separate reference data sets and mappings. As with master data management, a centralized stewardship approach simplifies implementation at an enterprise level.

  The requirements for how the data stewardship team is structured and how individual stewards can change what reference data will vary from organization to organization. One approach is to organize reference data stewards by business area: assigning a steward responsibility for reference standards for a particular business area. Business users should be able to raise change requests for additions or changes to reference data and the

change management process should include routing the request to the appropriate steward who is then responsible for evaluating the change request and for ensuring the collaborative review and approval process depending on the type of reference data, and the type of change requested.

- *Process*: What are the processes for stewardship and governance of the data? What are the processes for change management over the data? What are the processes for using the data in business scenarios?
  - Publishing of standards. At its heart, an RDM strategy must create the foundation for reference data standards across the enterprise, taking corporate, industry, and global standards into account. The process should support publishing reference data standards, and active and passive distribution of changes in reference data to the community of subscribing applications.
  - Change management: One of the key precepts underlying an RDM strategy is the ability to accommodate change. While some reference data remains static and changes relatively infrequently, other data might fluctuate on a daily or more frequent basis, for examples, currency exchange rates. For that reason, the enterprise must establish a procedure for accommodating change and updates. What are the change request processes? Who will manage the publication and dissemination of changes to systems that subscribe to the data? How will bidirectional changes between systems and applications be monitored and confirmed? Who among business and IT owners should be notified or consulted about changes? A typical process allows business users to direct the change requests that relate to the reference standards to the steward team and support a collaborative evaluation, review, and approval process for agreed changes.
  - Deployment and test strategies: How are changes to reference data coordinated across multiple applications? What is the deployment and test cycle for applications affected by reference data change? Publishing of changes to standards should include notification to business and application owners and can be scheduled in such a way as to allow those applications time to be changed to adopt the new standard (for example, monthly publish of standards changes).
  - Discover: Although reference data might be ubiquitous, it is not always obvious. An RDM strategy should include a way to discover reference data wherever it appears within an enterprise's applications and spreadsheets. The result of the discovery process should be a catalog that profiles all existing applications, and there should be a process for bringing discovered reference data under centralized governance.

- *Policies*: Organizations should establish policies around security, data ownership, audit, history, data retention, and other non-functional requirements for the reference data under management.
- *Tools*: Technology provides the automation to help manage and implement the governance program. The IBM Reference Data Management hub is designed specifically to support governance, and stewardship of reference data including enforcement of data management policies. The InfoSphere MDM Ref DM Hub supports defining, managing, and publishing the “gold” or canonical reference data standards for the enterprise while managing synchronization and mapping of application specific reference data representations to the canonical representation and to each other. The InfoSphere MDM Ref DM Hub supports import and onboarding of reference data into the hub, and publishing of reference data and related changes to consuming and subscribing applications. The InfoSphere MDM Ref DM Hub works in conjunction with ETL tools, MDM applications, and many other data quality and data management tools. IBM RDM also integrates with IBM InfoSphere Business Glossary IBM business data dictionary. IBM InfoSphere Information Analyzer supports discovery and profiling of reference data in back-end applications so that the reference data can then be easily brought under management in the InfoSphere MDM Ref DM Hub.

1.6 InfoSphere MDM Ref DM Hub feature overview

The IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) was released as a separately chargeable component under the IBM Master Data Management Product ID (PID) in July 2012. The hub was developed as a stand-alone reference data domain on the InfoSphere MDM Custom Domain Hub Platform which itself is the foundation for the InfoSphere MDM Advanced Edition. The InfoSphere MDM Ref DM Hub implements its own specialized domain model specifically for reference data, that is, reference data is supported as a first-class domain entity. The InfoSphere MDM Ref DM Hub includes a dedicated stewardship interface that is designed for managing reference data. The web-based user interface (UI) runs in the browser and no special code is required on the client. The UI is designed for business users, with intuitive and familiar navigation and controls. A flexible data model supports dynamic modelling of reference data properties through the UI ensuring a quick implementation and minimizing the need for IT involvement on an ongoing basis.

InfoSphere MDM Ref DM Hub is deployed as a stand-alone hub to provide a single point of management and governance for enterprise reference data.

InfoSphere MDM Ref DM Hub provides a robust solution for centralized management, stewardship, and distribution of enterprise reference data. It supports defining and managing reference data as an enterprise standard. It also supports maintaining mappings between the various application-specific representations of reference data that are used within the enterprise. The InfoSphere MDM Ref DM Hub supports formal governance of reference data, putting management of the reference data in the hands of the business users, reducing the burden on IT, and improving the overall quality of data used across the organization.

1.6.1 Key functions of the InfoSphere MDM Ref DM Hub

InfoSphere MDM Ref DM Hub is designed as a ready-to-run application. It can be quick to install, easy to use and understand, and delivers real value for immediate use without requiring extensive customization. The key functions include the following items:

- Role-based user interface with security and access control including integration with LDAP
- Management of reference data sets and values
- Management of mappings and relationships between reference data sets
- Importing and exporting of reference data in CSV and XML format through both batch and user interface
- Versioning support for reference data sets and mappings
- Change process controlled through configurable lifecycle management
- Hierarchy management

InfoSphere MDM Ref DM Hub is built on the proven InfoSphere MDM platform and delivers a master data management approach to managing enterprise reference data. It helps to reduce business risk, improve enterprise data quality, and enhance operational efficiency. InfoSphere MDM Ref DM Hub is based on a three-tiered component architecture, comprising a client and a server application interacting with a back-end database that hosts the application-specific data and required metadata.

Figure 1-5 depicts a high-level component architecture of InfoSphere MDM Ref DM Hub.

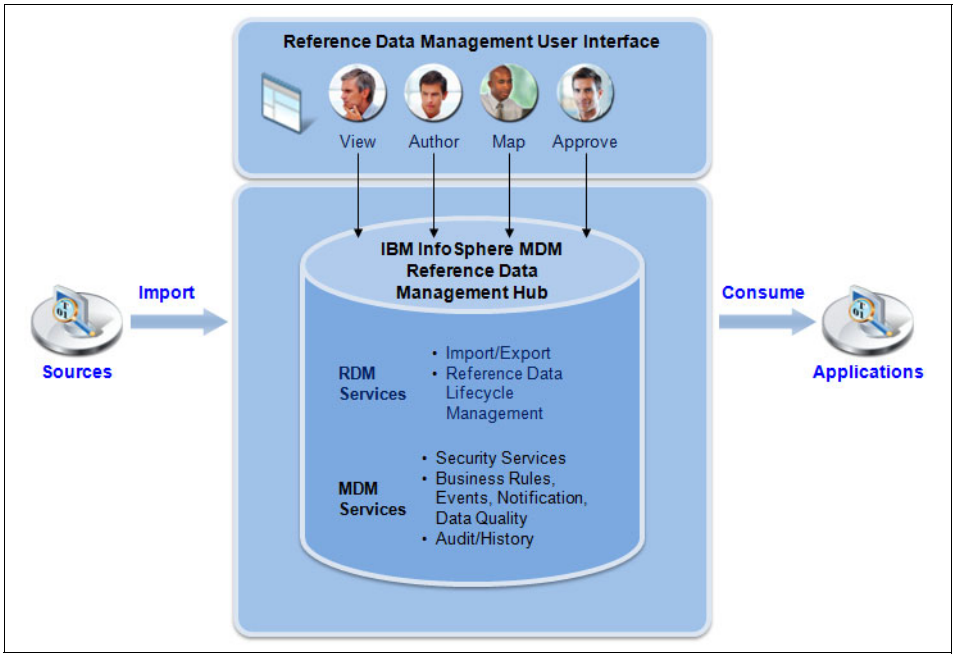


Figure 1-5 InfoSphere MDM Ref DM Hub logical architecture

The InfoSphere MDM Ref DM Hub user interface is a web application UI that supports collaborative authoring of reference data. Reference Data Stewards use the RDM web UI for the importing, managing, and publishing of reference data sets. The role-based UI allows a stewardship team to view, author, map, and approve reference data sets within a central repository. With this approach reference data sets can be created and managed in a controlled manner. User actions on the web UI trigger requests, which are handled by appropriate service controllers present in the REST layer. The REST layer services invoke the server-side transactions to manage CRUD procedures on RDM database.

The server-side is implemented on the proven InfoSphere Custom Domain Hub engine (the same engine that powers InfoSphere MDM Server and InfoSphere MDM Advanced Edition).

The reference data domain model elevates reference data to be a first class domain entity within MDM. By implementing the InfoSphere MDM Ref DM Hub as a new domain on the InfoSphere MDM platform, the InfoSphere MDM Ref DM Hub benefits from a wide range of base services and ready-to -use frameworks that InfoSphere MDM provides such as business rules, event notification, data

quality, and audit history. In addition, several reference data management specific services are implemented to achieve key functionality such as import and export, reference data set lifecycle management, transcoding, distribution, and versioning.

The client and server enterprise archives reside in a WebSphere Application Server instance. The currently supported databases are IBM DB2 and Oracle.

1.6.2 Understanding reference data sets

Reference data sets are at the heart of a RDM system and are used to manage and contain reference data. Every data set is associated with a reference data type that defines the properties of the data set.

When a reference data set is created, it is automatically given a version number, starting with 1. You can change the version number to another value, numeric, or alphabetic. When you create new versions, you can give them any label to indicate the version number. Each reference data set has at least one version. All versions of a data set are grouped together. For ease of management, reference data sets can also be grouped in folders.

The InfoSphere MDM Reference Data Management Hub user interface uses a drag and drop hierarchy widget to visualize and manipulate the reference data set folder hierarchy, and the reference data sets within them. Folders, the reference data sets that the folders contain, or individual reference data sets can be dragged from one folder to another.

Organize sets using folders

Folders are an aid for organizing and navigating reference data sets. A reference data steward can give the folders meaningful names that suit the context of the work environment, such as named for the project they belong to, the person who created them, or a date. Folders can also contain child folders to further help organize reference data sets.

The folders are listed in alphabetical order. Scrolling down the Folder View, any reference data sets that are outside of a folder are listed after the folders.

Figure 1-6 shows the reference sets, organized into folders in the InfoSphere RDM user interface.

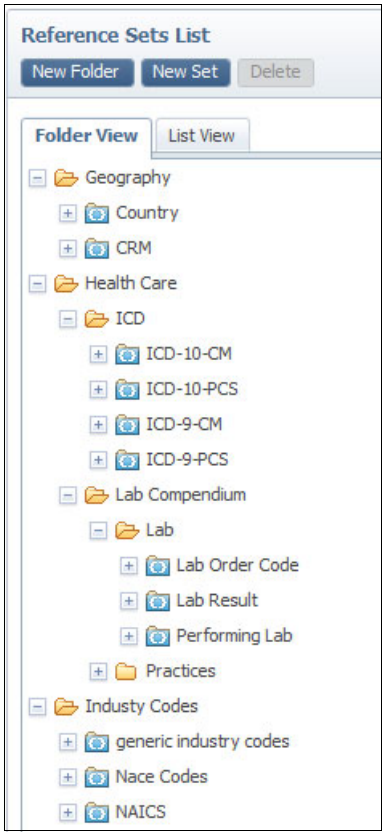


Figure 1-6 Reference data set folder view

Reference data set versions

Reference data sets support versions. Versions are used to provide a structured approval process for changes to reference data before those changes become active.

Data stewards can also create a copy of a reference data set as a new version of the original, preserving a set before changes are made. In the event the changes are unwanted, the later version can be deleted to restore the previous version.

Lifecycles and states

When creating a reference data set, a steward assigns a lifecycle process with that set. Every reference data set and mapping has a lifecycle process that defines the states and state transitions governing changes to the set.

Each lifecycle has a set of states that correspond to the steps in the change, review, and approval process. Four lifecycle processes, listed next, are available and ready for immediate use with the InfoSphere MDM Ref DM Hub application. Various lifecycles can be defined at implementation time to suit an organization's specific governance processes.

Simple Approval Process

Simple approval process defines a lifecycle where changes might be made to a draft version of a reference data set, and a single approval is required to publish the change. The set or mapping starts in Draft state allowing an authorized data steward to make changes. When the steward completes making changes and is ready to publish the change, the steward submits the change for approval that changes the state to Pending Approval. A user with the Approver role can then review the changes and approve or reject the changes that in turn changes the state of the set or mapping to Approved or Rejected. The set or mapping can be in the Retired state when it is no longer used, and from the Retired state to the Dropped state, disbarring any further edits.

Simple approval process is the most commonly used process for the lifecycle of a reference data set. The simple approval process includes the following states:

- *Draft*: This is the initial state for a reference data set or mapping. Data stewards request approval by changing the state to Pending Approval.
- *Pending Approval*: The reference data set or mapping is awaiting approval from a user with the Approver role.
- *Approved*: The reference data set or mapping is approved for use. Whether the data set or mapping is available to consumers also depends on its effective date and expiration date. Sets and mappings in the Approved state cannot be edited.
- *Rejected*: The reference data set or mapping is not approved for use. It can be edited in the rejected state by the data steward, or it can be changed to the draft state. To submit the rejected reference data set or mapping for approval, the state must be changed first to Draft, and then to Pending Approval.
- *Retired*: The reference data set or mapping is no longer used, although it can be edited, and new data set versions can be created from a retired data set.
- *Dropped*: The reference data set or mapping can no longer be edited, and its state cannot be changed. New data set versions can be created from a dropped data set.

State Machine - 2

The mapping or set using this lifecycle process has only two states, *Draft* and *Approved:*

- Draft - 2: This is the initial state for a reference data set or mapping. Data stewards request approval by changing the state to Approved - 2.
- Approved - 2: The reference data set or mapping is approved for use. Whether the data set or mapping is available to consumers also depends on its effective date and expiration date. Sets and mappings in the Approved - 2 state cannot be edited.

State Machine - 2 process might be appropriate where a single person or steward is responsible for editing the mapping or set. In that case, there is no need for a separate approver. This state machine allows changes to be made to a draft copy before being published.

Active Editable

The set or mapping using this lifecycle process is immediately available for use, and can be edited at any time. Active Editable might be appropriate for certain types of reference data where the changes are made automatically by a systemic process and do not require stewardship intervention or a review and approval. Active Editable process supports changes directly to the active sets. Because there is no approval process, the State property is disabled.

Two Step Approval

The mapping that uses this lifecycle process requires two approvers before it can become available for use. The two-step approval was designed for customers with more rigorous governance controls who require a multistep approval process. This lifecycle process includes the following states:

- *Draft*: This is the initial state for a reference data set or mapping. Data stewards request approval by changing the state to Pending First Approval.
- *Pending First Approval*: The reference data set or mapping is awaiting approval from a user with the Approver role. The Approver can change the state to Pending Second Approval or Rejected.
- *Pending Second Approval*: The reference data set or mapping is awaiting approval from a user with the Approver2 role. The Approver2 can change the state to Approved or Rejected.
- *Approved*: The reference data set or mapping is approved for use. Whether the data set or mapping is available to consumers also depends on its effective date and expiration date. Sets and mappings in the Approved state cannot be edited.

- *Rejected*: The reference data set or mapping is not approved for use. It can be edited in the rejected state by the data steward, or it can be changed to the draft state. To submit the rejected reference data set or mapping for approval, the state must be changed first to Draft, and then to Pending First Approval.
- *Retired*: The reference data set or mapping is no longer used, although it can be edited, and new data set versions can be created from a retired data set.
- *Dropped*: The reference data set or mapping can no longer be edited, and its state cannot be changed. New data set versions can be created from a dropped data set.

Reference values

Reference data sets contain rows of data values, which might be only a few rows in small data sets to many rows in large data sets, although data sets typically have fewer than one hundred thousand (100,000) rows. The data can be managed by using the InfoSphere MDM Reference Data Management Hub console, data imports, or through updates applied using the batch processor or web services.

The UI supports both simple and advanced filtering to limit what values are displayed for a set.

Each row in the data set can contain multiple properties, as defined by the data set type, and support a number of language translations. In addition, data values can participate in mapping and hierarchy associations.

Data deletion is always handled with a soft deletion mechanism. Data is never physically deleted from the InfoSphere MDM Reference Data Management Hub database on a delete action.

Reference data set translation

Reference data sets can have multiple language translations for each reference value in the set. The translations of a particular reference value can be provided manually on the Translations tab in the Set Values view. Alternatively, the translations can be imported from a comma-separated values (CSV) file.

Reference data set hierarchies

The InfoSphere MDM Reference Data Management Hub console supports the creation of two types of hierarchies over reference data:

- Tree-style hierarchy
- Level-based hierarchy

Tree-style hierarchies

Reference data set hierarchies within InfoSphere MDM Ref DM Hub are tree-style hierarchy structures, created over the values within a reference data set. You access the hierarchies by using the Set Hierarchies view.

The following actions are available from the Set Hierarchies view:

- Save or discard changes to a hierarchy.
- Select the hierarchy to work on by selecting Set Hierarchies from the View menu.
- Create a new empty or pre-populated hierarchy over the values in a reference data set.
- Import a hierarchy definition from a comma-separated values (CSV) file.

  A hierarchy structure can be defined between the elements of a set in a CSV file and imported over a set of reference data values. When dealing with hierarchies over a large set of values, an easier approach is to define the hierarchy externally, and create a CSV file to import the hierarchy into InfoSphere MDM Reference Data Management Hub.
- Copy a hierarchy.
- Delete a hierarchy
- Refresh the list of hierarchies

Note: Each reference value can exist only once within a hierarchy.

Level-based hierarchies

InfoSphere MDM Ref DM Hub supports level-based hierarchies across multiple sets, where each level of the hierarchy is associated with a different reference data set.

An example of such a level-based hierarchy is city/state/country where city, state, and country are each reference data sets. Managing the relationships between the values across the sets is both valuable and meaningful.

Reference data sets can be linked together by using the reference data set property type. The values from one reference data set can be surfaced as a lookup table within the value properties of a related reference data set. For example, a State reference data set has a property for Country, and this property is of type reference data set. State and country become linked by using the Country reference data set as a lookup table within State values. This linkage creates a value-based hierarchy.

The first step in supporting a level-based hierarchy is to create the appropriate reference data set types using the administration tab in the InfoSphere MDM Ref DM Hub UI.

Reference data set subscriptions

Subscriptions form part of the definition of the data distribution model within the InfoSphere MDM Ref DM Hub. The basic concept of subscriptions is that it allows properties to be set that will drive how application subscribers subscribe to the data within a particular reference data set.

A *subscription* is an association made between a reference data set and a managed system; this association indicates who the publishers and consumers of the reference data set are. Therefore, it is possible for a reference data set to have many subscriptions, because you have a single publisher and many consumers. Subscriptions are useful when you do import or export operations for the associated reference data set.

Figure 1-7 shows subscription properties for a reference data set.

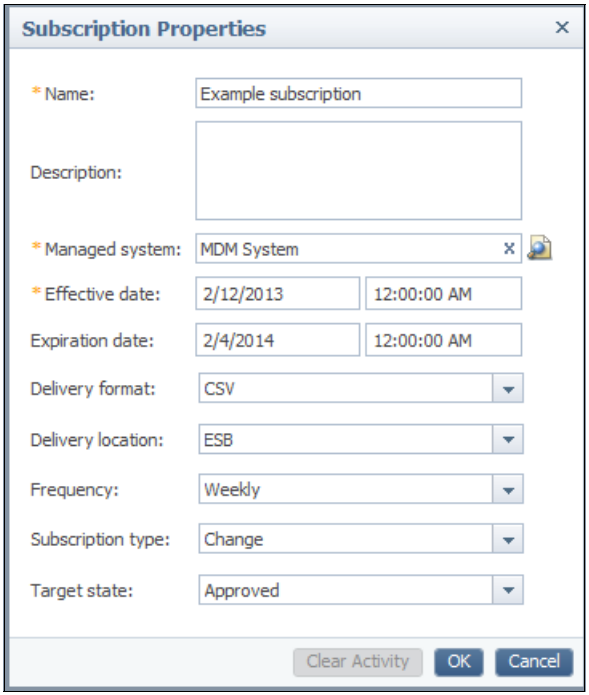


Figure 1-7 Subscription properties for a reference data set

Reference data set history

History data that reflects all the changes to a reference data set or map is maintained within the InfoSphere MDM Ref DM Hub in special history tables. This history data can be retrieved through use of customized database SQL calls.

Data that is deleted through the user interface is not physically deleted from the InfoSphere MDM Ref DM Hub database, but is marked as logically deleted. (This process is often called a soft delete.) Deleted data is not returned through the InfoSphere MDM Ref DM Hub inquiry services or user interface, but can be retrieved, for audit purposes, from the InfoSphere MDM Ref DM Hub database.

1.6.3 Understanding reference data types

Every reference data set (data table) or mapping that is created within InfoSphere MDM Ref DM Hub has a default set of properties. These reference data set and mapping properties are defined by using reference data types. Every data set and mapping is associated with a data type definition.

The InfoSphere MDM Ref DM Hub system includes two basic data types, Default Reference Data Type and Default Mapping Type, which can be used to define a simple reference data set or mapping. An authorized user can create reference data types with additional properties to define more complex reference data sets or mappings.

Properties that are defined in a reference data type can be applied at the reference data set level, or at the value level for sets and mappings. At the value level, properties apply to all the rows of data in a data table or to all value mappings.

Reference data types have a data type property, which designates whether the reference data type is used to create reference data sets or mappings. After the reference data type is saved, its data type cannot be changed.

Business keys

Reference data types can be defined with compound keys. Up to four properties plus the code can be defined as constituent key parts of a unique reference data value key. Business keys apply only to reference data set data types; mapping set data types do not use business keys.

The business key feature is enabled for a reference data type in the Administration menu in the InfoSphere MDM Ref DM Hub UI. Selecting the business key check box for a reference data type is one way to ensure that the code for a reference data set is unique.

Note: Name is also a required field for each reference value but is not used in determining uniqueness of the compound key.

InfoSphere MDM Reference Data Management Hub supports up to five properties within a compound key for a set, where Key 1 is always the set code. The other four properties must have the Key property enabled within the reference data type definition.

Compound keys allow records with the same code to be saved, if the combined key is unique. The uniqueness check is done against the full key that is defined for the type. If the overall compound key is unique, the record can be stored.

Importing values into a reference data set, defined as having a compound key, is the same as importing values into any other set. The same rules hold for the manual case: the key values cannot be null and the overall combination of keys and code must be unique for each value.

The business key check box and compound key definition can be applied for a reference data set with existing values if the values conform to the rules defined for the keys. The InfoSphere MDM Ref DM Hub checks for data integrity when the compound key settings for a reference data set type are changed, and does not allow actions that would create duplicate entries in the database where uniqueness is required.

Uniqueness is not enforced for the Code property unless the business key option is selected. Uniqueness is not enforced to support effective date-centric use cases that require multiple entries for the same code with different effective dates.

Preferred: For most reference data types, the business key is preferred.

Reference data set data types

When you create a data type property of type Reference Data Set that points to a data set in the system, the options available must be from a current version of that set. A current version of the set is required to fill that property in a reference value.

The current version of a set is one that meets all of these conditions:

- The state is in an approved state for the lifecycle process or state machine.
- The effective date is the current date or earlier.
- The expiration date is after the current date.

If more than one version meets these conditions, the most recent version is considered the current version.

Dates: The effective date and expiration date for individual reference values are not relevant to the definition of a current version.

Example

You create a reference data set named Countries, which contains reference values for country names. The status for the Countries data set is Approved. Its Effective date was last month, and its Expiration date is next year.

Next, you create a reference data type named Branches, which includes a Value Level property named Country. The data type for the Country property is Reference Data Set, and its Related set is the Countries reference set.

When you create a reference data set named Branches, basing it on the Branches reference data type, the Country property is populated with the values from the Countries data set. Note that the Country property is populated with the values from the Countries data set only if the Countries data set has a current version.

1.6.4 Mapping reference data sets

Mappings can be defined between two reference data sets, and the reference values within those sets, to relate associated data.

For example, NACE is the European standard for industry codes, and NAICS is the North American standard. You can map a reference data set that contains NACE codes to one that contains NAICS codes.

The mappings browser is where you perform the primary management tasks for mapping sets. With the mappings browser, you create, read, update, and delete a mapping between two reference data sets and map reference values in those data sets.

Each mapping has its own metadata, including version and lifecycle state. For instance, the version for a mapping is unrelated to the version of the sets that are being mapped. You can easily locate a map by filtering for the source or target set associated with the map.

Every reference data set and mapping has a state that corresponds to its lifecycle process. The lifecycle is used by the data administrators to control the versions of reference data sets and mappings that are in use.

1.6.5 Managed systems

Managed systems are references to external systems that represent the suppliers and consumers of the reference data defined within the InfoSphere MDM Ref DM Hub.

Managed systems support configuration of the integration with applications and other entities that provide data to InfoSphere MDM Ref DM Hub or that subscribe to data exports from the hub.

You can set properties for the managed system to enable InfoSphere MDM Ref DM Hub and integrated applications to communicate with the managed system. Managed systems are associated with reference data by subscribing to a specific reference data set.

Each managed system can have custom properties that assist your external integration services personnel in providing reference data to external systems. Integrating managed systems requires custom code as part of the implementation.

1.6.6 Batch export

You can install and run the batch export function from any supported system that has access to the InfoSphere MDM Ref DM Hub. The hub can be on the same physical system that is used to perform the export, or it can be on a remote system.

Batch export fully supports exporting the following entities:

- Data values
- Data types
- Translation
- Mapping
- Hierarchies

Batch export can produce CSV or XML formatted output. The formatted output is compatible with the CSV and XML import and export functions accessible with the user interface.

The resource definition and control for batch export is handled with a properties file and command line options. You use the properties file and command line options to specify the type and name of the resource that is exported. A log file is created that contains the result of the export and statistics on the number of rows exported.

Batch export properties

A sample properties file, named `RDMBatchExportClient.properties`, is provided with the InfoSphere MDM Ref DM Hub installation. The options in this file are used to specify parameters for running the batch export.

The options in the properties file have the following general format:

```
<option> = value
```

Table 1-1 lists the options that apply to batch export.

Table 1-1 Batch export options

Option	Description
provider_url	Provider URL of the server instance in the `host:port` format. Example: `myserver.ibm.com:9080`
target_dir	File output directory. Example: `C:\output\`
format	File format for the output. Supported formats are CSV and XML. Example: format = csv
export_separator	CSV delimiter, such as “,” or “|” delimiter. Example: `export_separator = ,`
export_wrapper	CSV wrapper is double quotation marks. Example: `export_wrapper = "`
timestamp_format	A string that defines the format for time stamps used in CSV exports. When exporting XML, batch export uses the time stamp format mandated by the XML specification. Example: `timestamp_format = "yyyy-MM-dd'T'HH:mm:ssz"`
timezone_offset	Used in CSV exports to designate the offset that the values have from UTC. Example: `timestamp_offset = +00:00`
date_format	A string that defines the format for dates. Example: `date_format = yyyy-MM-dd`
max_translations	The maximum number of translations to output in CSV files. A value of 0 instructs the software not to export any translations. For XML formatted output, all translations are exported unless this value is 0. Example: `max_translations = 10`
hierarchy_export	For hierarchy exports, “tree” or “list” options are available. Example: `hierarchy_export = list`

Option	Description
user	User credentials for web service authentication. Example: `user = tabs`
password	User credentials for web service authentication. Example: `password = tabs`
requestid	Request control parameters to identify the request. Example: `requestid = 100101`
requester_name	Request control parameters to identify the name of the user that makes the request. Example: `requester_name = Some Name`
request_lang	Request control parameter to identify the request language. Example: `request_lang = 100`
page_size	Page size used while retrieving data from the server. Identifies the number of rows to be returned in a single request. Example: `page_size = 250`
debug	When set to false, no debug information is produced. When set to true, debug information is written to the file debug.log in the output directory for the requested file type. If the debug.log file cannot be created, an error message is written to the command window used to start the batch export. Example: `debug = true`

2

Solution reference architecture

This chapter describes how and where IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) fits into an information management reference architecture. The chapter describes the core and supporting reference data objects and services that are provided by the InfoSphere MDM Ref DM Hub. It also examines the integration patterns with common enterprise information management components: InfoSphere Master Data Management, SAP, content and taxonomy management systems, and data warehouses.

2.1 Base reference architecture

Overall, the InfoSphere MDM Ref DM Hub serves as an integration, management, and distribution point in the enterprise for reference data sets, maps between reference data sets, and hierarchies over reference data.

Figure 2-1 shows an overall view of where RDM fits into an enterprise reference architecture.

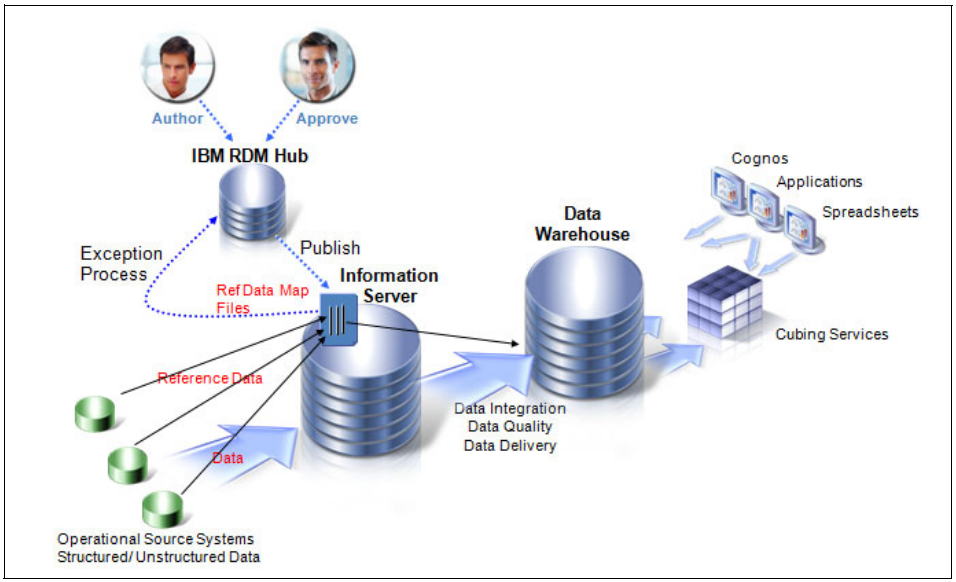


Figure 2-1 Reference Data Management Hub in an enterprise architecture

Reference data sets and hierarchies that InfoSphere MDM Ref DM Hub provides are consumed by enterprise information systems (such as InfoSphere MDM, SAP, data warehouses, business intelligence systems and so on) to ensure that business objects are accurately and consistently described across the enterprise. Reference data maps (described in “Maps” on page 39) are used by data integration layers (such as IBM InfoSphere Information Server, or an enterprise service bus) to map reference data values between source systems and target systems.

Stewardship of reference data consists of the following tasks:

- Importing or authoring reference data from source systems
- Managing the change to the reference data in an orderly fashion
- Distributing the reference data to downstream systems.

2.1.1 InfoSphere MDM Ref DM Hub

Master data is the common set of business objects (such as customers, products, location, account, and so on) that are shared across an enterprise. The InfoSphere MDM Ref DM Hub introduces a new domain of master data (in this case, reference data) that is hosted within the InfoSphere Custom Domain Hub. InfoSphere Custom Domain Hub serves as a framework that makes the InfoSphere MDM Ref DM Hub domain objects available as services, and provides ancillary services to InfoSphere MDM Ref DM Hub. The ancillary services include security, event notification, enterprise specific extensions to InfoSphere MDM Ref DM Hub services, and so on. For more information about master data and master data management, see *Enterprise Master Data Management: An SOA Approach to Managing Core Information,* by Allen Dreibelbis, Eberhard Hechler, Ivan Milman, Martin Oberhofer, Paul Van Run, Dan Wolfson.

Reference data (sets, maps, hierarchies) is either imported into the InfoSphere MDM Ref DM Hub or entered directly through the InfoSphere MDM Ref DM Hub user interface.

2.1.2 InfoSphere MDM Ref DM Hub services and user interface

All of the InfoSphere MDM Ref DM Hub objects can be accessed through the web services. Additionally, a REST layer on top of the InfoSphere MDM Ref DM Hub web services is used by the InfoSphere MDM Ref DM Hub user interface. The InfoSphere MDM Ref DM Hub UI is the standard way to manage the InfoSphere MDM Ref DM Hub and to govern reference data.

2.1.3 Core reference data domain objects

There are three core reference data domain objects: *sets*, *maps*, and *hierarchies* (Figure 2-2 on page 38). Each object supports the standard create, read, update, and delete (CRUD) operations. Each object also supports the notion of a validity period (the time when an object becomes active, and the time when an object is no longer valid). Sets and maps also support extensibility and lifecycle. These capabilities are described in 2.1.5, “Supporting services” on page 43.

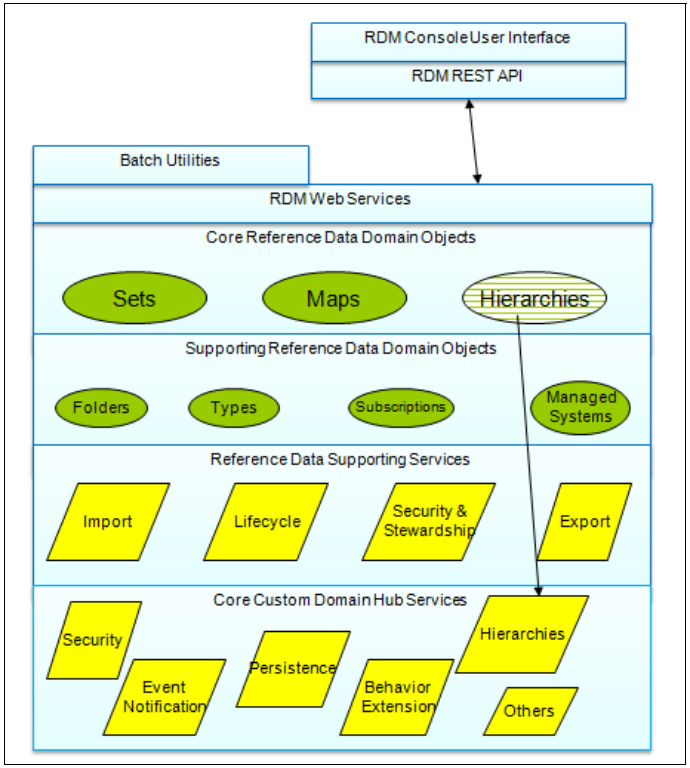


Figure 2-2 Reference data domain objects

Sets

Sets are the core object of a reference data domain. Sets refer to the list of values of a data element that is used to classify data. In other systems, reference data sets are known as *code tables*, *lookup tables*, *property lists*, and v*alue lists*. Examples of sets that are commonly seen include the following items, for example:

- Country codes
- Employee types (for example, full time, part time, temporary, contractor)
- Courtesy title (for example, Mr., Mrs., Dr., Miss, Rev.)

Values within sets have two required attributes:

- Code: This attribute is the underlying system representation of an individual value. Codes are the key for a value (and can be extended to support multiple codes).
- Name: This attribute is the actual meaning of the particular code.

Figure 2-3 shows a simple set.

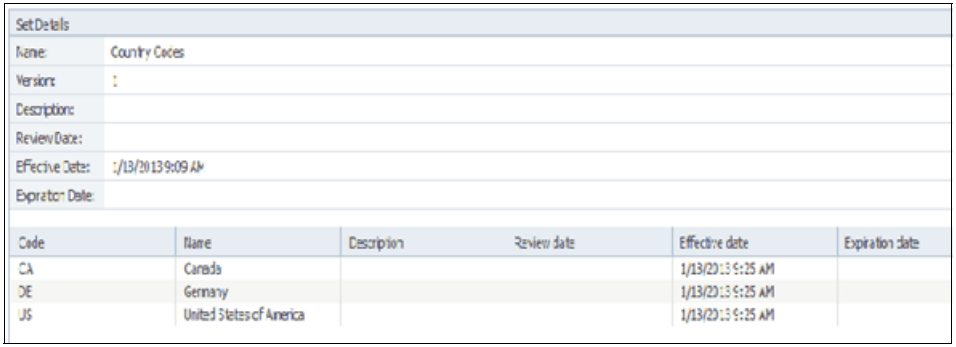


Figure 2-3 A set representing ISO-3166 country codes

Set values also have an additional attribute, *translations*. Names of individual values can be translated into separate strings for separate countries, for example, *doctor* in English and *medico* in Spanish. Sets can also be extended with additional properties, both at the set level, and at the value level.

Maps

Maps are a special type of relationship between sets; in particular, maps are used to link values in one set to values in another set. For example, Table 2-1 and Table 2-2 on page 40 both represent the concept of countries, but each set represents that country differently.

Table 2-1 shows the values for countries as telephone dialing prefix, where the value for each country is represented as a two-digit number.

Table 2-1 A set of telephone country codes

Code	Value
1	United States of America
44	United Kingdom
49	Germany
...	...

Table 2-2 shows a set based on the International Standards Organization (ISO) 3166 standard for country codes, and represents the value of each country as a two-character values.

Table 2-2 A set of country codes represented by two characters per ISO-3166

Code	Value
DE	Germany
UK	United Kingdom
US	United States of America
...	...

Table 2-3 is a map that shows which values in two sets are equivalent. In this example, the value `1` in the Telephone Dialing Prefix Set is equivalent to `US` in the ISO 3166 Country Code Set. The link between one value in a set and the equivalent value in another set is called a *value map*.

Table 2-3 A Map between telephone dialing codes for countries and ISO 3166

Telephone Dialing Prefix Set	ISO 3166 Country Codes Set
1	US
44	UK
49	DE

Reference data maps are used by data integration programs (where data is transferred between separate systems that represent the same information differently) and data distribution programs. The data integration layer takes a reference value in a source set, and uses it to map it to the corresponding value in a target set. Thus in our earlier example, the value of 1 is mapped to US.

Hierarchies

Hierarchies are also a relationship between reference data values, but are different from a map. A map is a list of relationships between values in *different* sets; hierarchies are a relationship of values *within* a set and are known as *set hierarchies*.

Table 2-4 on page 41 shows a set of travel expense accounting codes, with each code having a numeric value and a name representing the detailed type of expense.

Table 2-4 Travel expense accounting code set

Code	Name
100	Travel
121	Airfare
124	Hotel
200	Non-Travel
225	Phone
233	Internet Service
2251	Mobile Phone

Figure 2-4 illustrates a simple hierarchy of values in this set.

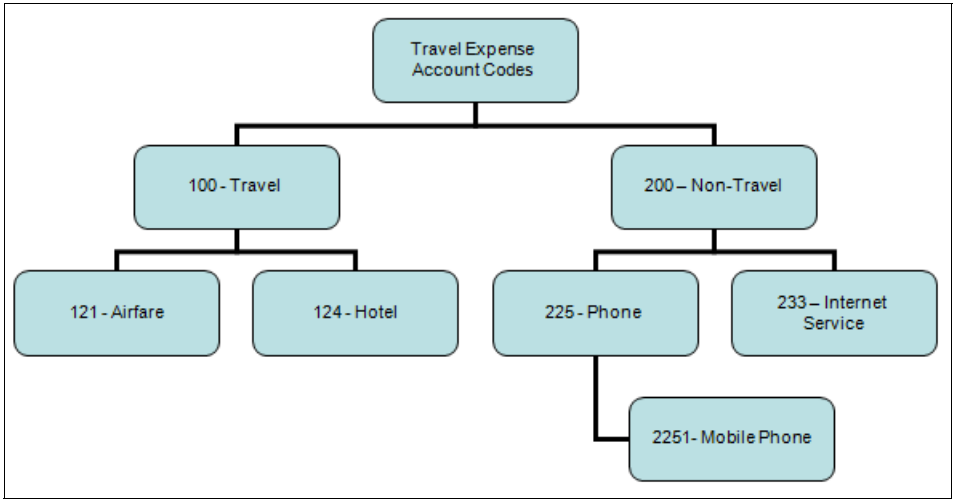


Figure 2-4 Hierarchy of expense accounting codes

Hierarchies can be consumed by a number of enterprise information systems, but are often used by business intelligence systems, for use in reports that roll up data. For example, a report might show expenses summarized by types of expense (travel versus non-travel).

Hierarchies are implemented in InfoSphere MDM Ref DM Hub as a layer on top of underlying InfoSphere Custom Domain Hub hierarchies.There are other types of hierarchical relationships between sets, but those are not explicitly represented as InfoSphere MDM Ref DM Hub objects. For example, a country set might have a property that indicates in which continent that country resides,

and that continent might be a value in a continent set. This way is known as a *level-based* hierarchy. Level-based hierarchies can also be represented in maps.

2.1.4 Supporting domain objects

In addition to the core reference data domain objects, some supporting objects are used in the reference data domain. These objects range from providing underlying support for core objects (types), to providing objects that link, to providing organizational containers (folders), and finally to providing a set of objects that are linked to the core objects (subscriptions, managed systems) as part of the reference data ecosystem.

Types

Types are used to specify the attributes and properties of reference data sets and maps. All reference data sets and reference data maps must have a type. InfoSphere MDM Ref DM Hub provides a default type for sets and maps.

A type can be customized for a specific set or map. The customizations for a set include the following items:

- Additional properties at the set level

  For example, a URL property can be added to reference the external standard for a set. Properties can be string, text, Boolean, URL, integer, date, time stamp, and other sets.

- A validation routine for the code of a set

  Validation routines are implemented as regular expressions. A validation for a two-character value can be implemented as follows:

  `[A-Z]{2}`

- Additional properties at the value level

  For example, a country set might have two additional properties, latitude and longitude. Each property might also have additional attributes:

  - Whether this property is also a key of the set

    Properties marked as keys are effectively concatenated together when used in a map, or when a set is imported and exported.

  - Whether the property is required
  - Whether the property is unique
  - A validation routine for the property (similar to the one at the set level)

Maps are also typed and they allow for extensibility at the value map level. For example, you can add a Boolean attribute called “verified” to a value map, indicating that the particular value mapping has been verified by a data steward.

Managed systems and subscriptions

Managed systems are enterprise information systems that are providers and consumers of reference data. Subscriptions are used to define which reference data is being consumed by a particular managed system. Applications that distribute reference data can use subscriptions to determine which managed systems should receive the reference data they require.

The managed systems objects contain information about what sort of enterprise information system is using or providing reference data (and can be extended generically with attributes like host name, and so on). Subscriptions can include the following information:

- Which managed system the subscription is for
- Frequency for which these systems want to receive updates
- Which delivery mechanism to use
- Which format to use to send reference data to the subscribed system (CSV or XML)
- Which lifecycle state (approved, draft, test, and so on) is of interest
- The last update of a managed system for this subscription

Folders

Folders are a convenient way to organize sets in the InfoSphere MDM Ref DM Hub. Sets can be placed into individual folders, for both viewing convenience and for security (different access rights can be assigned to different folders). In the InfoSphere MDM Ref DM Hub, folders are similar to folders that are used in email, directories that are used in file systems, and so on.

2.1.5 Supporting services

Besides the standard create, read, update, and delete functions on objects, the InfoSphere MDM Ref DM Hub includes additional services that are specific to the RDM domain. These services address the lifecycle, security, and stewardship of the RDM objects.

Lifecycle

The InfoSphere MDM Ref DM Hub core objects (sets, maps, and hierarchies) support two types of lifecycles:

- Approval and lifecycle state:

  The core InfoSphere MDM Ref DM Hub objects can have a well defined lifecycle that specifies the governance over the state of a reference data object. The lifecycle specifies how these objects are updated over time, and by whom. The current state is where the object is in the lifecycle in particular, how these objects are updated, and their current status. Some example states are *draft, pending approval, approved,* and *retired.*

- Version

  Reference data can support multiple instances of a set, map, or hierarchy. This type is useful in the following situations:

  – Major changes are introduced to these objects.

  – Consumers or providers of reference data are fixed on a particular version of a reference data object and cannot be easily changed.

  For an example, a particular application might work with version 1.0 of a set of employee codes and cannot be upgraded to work with version 2.0 because the data for that application cannot easily be updated.

Stewardship and security

The InfoSphere MDM Ref DM Hub implements a security and stewardship model on top of reference data objects and their lifecycle states. Different users can have access to different objects, based on the ownership attribute of the object (which is a list of groups that have access to the object). For objects with lifecycles, users can be placed in different roles in the lifecycle process (steward and approver, for example).

The InfoSphere MDM Ref DM Hub supports a set of basic predefined roles that can be mapped to groups in an enterprise directory (LDAP), as follows:

- Administrator: Can perform all InfoSphere MDM Ref DM Hub operations, including defining types.
- Data Integrator: Can set up managed systems and do import and export of InfoSphere MDM Ref DM Hub objects.
- Data Steward: Can create, update, and import and export sets, maps, and hierarchies, and can manipulate folders.
- Approver: Can approve changes to sets and maps.

Import and export

Reference data is prevalent in enterprise systems. For reference data to become *managed* reference data, it must be imported into the InfoSphere MDM Ref DM Hub for stewardship, managed there, and then exported back to the enterprise systems. To work with existing reference data (sets, maps, and hierarchies), InfoSphere MDM Ref DM Hub supports services to import and export reference data in both CSV and XML format. Transform operations can then be used to change the exported data into the format consumable by the enterprise systems (database tables, properties files, and so on).

2.1.6 Supporting Custom Domain Hub services

Because the InfoSphere MDM Ref DM Hub is a Custom Domain Hub application, the InfoSphere MDM Ref DM Hub can avail itself of common Custom Domain Hub services (security, persistence, and hierarchies). The behavior of the InfoSphere MDM Ref DM Hub can also be enhanced (behavior extensions) or linked to transaction subscribers (event notification).

2.1.7 Batch utilities

The InfoSphere MDM Ref DM Hub provides command-line utilities to import and export reference data sets, maps, hierarchies, and types. These utilities can be used to automate the integration of reference data into the InfoSphere MDM Ref DM Hub, and the distribution of reference data out to enterprise systems.

2.2 Enterprise system integration

One of the critical functions of InfoSphere MDM Ref DM Hub is to interact with the reference data that is found in other enterprise systems. This section explores how InfoSphere MDM Ref DM Hub obtains reference data from key enterprise information systems and how those managed reference data objects are then used in conjunction with those enterprise information systems. In particular, this chapter explores integration with master data management, SAP, taxonomy and content management, and data warehouse systems.

2.2.1 InfoSphere Master Data Management

The InfoSphere Master Data Management (MDM) Custom Domain Hub hosts the reference data domain, but the core server itself has some other interaction

patterns with MDM. In particular, InfoSphere MDM Ref DM Hub interacts with MDM for the following key functions:

- Management of MDM code tables. MDM itself has code tables for several items (customer type, types of privacy policy, address type, and so on). InfoSphere MDM Ref DM Hub can be used to manage them as reference data sets in InfoSphere MDM Ref DM Hub, and export those as code tables to MDM.
- Management of maps used to import data from different sources into MDM. As data is moved from source systems to MDM, reference data maps are used to transform reference data from a source system format to the MDM format for reference data. Usually this process is done as part of an extract, transform, and load (ETL) job loading data into MDM from source systems.
- Management of maps used to export data from MDM to target systems. This function is the opposite of the use case previously mentioned, where master data containing reference data is exported from MDM into consuming systems.
- Runtime transcoding of reference values used in transactions from bespoke systems that create, update, or retrieve data from MDM. As transactions are executing against MDM, the reference data in the transaction (for an update) might be in the format of a source system. MDM could call into RDM to read the map of values from the source system reference data to the values in InfoSphere MDM Ref DM Hub, and then transcode the reference data to the MDM value.

2.2.2 MDM and SAP

IBM InfoSphere Master Data Management Server is a repository that can centralize and manage an organization's critical master data entities such as customer, product, supplier, and more. The centralization of these entities creates a single view of customers and products that results in better service, improved customer satisfaction, and improved relationships with partners and suppliers. Because many, if not all, of the organization's applications (SAP applications, for example) and business processes operate on these entities, a reliable and flexible delivery of the master data is a key characteristic of the solution architecture. This section describes an MDM-SAP reference architecture to help you understand how IBM InfoSphere Master Data Management Server and the InfoSphere MDM Ref DM Hub can work with SAP regarding the management of customer data. The integration approach demonstrated here can also be applied for other business objects managed by the MDM Server (product, supplier, and so on).

Figure 2-5 on page 47 illustrates a scenario that includes both directions. Customer data is managed in MDM Server and sent to SAP. An SAP transaction

is used to add, for example, the tax ID to the customer record. This additional information must be sent to MDM Server to update the central customer entities.

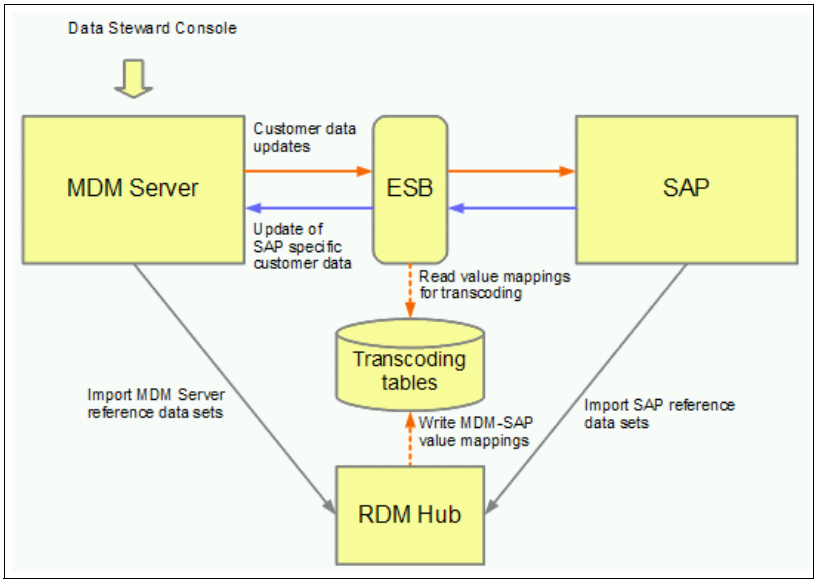


Figure 2-5 MDM-SAP system integration

A possible implementation of this architecture performs the following steps:

1. The customer data is created or updated with the Data Steward Console and saved on the MDM system.
2. MDM Server behavior extensions create an SAP customer ID (SAP KUNNR) for new records and send the customer data to a JMS topic.
3. An enterprise service bus (ESB) mediation flow reads the customer data from the JMS topic, performs the transcoding for country, province, and other codes of the record and calls the WebSphere adapter for SAP, which then sends an SAP IDoc containing the customer record to the SAP system.
4. If the SAP system provides additional information, for example, a tax ID, that must be stored in the customer record within MDM Server, SAP sends this information to the ESB, which then updates the customer record through MDM Server web service calls.

The bus implementation must translate various MDM Server-specific codes into the appropriate values used in the SAP system. The InfoSphere MDM Ref DM Hub is used to import the MDM specific values and the SAP specific values, define mappings and export the mappings to create the transcoding tables to be read within the ESB mediation flow.

2.2.3 Taxonomy management for ECM pattern

In Enterprise Content Management (ECM) systems, taxonomies are used to classify content to make it easier to find in content delivery systems. InfoSphere MDM Ref DM Hub can be used as a component in an enterprise taxonomy management solution for content management and delivery. This challenge is complex often for the following reasons:

- Many older systems are involved that have manual processes for managing reference data and taxonomies for content tagging.
- These systems often do not easily support the introduction of new versions of the reference data and thus cannot accept automatic feeds from the trusted source.
- Consuming systems often need to obtain superset and subset, and join and transform multiple objects from InfoSphere MDM Ref DM Hub.
- There can be hundreds of concepts and many variations of each that must be mapped to each other. An example might be a simple list of industries that might have three or four variations for the business to use, one for marketing, one for sales, and one for finance.
- These ECM systems often support a business function that must change rapidly and frequently in response to changes in the market. Therefore, the process and tools of applying taxonomy changes into the consuming systems should be simple and easy to use.

Figure 2-6 shows the components of a taxonomy management system (TMS) that includes InfoSphere MDM Ref DM Hub.

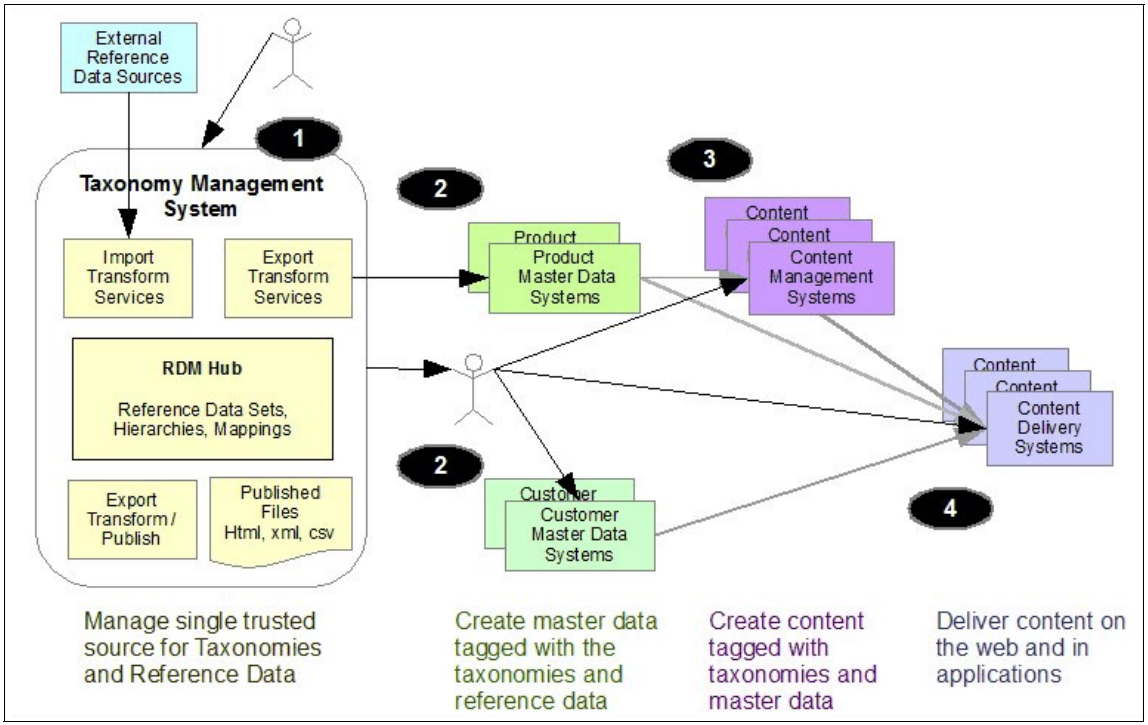


Figure 2-6 Components of a taxonomy management system with InfoSphere MDM Ref DM Hub

This solution pattern has the following key components, as shown in the figure:

- Configurable import, export, and publishing services that are built on base services of InfoSphere MDM Ref DM Hub and are part of a taxonomy management system (1).
- A transform component that extracts reference data sets, hierarchies, and mappings from InfoSphere MDM Ref DM Hub, and publishes reports and feeds. This published reference data that can be consumed by down stream system administrators.
- A services component that can combine InfoSphere MDM Ref DM Hub output and deliver it to a system that directly consume the output.
- Synchronization with various master data sources that use the InfoSphere MDM Ref DM Hub reference data (2).
- Because ECM systems often require a human to implement a change in a taxonomy, the TMS solution needs to include notifications, ideally as part of an integrated business process.

- Controls in the content management systems (3) that allow the following tasks:
  - Allow authors to select values from reference data lists and tag content.
  - Allow content owners to identify updated needed when new versions of a taxonomy is released.
  - Allow for the code from the taxonomy to by syndicated to the delivery platforms.
- Delivery systems (4), such as web applications, portals, search engines, and document management user interfaces, that use the official taxonomies.

  This component involves close change management coordination with the TMS and the ECM systems.

Not shown in Figure 2-6 on page 49 is the need for notifications and integrated workflow management across systems.

Figure 2-7 shows another view of a possible implementation.

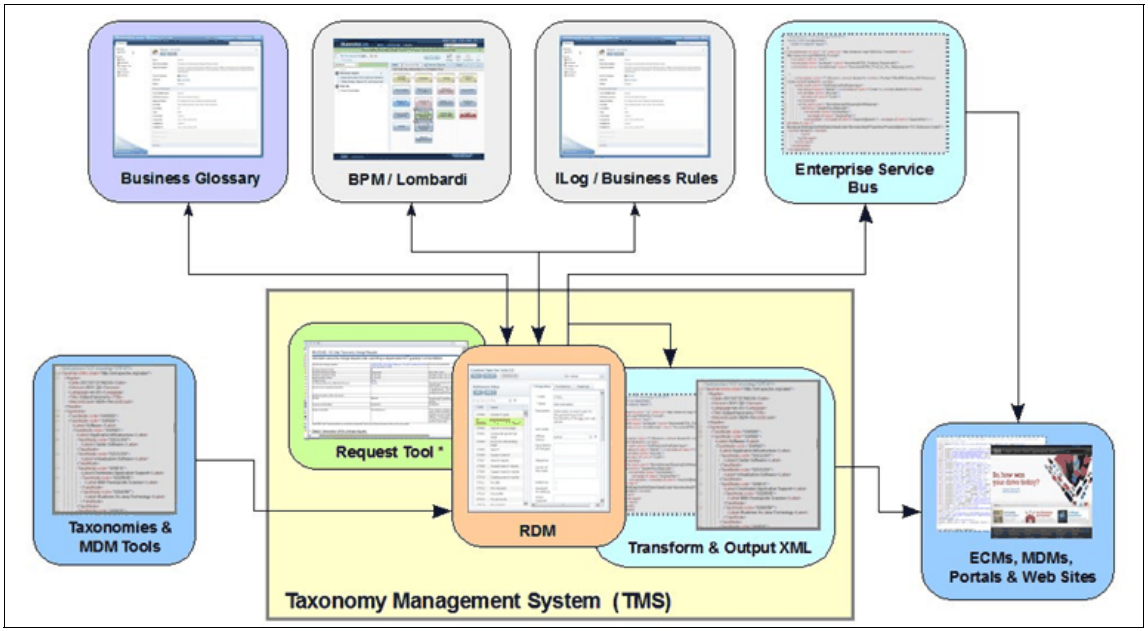


Figure 2-7 Taxonomy Management System and InfoSphere MDM Ref DM Hub pattern

2.2.4 Data warehouse

Data warehouses are used as the enterprise repository for business intelligence data. Information is collected from a variety of operational systems (typically through ETL) to provide a common repository for reports, dashboards, and analysis of enterprise information. Reports are usually rolled up based on common dimensions, such as country, region, product type, and so on.

Reference data plays three key roles in a data warehouse environment:

- Providing a consistent canonical definition of reference codes to be used in all tables in the warehouse. Reports are typically organized around common classifications (sales by country, types of transactions, and so on), therefore, having the reference data defined in InfoSphere MDM Ref DM Hub, and then pushed into the warehouse enables consistency and governance around reference data dimensions.
- Delivering a well managed set of hierarchies for reporting and analysis. Many reports are interactive and hierarchical in nature. For example, you might have the following data:
  - Sales by region
  - Sales by country within a region
  - Sales by province within a country within a region

  Regions, countries, and provinces can be represented as reference data. The relationship between regions, countries, and provinces is a hierarchy over that reference data. InfoSphere MDM Ref DM Hub can be used to ensure the consistency, integrity, and governance of that hierarchy before the hierarchy is published to the warehouse.
- Ensuring that reference data from source systems is mapped to the common canonical reference data used in the warehouse as part of the ETL process. The standard pattern for moving transactional data (and other dimensions, such as master data) into a warehouse is to use an ETL tool, such as InfoSphere Data Stage, to take updates from source systems and transform and load those into the dimensional tables in a warehouse. The representation of the reference data in the various source systems will likely vary from the canonical representation in the warehouse. For example, a source system might represent countries using a numeric code like 93 (dialing code for Afghanistan) or using the ISO-3166-1 two-character standard (AF), while the warehouse might use the ISO-3166-2 three-character standard (AFG).

  InfoSphere MDM Ref DM Hub is ideal for managing the sets that are associated with the different source system and the warehouse, and for empowering business data stewards to create the maps and version them for the mappings between the sources and the warehouse. The resulting maps

can then be exported to a lookup table used by the ETL system to transform reference data in source systems to the canonical representation for a particular warehouse.

A visualization of this architectural pattern is shown in Figure 2-8.

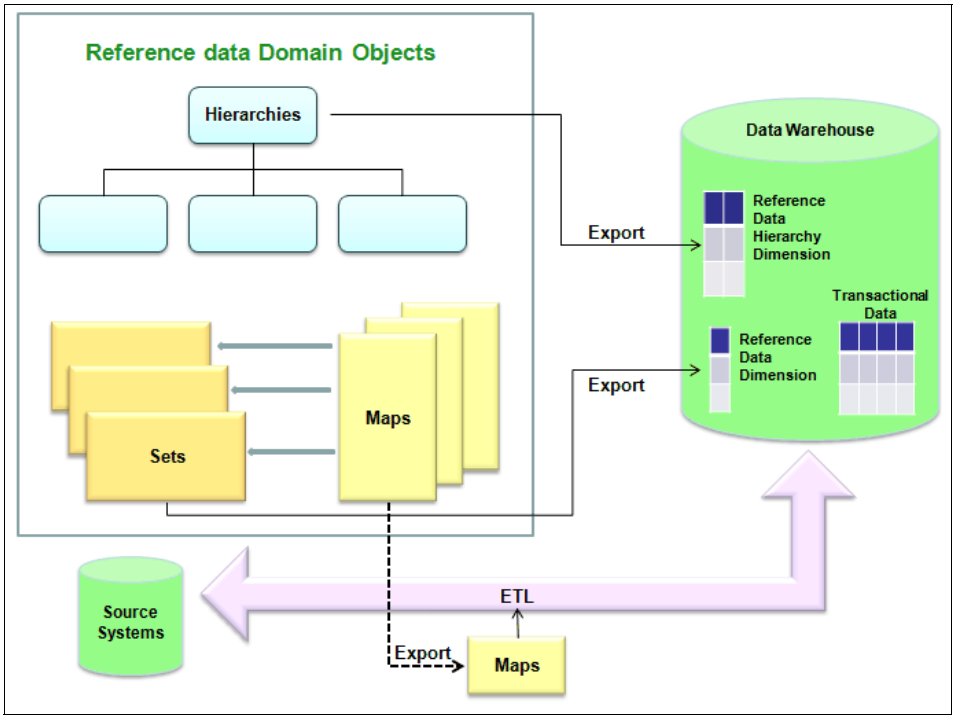


Figure 2-8 InfoSphere MDM Ref DM Hub integration pattern with a Data Warehouse

3

Planning a RDM project

This chapter describes the considerations of planning a reference data management project with IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub). The purpose is to detail the necessary resources and provide an overview of the activities that must be accomplished to provide enough factual information and understanding of reference data management to effectively drive and manage an implementation. More detailed descriptions of activities follow in subsequent chapters.

There are many comprehensive project management tools and methodologies in use; this chapter does not attempt to describe or to follow any specific methodology. This chapter describes general phases of a project and the considerations that are particular to reference data management with a checklist of inputs and output from each phase.

The roles and skills required of the project team members are outlined once in 3.4.2, “Transforming business roles to RDM roles” on page 63 to avoid repetition. It will be clear from reading the tasks involved in planning an InfoSphere MDM Ref DM Hub implementation which roles are applicable where.

3.1 RDM methodology overview

The structure used here to introduce a methodology for planning and implementing an InfoSphere MDM Ref DM Hub project is taken from the IBM Information Management Unified Method (IMUM). Only the basic phases are described, and only in the context of an InfoSphere MDM Ref DM Hub implementation. This methodology overview does not attempt to describe or explore IMUM, we only consider an operational and technical stream relevant to reference data management (RDM). Project management activities can be left to any of the preferred tools and methods that you choose in any implementation.

The phases are as follows:

- Analyze: Define what the RDM has to accomplish.
- Design: Identify the RDM components and their dependencies.
- Configure: Build the RDM solution.
- Deploy: Test and move into production.
- Operate: Run and develop RDM.

3.1.1 Analyze

The *analyze* phase defines what the RDM solution must accomplish in terms of functionality, understanding, and using the features of InfoSphere MDM Ref DM Hub, for example, set mappings, value hierarchies, and the non-functional requirements such as usability and performance.

During this phase, the most common working practice involves workshop sessions where the high level business requirements and any available use cases are matched to the functionality and processes of InfoSphere MDM Ref DM Hub. Repeating analysis for various requirements and use cases raises a set of common questions and answers that can be used to formalize the results of analysis.

The results of this work can give a picture of the business processes that use reference data, the amount of duplication found in the various code sets, and the amount of translating that is currently carried out to match or roll-up values from one set to another. It soon becomes apparent that some sets are duplicated in various places and systems and can usefully be collapsed into one “master” set. However, there might be business reasons why two or more editions or versions of sets should exist separately and continue to do so.

For example, consider “product” as a set of values. There are three product set instances found, as shown in Table 3-1.

Table 3-1 Product sets

Set owner	Set purpose	Governance
Sales product codes	List of all products that might be offered for sale	Owned by Sales and Marketing. Agreed with Manufacturing
Manufacturing product codes	List of all products that are manufactured; many products that are used as multiple components in other products	Owned by Manufacturing. Aligned with materials stocks and used in production scheduling
Accounting product code mapping	List of products that are costed, might be a mix of components and product. List of sales products by profit margin	Owns neither set but owns the mapping between them to use in costing work

For each set, an outcome from analysis should the following items:

- Code set owner: Which department or system owns this set
- Code set control: Specify whether set is an internal set or an external set, for example ISO currency codes
- List of values and descriptions held in the set
- List of applications where the set is found
- List of business processes that use this set
- For each set a list of related or similar sets
- List of business and system processes where set values are mapped or transcoded

The activity of analysis also gives identification and familiarization:

- Identification of issues in the business processes

  Identifies where the existing use or maintenance of reference data sets is giving problems. These problems might be in complicated interfaces or in the need to manually compare sets and codes. Noting these problems gives an input into prioritizing later development.
- Familiarization of RDM concepts for the business and technical teams and to the wider client environment

3.1.2 Design

In this chapter the *design* phase is confined to configuring InfoSphere MDM Ref DM Hub. There is a fixed data model in InfoSphere MDM Ref DM Hub that is designed to support a wide range of functionality for the handling and maintenance of reference data sets. As an overview, consider the functionality of InfoSphere MDM Ref DM Hub to do the following tasks:

- Manage reference data sets.
- Relate sets through mappings.
- Make subscriptions to link to external systems.
- Import and export sets.

The design phase makes decisions on exactly how to configure InfoSphere MDM Ref DM Hub. The first step in designing is to decide from the requirements and data sets found in the analyze phase, whether all business requirements can be met.

If significant requirements lie outside these areas, you can customize InfoSphere MDM Ref DM Hub by adding and extending the data model and by building custom services for add and update functions.

3.1.3 Configure

In the *configure* phase, the configurable parts of InfoSphere MDM Ref DM Hub are as follows:

- *Security and access* includes creating user accounts and associating users to groups, and groups to roles.
- *Configuring components* include adjusting transaction time-outs, configuring batch import and export, and customizing the InfoSphere MDM Ref DM Hub user interface web page.
- *Defaults* are date and time formats; default start and end dates used in sets, mappings, and import and export wizards; the character used in separated value files (usually comma separated value CSV).
- *Managed systems* is a term for any systems that are external to InfoSphere MDM Ref DM Hub where an integration is built to InfoSphere MDM Ref DM Hub. Managed systems are integration points held in InfoSphere MDM Ref DM Hub where the configurations and properties are stored that allow integration with a particular external system. The owners of the external system must collaborate with the integration from their side.

3.1.4 Deploy

The *deploy* phase is a set of tasks that moves the InfoSphere MDM Ref DM Hub system from a development environment into a production environment.

The technical specialists and client IT staff collaborate to build and test a working configured production environment. This setup is a repeat of previous installations of testing and development environments and incorporates the lessons learned of what specific procedures are needed in the unique client IT environment. All the configurations and customizations that are built into InfoSphere MDM Ref DM Hub are included in the installation. The joint effort with the client's deployment team should execute and document the mechanism of transferring the code from the test or development environment instances into production.

With the environment prepared, an initial load of the reference data sets can take place. All import files that were tested can be used to load reference sets, including hierarchies within and between sets. At this point, the solution is connected to the production environment.

The complete description of deployment is described in Chapter 7, "Implementation" on page 159.

3.1.5 Operate

The *operate* phase is after the actual transition and use of the solution is established in production. The final implementation activities include a comprehensive review of all of the project documentation and an evaluation of the system implementation results.

The purpose is to ensure the InfoSphere MDM Ref DM Hub system is running smoothly in production and that the baseline system behavior is captured for future reference. Part of the implementation effort will have established business and IT processes, enabling the future adoption of new reference data sets, mappings, and interfaces to be added into InfoSphere MDM Ref DM Hub.

3.2 Iterative implementation of InfoSphere MDM Ref DM Hub

As with all software implementations, various methodologies are used to support various development styles and business practices; no one process suits all situations (iterative or waterfall or any process in between). The methodologies themselves are not described here, only the use of iterative techniques. The InfoSphere MDM Ref DM Hub solutions are suitable for an iterative implementation process for the following reasons:

- The software is comprehensive and configurable, and can be extended and customized.
- Each functional area can be treated separately for use and development. A small sample of code sets can be created quickly.
- There is a dependency on the precedents of which artifacts are created in InfoSphere MDM Ref DM Hub, which becomes immediately apparent.
- Small numbers of sets can be manipulated in mappings, imported, exported, and copied. If they meet acceptance criteria for operational InfoSphere MDM Ref DM Hub, they can be preserved. If not, they can be easily discarded.

This approach has the advantages of putting project team members and business users into an environment where experimentation is possible and there is an emphasis on communication, simplicity, feedback, and easier compilation of documentation.

For an iterative approach, the overall functionality of InfoSphere MDM Ref DM Hub is described here in discrete features. These building blocks are created directly into the InfoSphere MDM Ref DM Hub system.

Table 3-2 on page 59 lists five sample iterations that are based on groups of individual features that are related by functional area and are few enough to consider together. In all implementations, the specific use cases and requirements influence the exact order and content of iterations. Any iteration can be repeated many times and include steps from previous iterations.

Table 3-2 Sample iterations

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Add set type	Add set types with properties	Prepare CSV files and import sets	Add mapping types with properties	Move sets through one or more lifecycles
Add mapping type	Add Sets with tree hierarchies	Export sets	Add mappings with some values mapped	Add set versions
Add two sets	Add sets with level hierarchies	Add translations	Add mappings with many to one mappings	Copy sets
Add a mapping between the two sets	Create users and groups	Add managed systems	-	Merge sets
Make folder structure	-	-	-	Manage owners

The benefits of using iterations are as follows:

- Team members can quickly learn the capabilities on InfoSphere MDM Ref DM Hub and discard any work that was illustrative or experimental.
- Concurrently a more formal approach can build work in the iterations that can be saved and documented to become part of the established solution.
- The results of repeated iterations give a basis on which to build test cases for functionality and user acceptance.
- Every iteration tests and proves that the solution architecture is stable.
- The durations of iterations become well known and as iterations are completed, the results give project management good input into planning, estimation, number of iterations, and an indication of the likely project schedule and effective collaboration techniques.
- Using an iterative approach engenders a collective ownership; as the architecture and processes of the system settles, the team can move towards a complete working solution.

3.3 Business requirements

InfoSphere MDM Ref DM Hub has a concise focus on managing and maintaining code sets throughout an enterprise. The business requirements found in many implementations therefore have much in common. The following list and discussion of business requirements cannot be exhaustive, because there will always be special cases, but these do cover a large number of requirements found in real implementations. In planning any specific InfoSphere MDM Ref DM Hub project, you can expect to find a significant number of these requirements expressed by the business.

Reference data management requirements

Identify and collect the following requirements for reference data management:

- Requirement to maintain and manage a standard view of reference data that is accessible and accepted throughout the enterprise.

  In meeting this requirement, consider each code set that is stored in InfoSphere MDM Ref DM Hub and from the population of data sets. Select only those that have a truly enterprise-wide interest. Each of these enterprise sets contains only the number of values that are necessary. For example, a set of mandatory Customer Status codes might require all values in use throughout the enterprise included, and can be described as canonical (includes all possible values). In another example, the enterprise is interested in sales at a country level. The enterprise sales area set must include only countries where the enterprise does business. This will have a smaller number of values than in the address country code set.

- Requirement to maintain and manage all reference data sets that are used in more than one business function or IT application.

  There are many more code sets and variations of code sets found in various business areas, functions, and IT systems that are local. They might not have an importance to the enterprise view, but their complex management and mappings require a solution.

- Requirement for a simple process to actively manage reference data in multiple systems.

  The InfoSphere MDM Ref DM Hub enables a single point of authorship and revision. Code sets in InfoSphere MDM Ref DM Hub can be set by subscriptions to accept changes or propagate changes from and to external systems. The requirement is stated as making changes to reference data and distributing to downstream systems.

- Requirement for a simple process to match the values across separate code sets. Now InfoSphere MDM Ref DM Hub compares any two reference data sets in one mapping. You can easily find the equivalent value in a similar set

or different instance of the same set. This feature can help you automate in-flight reference data transcoding to convert values between different systems and formats.

Requirements to simplify complexity

The following examples are actual cases where the requirement was stated as "*We need to make our applications and processes simpler.*" These requirements can all be enabled by using InfoSphere MDM Ref DM Hub to isolate and manage the reference sets involved:

- Two hundred batch jobs have separate code equivalents.
- There is siloed and parallel work duplication.
- Different sources use different formats.
- Many business processes are required.
- Authorship is a business task but now given to IT to implement every case.

Data governance and compliance requirements

The data governance and compliance requirements have the following characteristics:

- A need to demonstrate repeatable processes for enforcing data quality, ownership, auditing, reporting, storage, life-cycle management, security, sharing, and classification.
- A need to incorporate into enterprise information management processes, for example, aligned in common logical data model, complete information catalog, metadata standards, and Information Management maturity model.
- Are proved to various standards, for example, enterprise data warehouse, Solvency II, New Core banking systems, Foreign Account Tax Compliance Act (FATCA), Basel 111.

Data integration requirements

The following examples are of data integration requirements:

- A need for a common unified classification for many sets.
- A need for common consolidation rules and timings to allow an integrated flow for updates to the data warehouse. This is reference data including mappings and hierarchies.
- Ability to import external codes sets such as ISO and any used by the international or statutory authorities.

3.4 Analysis of data

This section looks in detail at the reference sets that will be managed in InfoSphere MDM Ref DM Hub. For as many candidate reference sets as possible, the aim is to know the following information:

- Reference set owner
- Systems where reference set is used
- Business processes where reference set is used
- All values found in every instance of a reference set
- Where the reference set has interfaces to and from
- Versioning of the reference set if any
- Validity dates of the reference set if any
- Categorization of the reference set, with hierarchy, with compound key
- Canonical or subset, that is, all values in an enterprise (canonical) or a subset used for a particular business process

In the process of getting and discussing this data, from sample data files, interviews, or workshops with business analysts and source system owners, it might be possible to discard sets or agree to adopt one instance of a set and impose that as a standard to take the place of other instances.

The outcome of analysis of this data is then used to provide information that informs other phases and activities in the InfoSphere MDM Ref DM Hub project, specifically the following items:

- A list of business owners and numbers of sets owned.

  This list can often be made into a reference data set itself and added as a property to other sets.
- A measure of the complexity of the sets and any additional data required.

  If many sets are found to need any additional data element, the elements can be included in a reference data set type, or in a linked code table added as a property to other sets.
- An indication of the folder structure and any naming conventions needed to organize sets.
- The number of mapping types needed to categorize mappings and the likely number of actual mappings.

- Which sets are sources and which sets are targets in mappings.

  It is likely that one set can appear as a target in one mapping and a source in another. For example District_Sales_Codes as a source to the target of Country_Sales_Codes; Country_Sales_Codes as a source to the target of Regional_Sales_Codes.
- A list of external systems that can be linked by subscription to receive or propagate codes changes automatically.

3.4.1 Discovering reference data

The focus of InfoSphere MDM Ref DM Hub is on reference data sets and, by their nature, they are distributed throughout business functions and IT applications. There is often no central source from which you can identify all the existing sets. However, somewhere in the enterprise will be awareness of those sets that are causing problems. From this starting point, it is necessary to trawl through as many sources as possible. In discovery workshops, the business analysts and subject matter experts document those reference sets that they consider critical. Next is the task of discovering all sets. The following sources are likely:

- Business analysts and subject matter experts
- Business process descriptions
- Logical data models and data glossaries
- User interfaces, directly on-screen or from design documents
- Physical models and database schema

The amount of analysis and discovery that can be done is limited by the normal constraints of resources and time. However, effort made here will simplify the solution in later stages of the implementation. For example, incorporating newly discovered attributes might require the rework of building new types and assigning existing sets to the new type, which must be achieved by deletion and re-creation.

3.4.2 Transforming business roles to RDM roles

This chapter has an assumption that a project team exists that has is a high level of business and technical skills and a considerable degree of continuity. Typically, the project team is composed of people from the client side and from IBM; their skills together can cover all the necessary subjects around an InfoSphere MDM Ref DM Hub implementation. All the work described is done by the team. Undertaking a project to implement InfoSphere MDM Ref DM Hub in a business is a substantial commitment that requires enabling people during the implementation process and all team members learning from their colleagues.

Some part of the team typically remains after implementation to develop and maintain the solution.

The necessary skills range from a comprehensive knowledge of how the client's business requirements are met in the current IT systems to detailed design and build of databases, interfaces, and services. The headings in this section are generic but well known; they vary from business to business. No numbers are given here because resources vary in availability and skill level. Every business will have a preferred spread of skills over a number of resources depending on availability and criticality.

The usual case is for InfoSphere MDM Ref DM Hub implementations to be staffed with a joint team made up of a combination of resources from the client and IBM with a third-party integrator if required. Any degree of skills transfer from IBM and any integrator to the client staff is possible depending on the client's business practices, from requiring ongoing support in all development and maintenance of the product to a level close to self-sufficiency. The team effort of implementation is a good environment for skills transfer.

Enterprise architect

The enterprise architect role has three views:

- A vision, shared by the business, of the future state that the company wants to achieve and an evolving road map of how IT will grow to support it getting there
- A complete view of the existing business systems in use and their capabilities, ability to integrate, and potential for future enhancement
- A prioritized set of all the shortcomings or difficulties with the present systems functionality to serve all business requirements

The enterprise architect's involvement with the InfoSphere MDM Ref DM Hub project begins with the awareness of the business problem, as described Chapter 1, "Reference data management" on page 1 and continues through requirements analysis, product selection, and the overall implementation and integration. When InfoSphere MDM Ref DM Hub is implemented, the enterprise architect revises the views on the overall architecture by adding the InfoSphere MDM Ref DM Hub functionality and its relevance to business needs and capabilities. The ongoing promotion of use and development of InfoSphere MDM Ref DM Hub is overseen by the enterprise architect.

This role is universally found in the client organization and remains there, acting as liaison with business analysts, and learning the necessary influences, uses and potential of InfoSphere MDM Ref DM Hub.

Business analyst, subject matter expert

Many business analysts and subject matter experts are in any enterprise. Their knowledge of business processes, and associated IT systems and applications gives these analysts and experts a view of the actual use and distribution of reference data throughout. These people are also aware of the quality of reference data and the problems that must be overcome in using reference data. In planning and implementing an InfoSphere MDM Ref DM Hub project, a good practice is to consult as many business analysts or subject matter experts as possible because their roles often encompass subsets of departments, systems, and functions. Involving as many as possible will give the most comprehensive view across the enterprise and include all variations of practice and coding.

This role is universally found in the client organization and will remain there. As the implementation matures, more decisions regarding management of reference sets will be made by the business analyst community because learning and practice of InfoSphere MDM Ref DM Hub will transfer skills from the solution architect to the client team.

InfoSphere MDM Ref DM Hub solution architect

The InfoSphere MDM Ref DM Hub solution architect, sometimes known as the product consultant, designs the implementation based on business requirements and the goals of the client. The role collaborates with other technical and functional project team members to create a well-defined and agreed solution architecture, including any data models, services, interfaces, and custom and configuration work on InfoSphere MDM Ref DM Hub. All requirements and processes that require modifications to the standard product are evaluated by the solution architect for their feasibility and degree of difficulty, and designed into the implementation.

This role is typically supplied to the project by IBM or specialist integrator and their work of translating business requirements to technical requirements is done in close company with the business analysts and technical specialists. There should be complete agreement and documentation of the design. The solution architect working closely with business analysts and technical resources and will provide as much skills transfer as required by the client.

Technical specialist

The technical specialist has overall responsibility for the InfoSphere MDM Ref DM Hub solution construction, implementation, and system integration. This role ensures the solution can run in the client's technical environment, and that after it is installed, the product is configured as in the implementation design to meet the client's business requirements. The technical specialist is responsible for the technical design and correct functioning of any custom work in the solution and testing and integration with the client environment.

This role is typically supplied to the project by IBM or specialist integrator and can work with client technical resources. As with the solution architect, there is a good opportunity in the project environment for carrying out skills transfer as required. Tasks that are done by the role, such as scripting, and programming skills that support database performance tuning and leadership of developer teams, are probably done in-house to an extent; the technical specialists bring their deep specific product knowledge to these tasks.

Developer

The developer has specialist expertise and, during implementation, will create and assemble components to build any customization to InfoSphere MDM Ref DM Hub, write code, and execute unit tests, integration tests, and functional tests.

The developer role is a specialist role that works closely to the detailed designs that are provided by the technical specialist. A worthwhile approach is to use in-house developers, if available, because engagement on the project will build the InfoSphere MDM Ref DM Hub skills, and these skills will remain with the client.

Data steward

The data steward is a business user but with a deeper knowledge of how to use InfoSphere MDM Ref DM Hub in everyday operating tasks, principally through the InfoSphere MDM Ref DM Hub user interface. The data stewards execute tasks such as creating reference data sets, organizing the folder structures, creating types and mappings, and carrying out import and export jobs, which means all the regular maintenance and development of reference data management with InfoSphere MDM Ref DM Hub. A good practice is to introduce people who will be data stewards into the project at an early stage so they understand the solution and provide input on the current custom and use of data and codes, which might differ from existing specifications.

This role is universally found in the client organization and will remain there. Data stewards work with the business analysts and other users to actually operate the system.

3.5 Model design for planning

The model design here is a simplified view of the reference data management components and their relationships within InfoSphere MDM Ref DM Hub. These components are the major components that are considered for planning an InfoSphere MDM Ref DM Hub implementation. For a full discussion on the

complete detailed data model in InfoSphere MDM Ref DM Hub, see Chapter 5, “InfoSphere RDM model design” on page 99.

Figure 3-1 shows a high-level view of reference data management components and their relationships in InfoSphere MDM Ref DM Hub.

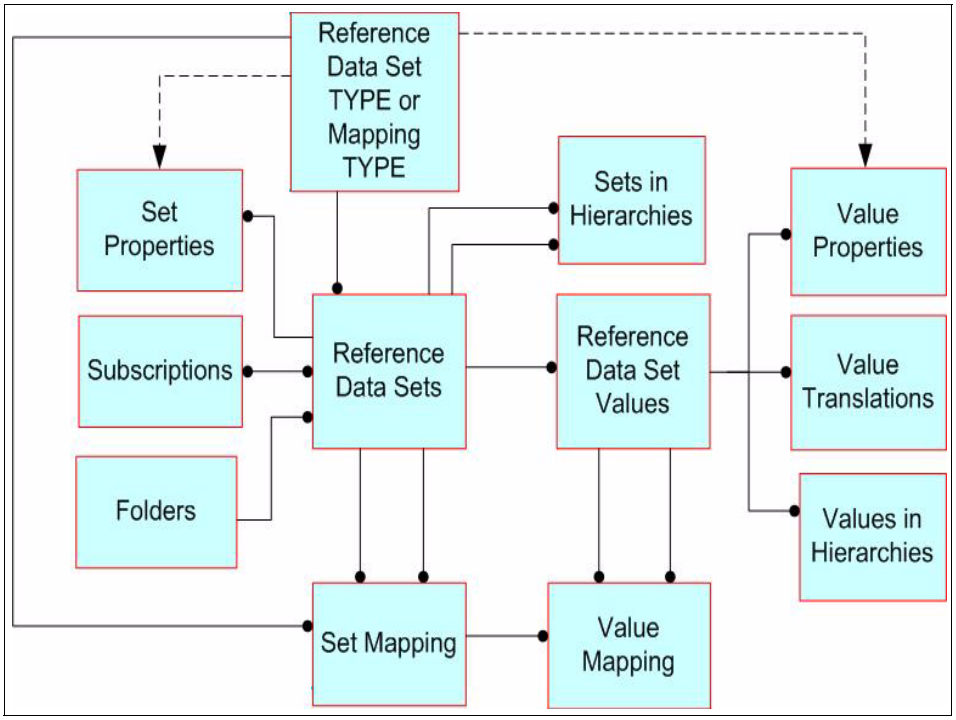


Figure 3-1 High-level view of RDM components and their relationships

Reference data set type or mapping type

A data set type works as a definition. The data set type contains properties that apply to reference sets at both set-level and value-level. The InfoSphere MDM Ref DM Hub system contains many instances of sets. Every set must belong to one set type. The set type properties define and store metadata about the set.

The initial InfoSphere MDM Ref DM Hub installation has two basic set types: one for sets, and one for mappings (known as Default Type), which can be used for many sets and mappings. If more data must be stored to define or control a set and the set values, then a new set type must be created. Examples of custom properties are as follows:

- Reason_Changed_Code: A mandatory code added when a value is expired
- Source_System_Code: A field containing the origin of the set or value

Figure 3-2 illustrates a reference data set type that contains properties applied to all associated sets. Types can be created with custom properties added.

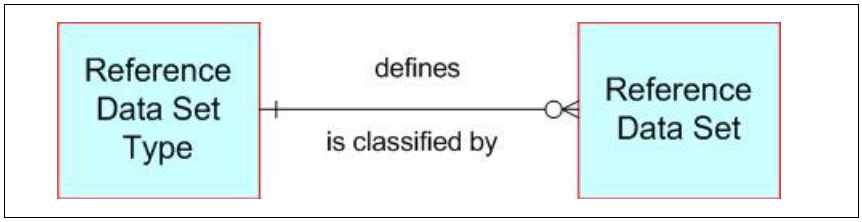


Figure 3-2 Reference data set type contains properties applied to all associated sets

Figure 3-3 illustrates the data mapping set types using an identical structure to reference data set type. The data mapping set type can be customized.

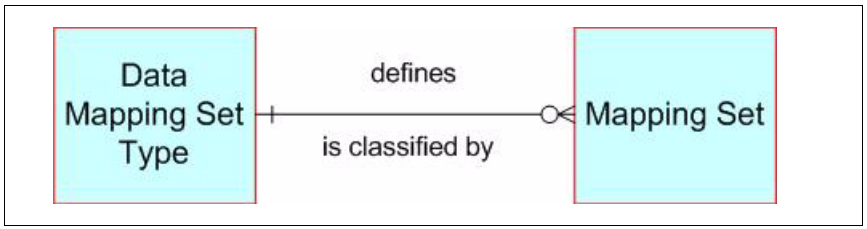


Figure 3-3 Data mapping set types use an identical structure to Reference data set type

The simplest approach is to create one type for every reference set so the structure in Figure 3-2 is always 1:1. Although this approach is simple to use initially, it can lead to an excessive amount of types to manage and repeated information.

Set types and mapping types can be organized by owner or by function or any classification system that works in a particular business. Consider using a naming convention for sets, types, and mappings. Naming conventions also apply to folders in the same way.

Reference data sets

Reference data sets are the instances of code tables that are managed in InfoSphere MDM Ref DM Hub.

Subscriptions

Subscriptions are a means to link to external systems, referred to here as managed systems (Figure 3-4). This mechanism allows automated import and export of changes between InfoSphere MDM Ref DM Hub and the managed system.

Figure 3-4 Reference data sets are linked to any number of external systems through a subscription

Any set can be linked to any number of external systems through one subscription. The flexibility of using subscriptions to manage external systems requires understanding the supported business processes and setting permissions on whether InfoSphere MDM Ref DM Hub is allowed to perform automatic updates in an external set or whether InfoSphere MDM Ref DM Hub will accept automatic updates from external sets. The most common scenario is that changes are authored in InfoSphere MDM Ref DM Hub and propagated to external sets.

All versions of a set are carried in the subscription, therefore, the external set must be aware of the versioning that is used or have a limitation built on what data to accept.

Folders and naming conventions

A folder structure in InfoSphere MDM Ref DM Hub is used to organize reference data sets. The folders can be organized into a hierarchy so that users can create as many folders and subfolders as they want. Each reference data set can exist in only one folder, different from some file explorers where copies can exist in multiple locations. It is possible to copy and move a set to another folder but InfoSphere MDM Ref DM Hub enforces a different name. The copy is intended for shortening the set authoring process, not for allowing duplicate sets.

The folder structure is seen in an InfoSphere MDM Ref DM Hub user interface and can be modified by dragging and dropping. When a new set is created, it is then listed under the folders and can then be dragged into the appropriate folder.

Imposing a naming convention on the organization of folders is beneficial. This convention establishes a consistent and logical way to distinguish folders and their contents. Many businesses already have their own formal or informal

naming conventions to draw from and understand the subject areas and departmental boundaries. A strong naming convention enables users to browse the content of InfoSphere MDM Ref DM Hub more effectively and to add their own folders and sets without having to re-think the process of structure.

An example of a naming convention in current use takes two views of categorization, a set view and a departmental view.

Figure 3-5 shows a fragment of naming convention used in folder structure and also for naming of types and sets.

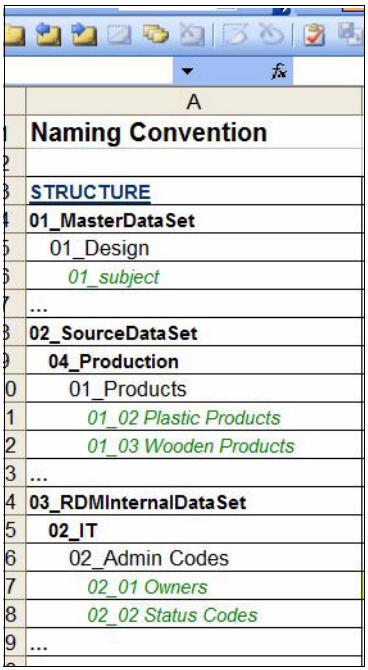


Figure 3-5 Fragment of naming convention used in folder structure

Set category

Sets are split into one of three categories:

- Master set: This set's content is regarded as the enterprise standard.
- Source set: This set's content serves one or more business needs and may be mapped to other sets and also to the master set.
- RDM internal set: This set is used within InfoSphere MDM Ref DM Hub, for example a set of status codes.

Departmental category

Departmental category is shown in the illustration by the second level where the business structure is represented, and can be seen acting as a master data set, a source data set, or an RDM internal data set. The numbering given to the departments themselves is another taxonomy within the enterprise. The following values shown are samples:

- 01_Design
- 04_Production
- 02_IT

There is no constraint on which department can use which set category.

Reference data set values

After sets are established, an unlimited number of values might be added to any of the sets. Values can be added either manually or by using the import functionality. When adding a value or values to a set, you are constrained by the properties that are defined in the associated set type that is established for this set. When adding values, you can see the conditions of the properties fields, for example, whether they are mandatory, forced to be unique, part of a primary key, or have a default imposed.

Values translations

Values can be stored in an unlimited number of languages; the translation is a manual task but the input can be manual or done by the import function. The languages available are held in a reference set as part of InfoSphere MDM Ref DM Hub administration and are used to qualify the language used in other sets.

Values in hierarchies

Within any reference set, the values can be arranged in a tree such as hierarchy. A set might have an unlimited number of hierarchies and each one might contain all or some of the values in the set.

Hierarchies in sets are useful where there is a common set but different views are required, for example the GroupOrganizationCode, split by function.

Figure 3-6 shows one set represented in three hierarchies. The first hierarchy (on the left) contains all values; the second hierarchy (middle) contains only values that are used in Finance, the third hierarchy shows all values in a different order.

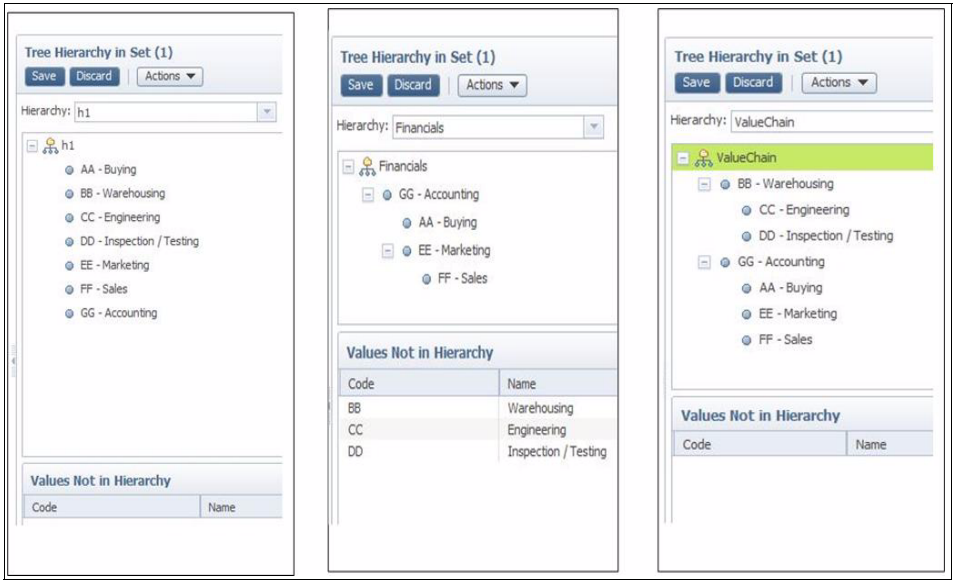


Figure 3-6 One set represented in three hierarchies

Sets in hierarchies, level-based hierarchies

When creating value properties in a reference data set type, you may choose to define the property by selecting another pre-existing reference data set. Only the values found in this other reference data set can then be used to populate the property.

There are the choices of format when creating a property. See Figure 3-7 on page 73. Selecting Reference Data Set allows you to nominate any set from

which values might be taken to populate the property. Only values from that set are allowed.

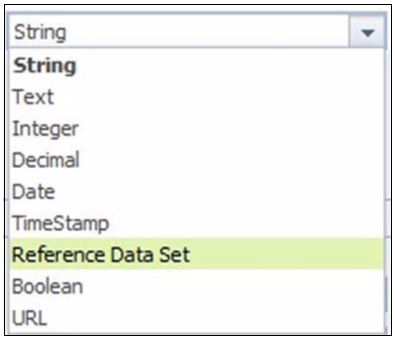


Figure 3-7 Property format

An example of using set hierarchies is to answer the business requirement:

> "*All Marketing reference data sets must have the number of the Approval Board included in the set data*"

To satisfy this business requirement, you can make a property at set level named Approval Boards. This property can be a free text field allowing anything to be written in it. There is also the option of choosing to link the property to a reference data set named Approval Boards. Only values found in Approval Boards can then be used in the set property.

Figure 3-8 shows reference data set MARKETING with the property Approval Board controlled by the contents of another reference data set. Here the values can only be AB001, AB002 and so on.

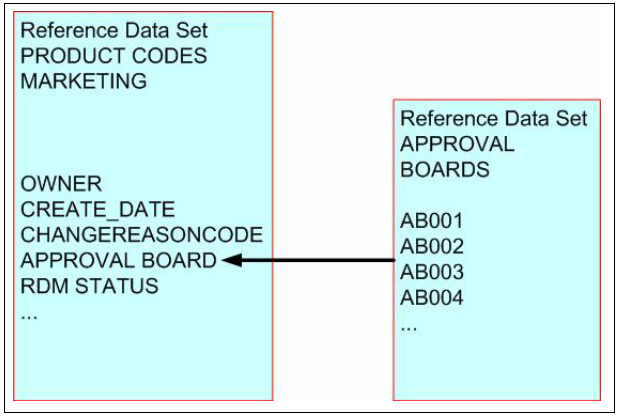


Figure 3-8 Reference data set MARKETING with the property Approval Board

Set mappings and value mappings

As with reference data sets, mappings cannot be created unless there is a pre-existing data mapping set type. As an example of the requirement and use for mappings, consider the case where analysis shows many instances of a product code set within a business. Several departments hold their independent versions of product codes for their own internal use, and there is a business requirement to communicate and transcode between the departments and their IT systems.

It is apparent that all instances of the product code set have a collection of common metadata, and this is needed in the mappings between any two sets. The common data is, for instance, a field containing a URL that points to a web page, which describes product usages. The URL property can be added to a mapping type named ProdCdTp, and the following rule introduced:

"*All product code set mappings must use ProdCdTp mapping type*"

This way enforces consistency in the mappings and helps locate all the mappings.

After the types and sets are identified, a mapping can be made. A mapping takes one set called a *source* and another set called a *target*. The use of the words source and target is subjective and in each case the user must make a decision based on the user's business knowledge. This mapping is at set level.

The next step is to select values from the source set and link them to values in the target set. In many cases, the codes used are the same and the mapping is straightforward. However, in other cases, different values from different sets might have the same meaning or a close equivalent, so the user, with expertise, must establish a meaningful link.

Not all values in a mapping must be included; it is possible to map one value from either the source or target to many other values in the other set. This way is useful where a source set has a finer grain of codes and a target set has fewer codes, each of which covers a wider subject. For example, a source set with two codes "steel" and "iron" can both map to one code in the target named "metal."

From the data in the sets shown in Figure 3-9, there is a requirement to consolidate the regional product codes to an enterprise master set of product codes.

Figure 3-9 Three sets of product codes, Sales Region 1, Sales Region 2, and a Master product code set

Figure 3-10 shows the result of mapping of Sales Region 1 product codes to the Sales Master set of product codes.

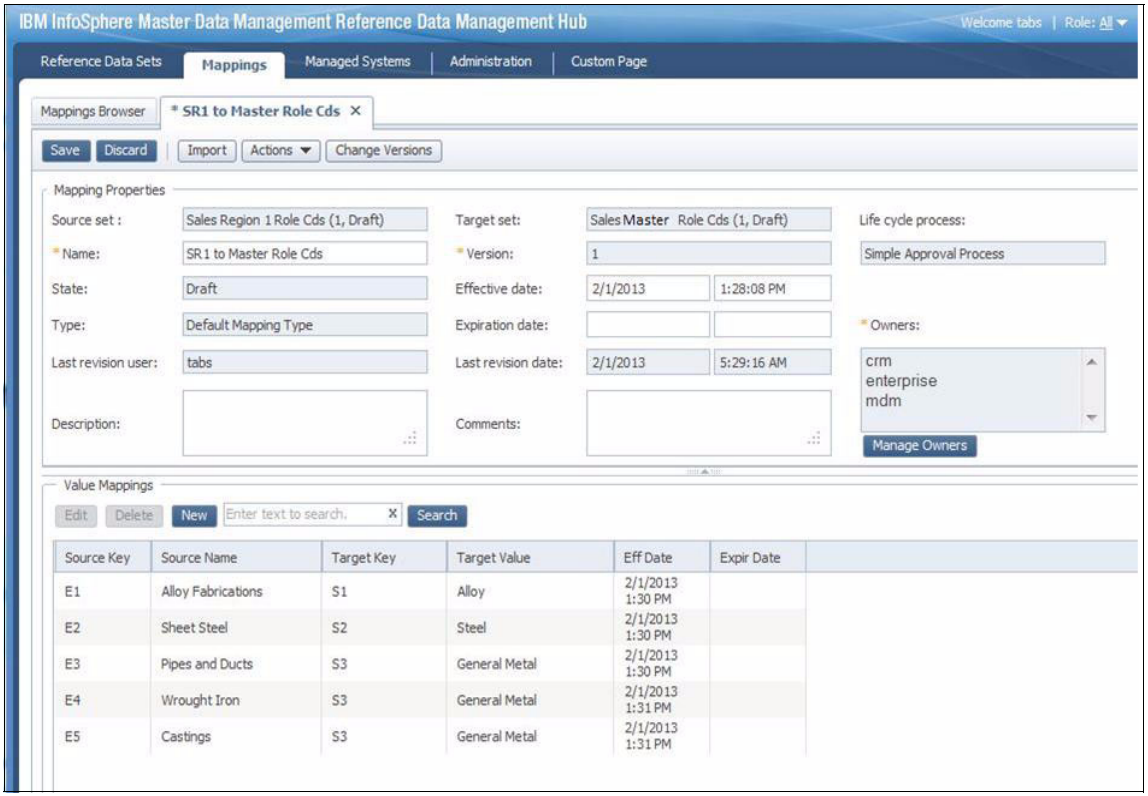


Figure 3-10 The result of mapping Sales Region 1 product codes to the Master set of product codes

3.6 Implementation

Planning for implementation takes in the experience of previous iterative cycles and will enable a straightforward implementation. The commented list here is an example. It does not replace he InfoSphere MDM Ref DM Hub documentation; it is a quick reference of the most important installation and configuration steps taken from an actual implementation.

Installation of infrastructure

The basic installed and tested components are DB2, WebSphere Application Server, and WebSphere MQ. These each require a small adaptation to the local environment file structures and paths.

Installation of InfoSphere MDM Ref DM Hub

InfoSphere MDM Ref DM Hub requires users and groups to be defined. This step can be done by linking to any pre-existing (external) repository such as LDAP or WebSphere Application Server, or through InfoSphere MDM Ref DM Hub. In this instance, the InfoSphere MDM Ref DM Hub the role-based security concept was used, where both activities and stated entities are related to roles that are linked to groups that are defined in WebSphere Application Server. The actual users are members of these groups. The following examples are of standard groups that were used:

- RDMRole_Administrators
- RDMRole_All
- RDMRole_Approvers
- RDMRole_Custom
- RDMRole_Integrators
- RDMRole_Stewards

InfoSphere MDM Ref DM Hub post installation activities

The following activities are done after the installation:

- Adding any customer-specific customizations to InfoSphere MDM Ref DM Hub
- Using scripts to apply additional constraints to the database (mandatory and optional changes, format and length changes)
- Using scripts to create materialized query tables to be used for reporting
- Using scripts to configure notifications arising from subscriptions

Testing and reviewing

Testing is done at all stages of development. Tests that are done in deployment are the final acceptance that the InfoSphere MDM Ref DM Hub is ready for use (it can “go live”). User acceptance testing is not normally repeated here because that process was closed earlier in the project.

For deployment, the following tests are repeated:

- All InfoSphere MDM Ref DM Hub customizations unit tests
- All customized code unit tests
- System end-to-end testing where sets and types and mappings are moved through their full lifecycle
- Integration testing in the user IT environment
- User interface testing

Populating InfoSphere MDM Ref DM Hub

An order of creation must be observed when InfoSphere MDM Ref DM Hub is populated with data.

Figure 3-11shows a summary of the dependencies in InfoSphere MDM Ref DM Hub. Reading from left to right, the sequence is that types must exist before sets or mappings can be created. Sets must exist before they are used in mappings or linked to provide values in other sets.

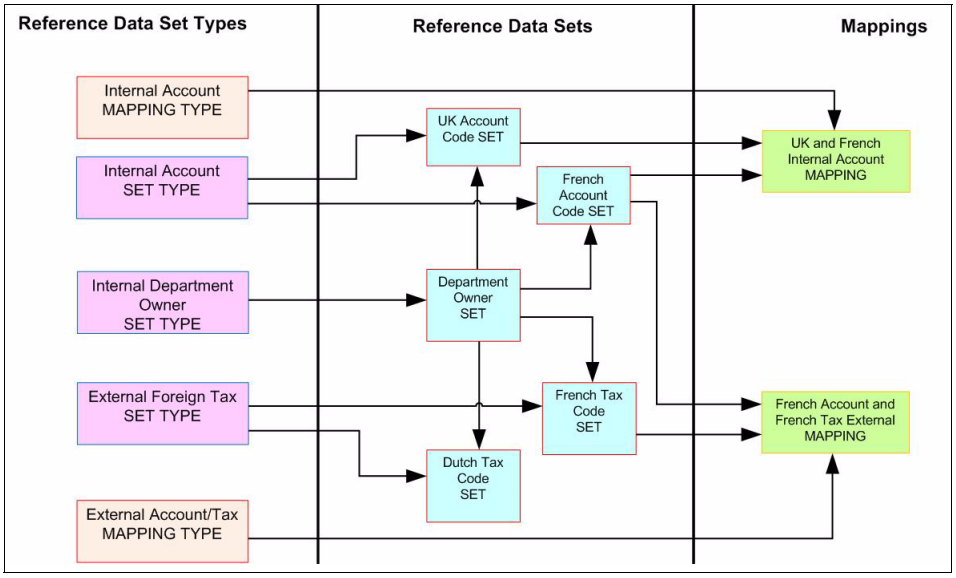


Figure 3-11 Dependencies in RDM

The result of analysis and iterations identifies the actual reference data sets that are to be loaded into InfoSphere MDM Ref DM Hub. Sets can be loaded manually or by using a batch import job. Another possibility is to import sets with their associated types at the same time. In practice, a mix of manual and import routines will be used depending on the complexity of sets. Perhaps the effort of creating an import file is the same or greater than manually entering a set. All import files that were proved in previous iterations can be used at this stage.

Mappings can be built either manually or by importing; the same considerations of complexity apply as for sets.

Other tasks to complete during implementation include distributing reference data sets and mappings manually or by exporting, incorporating any workflow process, and setting up subscriptions.

This description is for planning purposes only. See Chapter 7, “Implementation” on page 159 for a full discussion and information about implementation.

4

Requirement analysis

This chapter describes requirements that must be gathered and analyzed during an IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) implementation. At the end of the analysis, you must be in a position to have a scope of work that the client agrees to, and a clear specification of what must be implemented.

This chapter includes the following topics:

- Discovering reference data
- Data requirements
- Data quality
- Discovering business rules
- Use cases
- Security requirements
- Integration requirements
- Test cases
- Deliverables

4.1 Discovering reference data

As organizations and their business-critical applications grow over time, reference data becomes widely dispersed over many systems. One of the first tasks in identifying reference data is to identify the systems that create the reference data. This topic must be discussed with a client; this topic defines the scope of the assignment. When a system or a group of systems are identified, you can locate the reference data within by using various methods:

- Data model
- Profiling tools
- Existing data

4.1.1 Data model

If the client documented its data assets by creating, maintaining, and using data models and metadata reports, you can use these artifacts to understand where reference data is being used. For example, by analyzing the data model or schema, it is possible to get an idea of the tables that contain lookup values that are being referenced in other tables. Often, lookup tables, such as state codes or industry classifications, reference other grouping or classification tables. These can be identified too through the data model.

4.1.2 Profiling tools

Sometimes, you might not have a data model as a starting point. Even if you do have data models, you do not necessarily understand the quality of the reference data. This is where you can use data discovery or profiling tools, such as Information Analyzer or InfoSphere Discovery, to discover and profile reference data.

As part of discovery, it is also important to understand where reference data is being consumed. The organic growth of applications and systems over a period of time makes this data lineage increasingly difficult for the data stewards of reference data to understand the impact of a change. The identification of reference data lineage, however, is not a trivial task and often requires due diligence in speaking with the consuming application owners and understanding their systems interfaces.

4.1.3 Reference data is already known

Sometimes, organizations have already taken the step of consolidating the reference data in a single system. In such situations, the reference data is identified and documented. Different organizations have different views of what reference data is. Having reference data, master data, and operational data housed in the same “reference data” application is possible. In these cases, a critical task is to “tease” the reference data apart from master data domains such as customer, product, and location. Although storing a reference to master data (for example product code and description) within InfoSphere MDM Ref DM Hub is natural, avoid the storage of any further master data attributes.

After reference data is discovered, the process of analysis begins. Data and process requirements of the client must be analyzed and mapped to InfoSphere MDM Ref DM Hub capabilities. The following sections discuss the requirements that must be analyzed.

4.2 Data requirements

The discovery process results in a catalog of reference data that then must be analyzed to determine how the reference data map to InfoSphere MDM Ref DM Hub objects such as reference sets, mappings, and hierarchies. Identifying patterns of reference sets is important in this process.

As an example, Table 4-1 and Table 4-2 on page 84 list two patterns of reference data. Table 4-1 describes a simple pattern that stores a state code and name.

Table 4-1 Simple reference data defining a code along with a description

State code	Name
AL	Alabama
AZ	Arizona
CA	California
CO	Colorado

Table 4-2 on page 84 describes a pattern which stores some country information such as a name, time zone, and currency.

Table 4-2 Reference data defining a code, its description and attributes

Country code	Name	Time zone	Currency
AF	Afghanistan	UTC+04:30	Afghani
AG	Argentina	UTC-03:00	Argentine peso
BE	Belgium	UTC+01:00	European euro
CA	Canada	UTC-08:00	Canadian dollar

These two patterns must be modeled in InfoSphere MDM Ref DM Hub as different reference data types.

Use the following analysis tasks:

- Map metadata such as primary keys, foreign keys, attributes, and their data types to reference data properties and associated property data types.
- Map metadata such as foreign keys and their data types to reference data custom properties and associated property data types.
- Identify required and optional attributes.
- Capture validation rules.

InfoSphere MDM Ref DM Hub implements validation of attributes by using regular expressions. Therefore, any validation that can be represented as a regular expression can be configured in InfoSphere MDM Ref DM Hub. Table 4-3 lists the metadata that must captured for our example.

Table 4-3 Metadata describing a reference data type

Reference table	Attribute Name	Attribute data type	Primary key[a]	Required	Default value	Validation rule
Country type	Code	String(10)	Y	Y	N/A	Cannot contain numeric values
	Name	String(30)	N	Y	N/A	N/A
	Timezone	String(30)	N	Y	N/A	N/A
	Currency	Reference set	N	Y	N/A	References currency

a. Indicates whether the attribute is a primary key (Y) or is not a primary key (N).

Mappings in InfoSphere MDM Ref DM Hub address the requirement of translating values between different types of reference data. One example of where this translating occurs is when standard classification codes are being

retired and new standards are being created, such as the International Classification of Diseases standards *ICD-9, ICD-10*. Another requirement that the mappings address is when reference values from a source system must be translated into enterprise values represented in a data warehouse.

Often, there are requirements to manage hierarchies. Such requirements need careful analysis to determine how best to model hierarchies in InfoSphere MDM Ref DM Hub. A requirement to model a region, country, and state hierarchy can be modeled in InfoSphere MDM Ref DM Hub by creating Region, Country, and State reference sets so that the State is related to Country, and Country is related to Region. Such hierarchies are referred to as *level-based hierarchies* in InfoSphere MDM Ref DM Hub. Figure 4-1 is an example of a level-based hierarchy.

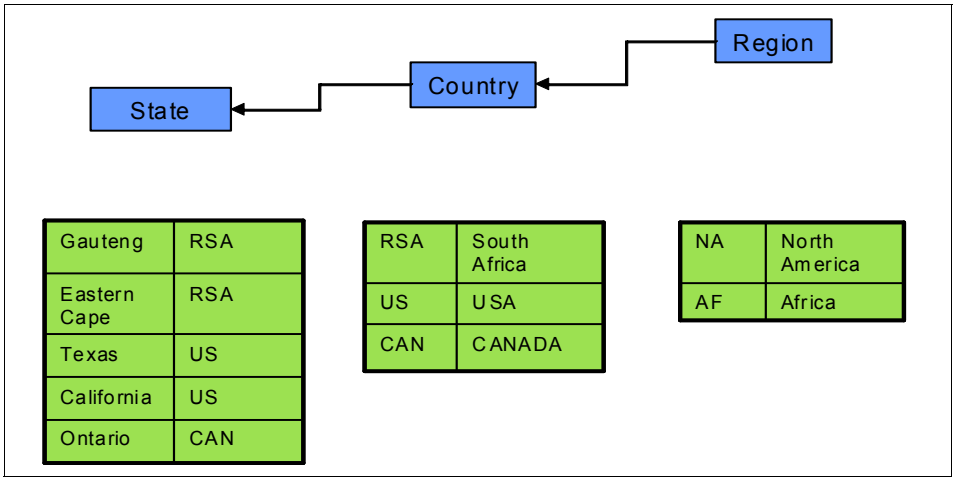


Figure 4-1 Example of a parent child relationship across different reference tables

However, a requirement to model an industry classification structure, which is self-referential in nature, can be modeled by creating a hierarchy object that has parent and child values that belong to the same industry classification reference set.

Figure 4-2 shows a self-referential hierarchy. In this example, both the parent and child codes, and also the relationship between parent and child codes, is defined within the same source reference table.

Code	Name	Parent Code
A	Agriculture, hunting and forestry	
1	Agriculture, hunting and related service activities	A
1.1	Growing of crops; market gardening; horticulture	1
1.11	Growing of cereals and other crops n.e.c.	1.1
1.12	Growing of vegetables, horticultural specialities and nursery products	1.1
1.13	Growing of fruit, nuts, beverage and spice crops	1.1
1.2	Farming of animals	1
1.21	Farming of cattle, dairy farming	1.2
1.22	Farming of sheep, goats, horses, asses, mules and hinnies	1.2
1.23	Farming of swine	1.2
1.24	Farming of poultry	1.2
5	Fishing, fish farming and related service activities	B
5.01	Fishing	5
5.02	Fish farming	5

Figure 4-2 Example of a parent child relationship within the same reference table

If the source reference data contains self-relationships such as the one listed in the figure, these relationships can be imported into InfoSphere MDM Ref DM Hub through the hierarchy object. The hierarchy object is part of the Custom Domain Hub framework upon which InfoSphere MDM Ref DM Hub is built. You can then view the hierarchy as a ragged tree structure by expanding and collapsing the nodes, and also change the parent-child relationships by clicking and dragging the nodes.

Figure 4-3 shows a representation of a ragged hierarchy

Figure 4-3 Representation of a ragged hierarchy

Certain data requirements must be captured from an InfoSphere MDM Ref DM Hub user interface perspective, such as reference set naming standards and reference sets classification. These requirements govern how the reference sets are named and organized in the user interface. Although a classification system helps to organize the reference sets into folders for easy access, an appropriate naming convention with search keywords embedded in the reference set name helps to provide a quick search facility. For example, if a client is implementing a mapping from source values to canonical values, the reference sets that contain the source values might have a naming convention that includes the source system and also the reference set. Therefore, if the enterprise code in question is “Fee Type,” which is from two applications, “System A” and “System B,” naming the sets System A Fee Type and System B Fee Type helps in searching the reference set by using either the source system name or the enterprise code name. As an example consider the following reference sets:

- System A Fee Type
- System B Fee Type
- System A Loan Type
- System B Loan Type

A search by using `System A` as a criteria lists all the reference sets that are sourced from System A, namely “System A Fee Type” and “System A Loan Type.” However, a search having the criteria as `Fee Type` lists the Fee Type

reference set from all sources, namely “System A Fee Type” and “System B Fee Type.”

The data quality reports produced from the discovery phase must be analyzed for completeness and accuracy. If there are requirements for InfoSphere MDM Ref DM Hub to perform validation checks at the attribute level, then the quality of the data must be assessed to determine whether a cleanup of existing source reference data is required before loading. The next section describes the data quality issues to consider before migrating reference data into InfoSphere MDM Ref DM Hub.

4.3 Data quality

Definitions of data quality refer to the tools and processes that enable the creation of correct, complete, and valid data necessary to support good decision-making. Data is good quality if it accurately represents the real-world construct it is intended to. Reference data that is maintained and stored in InfoSphere MDM Ref DM Hub is meant to enforce data quality in applications by the use of code tables. However, there are data quality problems found within the code tables themselves. Quality problems within the code tables themselves can allow errors to propagate throughout systems, applications, and data warehouses.

Problems can be of the following types:

- Completeness within code tables

  Code tables have, as a minimum, the attributes value, description, and start date. More attributes are possible, such as owner, end date, language, lifecycle, and status. Within these few attributes, code tables might have, for example a space character instead of a value, descriptions are missing, or sometimes simply repeat the value; so A1 is described as “A1.” Different date formats are seen in the same table, or in related tables that should be interchangeable.

- Completeness about code tables

  You have incomplete knowledge of where the code table is used in applications, interfaces, and business processes. Lack of knowledge and therefore control of authorship and approval processes is difficult to manage.

- Use of code tables

  Cases are found where the codes are misused in business processes. For example, a country code that is used to define citizenship might classify a person as “Great Britain and Northern Ireland” instead of “British.”

Remedying data quality in code tables is a judgment regarding the extent to which the lack of quality adversely affects enterprise data quality, and detracts from operational efficiency, balanced against how much work needs to be done to detect and improve the quality. By their nature, code tables are small compared to transactional data files, and every line in the file should be different. Because of this data quality issue and because mining tools are of limited use, managing data quality in code tables is still largely a manual effort.

4.4 Discovering business rules

Business rules are additional constraints that are applied to the default behavior of InfoSphere MDM Ref DM Hub. These constraints are normally beyond the product capabilities, but can help customers to better manage reference data. The discovery of business rules plays an important role to help identify custom validations needed in InfoSphere MDM Ref DM Hub during the loading, maintaining, and publishing of reference data.

Consider the following scenarios where custom validations might need to be applied:

- A mapping cannot be approved until the underlying reference sets are also approved.
- A reference value that is expired cannot be used in a mapping.
- When populating a mapping within an automated load process, can a source reference value be mapped to multiple target reference values over time? If yes, what rules must exist to set the effective date of the mapping value?
- During initial load, what is the default lifecycle state that a reference set needs to be set to?
- When extracting or exporting reference sets, are there specific criteria that might prevent a reference set from being exporting? For example. should the reference set be in an approved state to be considered for export.
- Must reference values be validated? For example, should a reference value be only numeric or only alpha? Another example is that the reference value must have a maximum of n digits.
- Are there specific rules to change notification? For example, whenever a reference value changes, a change notification must be sent, and the information in the notification data must contain the values before and after.

These scenarios are examples of where the product might not have the capabilities available initially but can certainly be built into the product.

4.5 Use cases

Development of use cases is critical to understanding how reference data will be managed within the InfoSphere MDM Ref DM Hub user interface. Although the capabilities of the interface are already defined in terms of what it can or cannot do, working with the client and mapping their process requirements to the capabilities within InfoSphere MDM Ref DM Hub is important.

An understanding from a client perspective, of how InfoSphere MDM Ref DM Hub fits into the client business processes, is critical to define what product capabilities can be used. Establish a series of meetings to understand how referenced data will be created, managed, and consumed by the organization. The identified use cases, as a result of these meetings, determine what configuration or customization must be done to achieve the business requirements.

An example might be the way that changes to a reference set post approval might be managed differently by different organizations. In some cases, after a reference is approved, it cannot be changed; the change can happen only by creating a new version of the reference set. In other cases, creating versions might be considered as an overhead and the solution here might be to move the state of the reference set from Approved to Draft. Because there is no one way of implementing a change, these requirements must be captured, and usually result in certain types of configuration to InfoSphere MDM Ref DM Hub. The configuration result in the creation of a new lifecycle process and new states.

Sometimes, the client might make a distinction between types of changes which can occur. For example, creation of a new reference set along with its data might not need to be communicated, whereas, adding new data or changing data on an existing reference set might mandate a notification. In some situations, the type of reference data also matters. Although reference data that is created internally within an organization most certainly requires an approval lifecycle process, this might not be the case for external reference data (such as NAICS, ICD-10) that is subscribed by the organization.

Prototyping workshops provide the opportunity for the implementation team to demonstrate the InfoSphere MDM Ref DM Hub user interface to the client. A good practice is to ask the client for sample data that can be used to drive these requirements. Because you are dealing with reference data, the volumes are quite sizeable and can be loaded within a relatively short period of time into the InfoSphere MDM Ref DM Hub database. After such a prototype is created, the client can more easily articulate requirements such as folder organization, reference set naming standards, and so on. This process also helps the client to understand the functionality of the user interface. During these sessions, gaps between business requirements and InfoSphere MDM Ref DM Hub capabilities

can be identified. Any such gaps must be addressed in the project governance, either through customization or by scoping out the requirement to a future phase.

The outcome is a list of well-defined use cases that clearly articulate the InfoSphere MDM Ref DM Hub capabilities that are required to fulfill the business requirements.

4.6 Security requirements

InfoSphere MDM Reference Data Management Hub provides role-based security control over the reference data sets, mappings, hierarchies, and managed systems. Requirements for security must be viewed from three perspectives:

- User Interface capabilities that are available to a role
- Update operations that are permitted to a role in a specific lifecycle state
- Entity level access through ownership groups

Figure 4-4 lists the user interface capabilities that are provided to the roles, by default. Custom roles can be added in InfoSphere MDM Ref DM Hub. When determining security requirements, the user interface capabilities permitted for these custom roles must be indicated.

UI Tasks	Super	Administrator	Data Steward	Approver	Data Integrator
Maintain Reference Data Set	X		X	X	
Compare Versions	X		X	X	
Import from CSV	X		X		X
Export to CSV	X		X		X
Export to XML	X		X		X
Maintain Mappings	X		X		
Import Mappings	X		X		
Maintain Managed Systems	X	X			X
Import From Managed System	X				X
Export To Managed System	X				X
Maintain Reference Data Types	X	X			

Figure 4-4 Role and functionality matrix

A role can also be granted the create, update, and delete permissions to a reference data set or mapping based on a lifecycle state. For example, a Steward might not be able to update a reference data set while it is in a Pending Approval state; however, an Approver can update the reference set values. Such requirements must also be captured.

Security can also be granted to specific reference data sets or mappings by associating roles to ownership groups. Figure 4-5 associates the default roles to the default ownership groups in InfoSphere MDM Ref DM Hub.

User	User Role	Ownership Group
super, tabs, mdm	All	crm, enterprise, mdm
administrator	Administrator	enterprise
steward	Data Steward	enterprise
steward2	Data Steward	mdm
approver	Approver	enterprise
integrator	Data Integrator	enterprise

Figure 4-5 Role and ownership group association

When defining the security at the ownership group level, understand the following information:

- Users are assigned to user roles.
- Users also assigned to ownership groups.
- Users who belong to an ownership group can only update data sets that are created by a user who belongs to the same group.
- All users are able to view all data sets.

In the configuration in Figure 4-5, the steward user can only modify reference sets that are created by using the enterprise ownership group. The reference sets that are assigned to the mdm ownership group cannot be modified by steward user but can be modified only by a steward2 user. Similarly, steward2 cannot make changes to the reference set that is assigned to the enterprise ownership group.

As part of the requirement-gathering process, the project team should ask the customer to provide information about the ownership groups that are associated with each reference data set or mapping.

4.7 Integration requirements

The InfoSphere MDM Ref DM Hub solution runs in the client IT environment and is administered by IT specialists. Data stewards and business users carry out the authorship and maintenance of the actual reference sets and mappings themselves. For InfoSphere MDM Ref DM Hub to run smoothly, it must integrate with the existing business processes and run in the IT environment meeting non-functional requirements.

4.7.1 Business process integration

The requirements for business process integration are often arrived at or finalized during the analysis stage of an InfoSphere MDM Ref DM Hub implementation instead of traditionally being a predetermined collection of requirements that must be met. The reason is that the functionality of InfoSphere MDM Ref DM Hub has a significant influence on designing the business process. In many cases, analysis shows that code tables are managed in many places and without a general standard. With InfoSphere MDM Ref DM Hub, the introduction of a central user interface and import and export routines enable a rationalization and standardization of a process that was not possible before.

For example, an existing business process uses an external set of industry codes to classify business customers. Updates to the code set are issued twice yearly and might change 30% of 10,000 values, retiring some and bringing in new ones. The staff personnel, who need the codes, concentrate only on the subset they frequently use and manually check on data entry of whether the value is still valid. This process is a candidate for change where the new code set can be updated by a batch import process: either the whole set with a new version, or only the delta incorporated with validity dates updated.

Another example is where a code set is updated in one system and that change is required to be repeated in many other systems. The existing process circulates the change by email or spreadsheet for the administrators of the consuming systems to update their own instance of the code table. This way is inherently error prone and can be helped by setting up a subscription in InfoSphere MDM Ref DM Hub.

The new business process might become the following process:

1. The business user approves automatic updates from one system to many others and then requests IT specialist to implement.
2. IT specialist sets up subscriptions in InfoSphere MDM Ref DM Hub and tests.
3. The new process is reviewed after a period of time.

The documented business processes in most enterprises reveal an emphasis on ensuring the process is documented, mandatory, and meets external compliance and governance standards. Little of most processes deal with the authorship and management of reference data sets, giving an InfoSphere MDM Ref DM Hub implementation an opportunity to add value by supporting new and improved business processes.

4.7.2 System integration

There are published prerequisites for system integration which detail precisely all the expected components that are required, down to the detail of release number and fix pack. These details vary with the environment into which InfoSphere MDM Ref DM Hub is being introduced but a purely illustrative example might be as follows:

- DB2 9.7 Workgroup or Enterprise Edition installed
- WebSphere Application Server 7 with fix pack 15 or later installed
- Web 2.0 feature pack for WebSphere Application Server installed
- WebSphere MQ 7.0.1.3 installed

Verify that these versions are installed. The requirement is that they are proved to run on the client IT infrastructure.

The InfoSphere MDM Ref DM Hub installation itself is made in the WebSphere Application Server and DB2.

If there is a suitable Active Directory in the client environment, a more straightforward approach is to link InfoSphere MDM Ref DM Hub to it, rather than set up InfoSphere MDM Ref DM Hub specific permissions.

4.8 Test cases

As with any system implementation, identification and execution of test cases is necessary to check whether any configuration or customization of the product meets the requested functionality. This section describes the test cases that must be documented and executed in the testing phase of an InfoSphere MDM Ref DM Hub implementation.

4.8.1 Configuration testing

Several aspects of configuration apply to InfoSphere MDM Ref DM Hub. The following list is an example of where configurations occur and therefore must be tested properly:

- Creation of reference data types

  These test cases check whether all the specified reference data types are created properly and that all the attributes are created with correct data types.

- Lifecycle

  These cases test whether the reference set flows through the lifecycle states as preferred. Another aspect of the testing is whether all the required roles have the appropriate permissions for every state in the lifecycle.

- Validations

  Determine whether all the required properties are specified. In cases where property values are being validated, determine whether they are being validated according to the requirements.

- Security

  Determine if all the authorized users have the correct roles and ownership groups.

- User interface

  Be sure the folder structure is set up according to the requirements. Check whether the proper naming convention was adopted, and whether it is possible to search for reference sets and reference values according to the defined search criteria.

4.8.2 System integration testing

Because InfoSphere MDM Ref DM Hub is all about the process of authoring, managing, and distributing reference data, the project test plan must cover some form of integration testing. For example, if a use case involves a source system sending an update of reference data to InfoSphere MDM Ref DM Hub which subsequently gets approved by the data steward and ends with a notification being sent to a consuming application, the interfaces into and out of InfoSphere MDM Ref DM Hub must be tested to ensure that updates and notifications are happening as expected. All interface functionality whether a batch update, notification, or real-time web service call, must have a set of test cases.

4.9 Deliverables

The deliverable of this phase is a requirements document that lists the data and functional requirements of the reference data solution. It is essentially a summary of the topics discussed in this chapter.

The following list is a summary of requirements that must be captured:

- Data requirements
  - List the entities that will be modeled and managed by InfoSphere MDM Ref DM Hub and provide the requirements for each entity. This section might include a mapping document that defines where the source reference data must be stored in the InfoSphere MDM Ref DM Hub data model. For example, a particular source table is to be stored as a reference set and another is to be stored as a mapping.
  - List and define the areas of the InfoSphere MDM Ref DM Hub data model that will be used, that is, reference sets, reference data type definitions, mapping definitions, managed systems, subscriptions, folders, and so on.
- Integration requirements
  - List the requirements about batch loads or web service calls from consuming applications.
  - List the requirements to communicate or notify changes to consuming applications.
  - List the integration requirements with other InfoSphere Information Server products such as Business Glossary.

- Data governance requirements
  - Document the approval lifecycle process for reference sets and mappings.
  - Document any requirements to integrate InfoSphere MDM Ref DM Hub within an external workflow process. Although external to InfoSphere MDM Ref DM Hub, such requirements can be used to develop and direct the use cases regarding data governance.
  - Capture and list business rules for validating or standardization of data.
  - Capture and list any reporting requirements necessary to monitor data governance metrics.
- Security requirements
  - List the users, roles, ownership groups.
  - List the user to role mappings, and user to ownership group mappings.
- Use case specifications
  - Document the set of interactions between external actors and the InfoSphere MDM Ref DM Hub user interface.
  - Identify and detail the use cases required from a reference data governance perspective.

5

InfoSphere RDM model design

The purpose of this chapter is explore the InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) model in more detail and show how some of the concepts work. First, several of the more complex capabilities within the InfoSphere MDM Ref DM Hub are examined, with descriptions of how the features work.

Next, how to use the features within InfoSphere MDM Ref DM Hub to handle various data modeling issues with mapping reference data to InfoSphere MDM Ref DM Hub is described. Also, the various approaches are explained along with the trade-offs between them.

5.1 InfoSphere MDM Ref DM Hub model

Underlying InfoSphere MDM Ref DM Hub is a data model that gives the structure to support the storage of reference data in a way that can be controlled and consistent. A rich set of services allow the data to be maintained, and a user interface (UI) provides ready access to managing reference data.

Figure 5-1 shows a simplified diagram of several major entities of the RDM data model. The managed entities are described in the following sections.

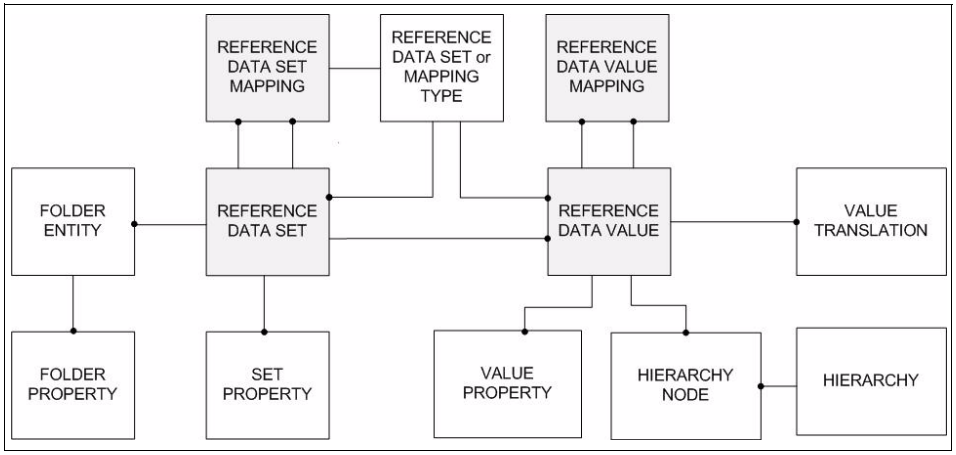


Figure 5-1 Simplified diagram of some of the major RDM data model entities

5.1.1 Managed entities

Managed entities are reference data sets, the values within data sets and set mappings. These artifacts are proactively managed for authoring and administration in InfoSphere MDM Ref DM Hub. The principal means of administering managed entities are through the use of versioning, lifecycle, and ownership functions. Other parts of the application, more client-specific, but still relevant to managed entities include the chosen folder structure, the naming conventions that are used, and the added custom properties.

Managed entities allow different versions to exist and the entities have a governance process around which versions are to be used for a given purpose at any given time. Typically, in an operational InfoSphere Master Data Management (MDM) system, there is only one copy of an entity. There might be historical copies that are used for auditing purposes, but there is only one active copy of the entity. In InfoSphere MDM Ref DM Hub, there can be many versions of an entity and more than one can be active at the same time. The business decides

which version is active. For example, a catalog list of fashion products might have Spring and Summer versions both active in May and June.

In an authoring environment, there is a need to know what the current version is while you might have one or more future versions waiting to become active or being working on by data stewards. In InfoSphere MDM Ref DM Hub, entities that are managed have certain properties and attributes that allow for both holding multiple copies and facilitating the data governance process.

There are two main entities in InfoSphere MDM Ref DM Hub that are subject to versioning: reference data sets and reference data mappings. These managed entities have several child entities that are also versioned along with the parent entity.

Versioning and lifecycle ownership

Versioning support allows for multiple copies of an entity. Versioning is modeled on top of the MDM base entity support. Attributes are added to store version label and lifecycle state. Versions of an entity are copies of the entity. The MDM framework assigns a unique primary key identifier (pkID) to every instance of entity. Because versions are entity instances, each has its own pkID. Also, a common identifier is added; it is called a baseID. The baseID is the same in all versions of an entity. You use the pkID to access a specific version or use baseID to refer to all versions of the entity.

Ownership allows the system to store a list of user groups that are allowed to work with entities. The owner field is a list of group names. When accessing a managed entity, the user registry is checked to determine whether the user requesting the access is a member of one of the ownership groups. InfoSphere MDM Ref DM Hub supports the same user registries that WebSphere Application Server does (local and LDAP based).

The user registry is then checked to determine whether the user is a member of these groups. If not, the user is given read-only access. A `group.properties` file configures the list of group names that are allowed to be owners in InfoSphere MDM Ref DM Hub. This list is a subset of all the groups available in your registry system.

Lifecycle adds the ability to apply a governance model for how versions are handled. It identifies when a version is approved and active, when it is in a draft state, pending approval, or in testing states. InfoSphere MDM Ref DM Hub includes sample state machines (Simple Approval, Two Level Approval, Active Editable, and so on). The built-in state machines can be modified or new sate machines added. Each managed entity is assigned a state machine when it is created. The entity is created in the default or draft state specified by the state machine configuration.

Lifecycle actions are then applied to the entity to move it from one state to the next. Security rules control which lifecycle actions can be performed by each user. This allows users who can edit entities in a draft state to perform the request approval action, but then not be allowed to perform the approval action. Approvals require a user in the Approval role.

Lifecycle attributes that are added to managed entities consist of a state machine ID that identifies which sate machine is being used. It is the state machine which holds the set of states and their transitions that apply to an entity. Each state machine reflects the governance model being applied to the entity.

Children of managed entities

The managed entities can contain child entities (versions and copies of that set) that also need to be managed. For example sets contain reference data values. When a new version of the set is created, the values must be copied from the original set to the new version of the set. Contained children of managed entities do not necessarily inherit all of the same attributes, because children can be modified independently. Some of the management concepts, such as version, owner, and lifecycle are always inherited from the parent-managed entity.

These contained entities must be referenced in a version-independent way. To do this reference, entities that are versions of the same set also have a common baseID attribute. For example, the baseID in values is the same for all values that are the same but in different versions of the set. The pkID of these values differs in each version.

By using the pkID, you can refer to a specific entity. By using the baseID, you can refer to all versions of that entity. In InfoSphere MDM Ref DM Hub, you see both of these IDs used in different ways.

The parent Set of Values refers to the pkID of the set. A value can exist only within a single version of the set. For relationships between values, the baseID is used. If Texas is in the United States, then we want the baseID to refer to all versions of United States.

Figure 5-2 on page 103 shows two versions of a Country Set. The Abstract Country is not persisted in InfoSphere MDM Ref DM Hub. Each version of the country set has the version that is specified by pkID, and baseID. The baseID is the same for both versions. The country set contains two values in each version.

The values contain their own pkID and a baseID. Again, the baseID is the same for different versions of the same value.

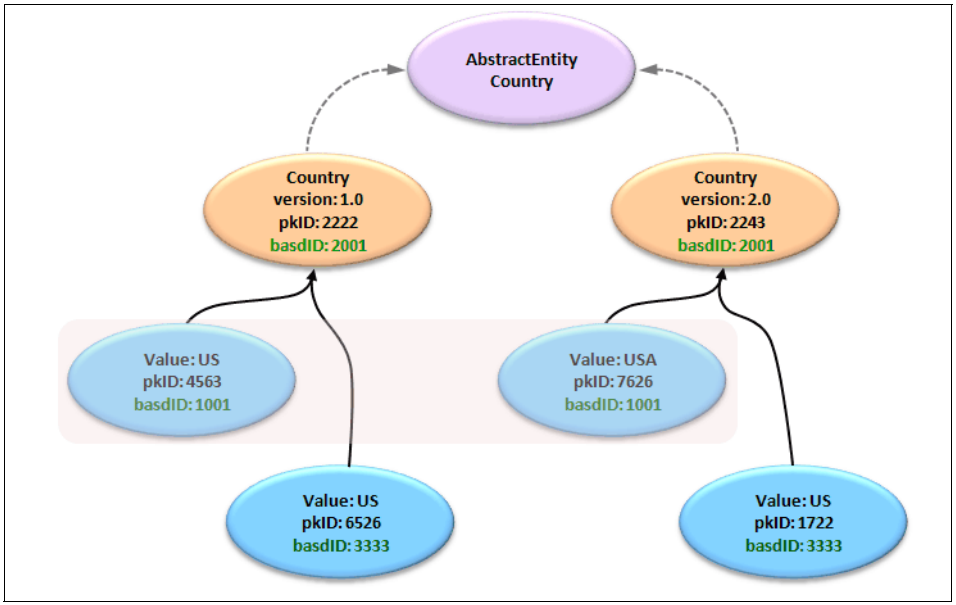


Figure 5-2 Two version sets with different value

Business keys and compound keys

InfoSphere MDM Ref DM Hub supports having a business key for values. The business keys represent external unique identifier for the values. By default, the code field from any reference set is used as the business key for values. The system ensures that there are not multiple values with the same key. There is also support for compound keys. Optionally, up to four additional attributes can be selected as part of the key of a value.

Business keys are configured in the reference data set type. There is an option to disable business keys in the data set type definition but this not recommended, and this feature may be removed in a future release. When adding more properties to the data type, the choice is whether to make the property part of the key or not. Making a property a part of the key will also make it a mandatory field. When importing data, the business keys are used to find existing values and to update the existing values, if the business keys are not found (they are new in the import), new values are created. Business keys are also used to find values when establishing relationships between values for hierarchies and mappings. When importing data, the business keys are used to find existing values and will update the existing values, otherwise, new values are created. Business keys are

also used to find values when establishing relationships between values for set properties, hierarchies, and mappings.

Hierarchy

Two types of hierarchies can be created: hierarchy over the values of a set; level-base hierarchy. See Chapter 3, “Planning a RDM project” on page 53 for more detail.

Hierarchy over the values of a set

Within one set, any number of hierarchies can be built, with no limit on the number of levels. These hierarchies may contain all or some of the values in the set.

Level-base hierarchy

One set can be linked to a field in another set. For example, you have a set of Towns, within which there is a field for adding Province. That field can be linked to another set where all provinces are held, and only provinces from that set can be chosen to appear in the Towns set.

Mapping

Any reference data set can be compared to any other reference data set in a mapping. It is a business decision as to whether there is any benefit in creating a mapping. For example, two sets of product codes might exist in two established systems, both of which actually have the same meaning and must be combined in a data warehouse. A mapping joins two sets and each of the values within those sets can be mapped to an equivalent value in the other set. Not all values must be mapped, and multiple values from one set can be mapped to the same value in the other set.

5.1.2 Types

InfoSphere MDM Ref DM Hub contains types so you can customize your reference data. There are two types: one for sets and another for mappings. The types allow you to configure options for how your sets and mappings behave.

Custom properties

One of the main reasons to use types is to allow custom properties to be defined for sets and mappings. InfoSphere MDM Ref DM Hub has common attributes on both sets and mappings. However, for various kinds reference data, you might need to define and capture different properties about the reference data. When needed, additional properties can be created for reference data sets and mappings.

The supported data types for custom properties are as follows:

- String: Any character data. The contents can be constrained by applying a regular expression to validate the content.
- Text: A multiple-line list of characters.
- Integer: A valid integer.
- Date: A valid date, containing day, month and year.
- TimeStamp: A two-part value that consists of a valid date and valid time of day.
- Reference data set: A relationship to another reference data set. The range of values taken by this relationship is decided by the reference data set that it points to, such as a relation called hasState from a City set to a State set.
- Boolean: A value representing a true or false state.
- URL: A valid URL string.

Validation rules

Certain fields in InfoSphere MDM Ref DM Hub can contain validation rules. Regular expressions are used to limit what data can be entered in the fields. Validations are available for the code field and for string type custom properties.

You can use the following example rule for code to limit the field to two uppercase characters:

```
[A-Z]{2}
```

5.1.3 Ancillary entities: Format

When exporting sets, InfoSphere MDM Ref DM Hub allows you to define a format for how you want to export the data. The format contains which subset of attributes you want to include, what order to include them, and if you want to change the name to include in the export. You can use the format one time during the export process or format can be persisted for future exports.

Formats are associated to the set and not a set version, so a format saved against one version of a set is available for all versions of the set.

5.2 InfoSphere MDM Ref DM Hub model design considerations

When working with InfoSphere MDM Ref DM Hub, you must make design decisions about how to model your reference data. Within InfoSphere MDM Ref DM Hub, you can define sets, mappings, and set hierarchies.

5.2.1 Versioning and implicit versus explicit relationships

There are two ways (implicit and explicit) to establish relationships between various kinds of reference data value properties and reference data mappings. Through the use of types and properties, relationships can be defined on values that link the values in one set to a value in another set. These are one to many relationships. One value can be linked to by many other values in another set. For example, if you have a City set, you can define a property to define what state the city is in: Austin is in the state of Texas. Relationship properties are part of the set containing the property. When you import the set, you can also import the relationship as part of the set.

However, reference data mappings are maintained independently as managed entities. Reference data mappings provide a way for you to define relationships across reference data sets and set values. In addition to the managed entity properties, a mapping has other properties to represent source and target sets.

After a mapping set is created, selecting a source value from a source set and a target value from a target set can create value mappings. The idea of source and target is a business interpretation of which set you are working “from” and “to,” according to which business process is being considered.

Mappings are imported independently from the source and target sets.

Mappings can also contain custom properties so you can add information that can be helpful in deciding how to use the relationships.

Properties on the mapping can also be relationships to other sets. For example, you might have a mapping relationship to another mapping.

5.2.2 Versioning strategy

A good versioning strategy relies on an understanding of the relationships between separate versions and knowledge of the business processes that determine which version to use under what circumstances.

A common issue with the relationships between versioned data is how to determine which version of the data to use. When there is a change to a Country, which version of the Country does a particular State refer to? Logically all the versions of the State are the same entity, however when you are resolving a relationship you need to know which version to use.

The two ways that relationship target versions are determined in InfoSphere MDM Ref DM Hub are implicit and explicit version selection (Figure 5-3).

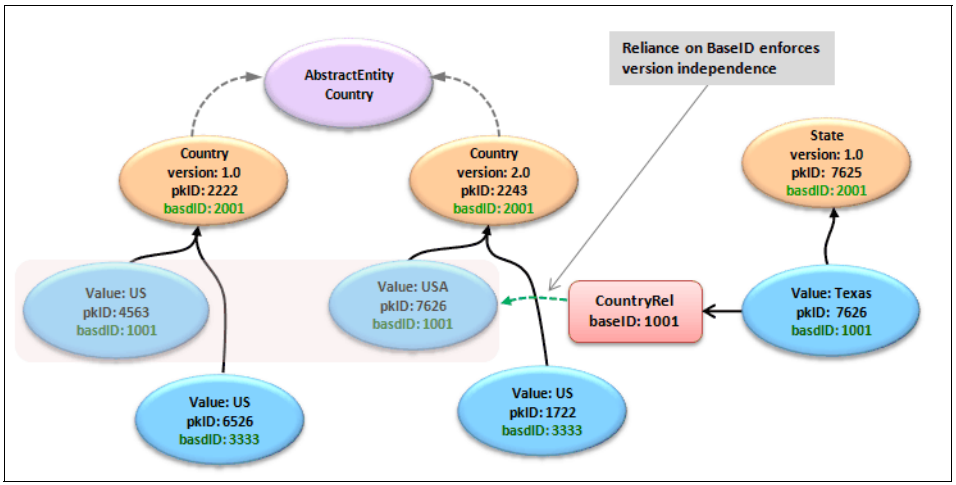


Figure 5-3 Relationships and versioning showing Implicit versus Explicit

Implicit version selection

With implicit version selection, RDM applies a business rule to determine which version of a target set to use. Implicit version selection is a simpler model to manage. Extra data does not have to be stored identifying specific versions. When new versions of the reference data are made, data stewards do not have to update the relationships to use the new versions.

For example, if you create a new version of a Countries set and approve it to be the new active version, with implicit version selection all the relationships to the country automatically work against the new version. Without this kind of a rule, a user must update the relationships to reflect the new version of the data. In the case of a new Countries set version, a change is required to the State set, where the relationship between States and Countries are stored.

The relationship between sets, which use properties, is using implicit version selection. The business rule that is applied is to use the current version. The current version of a set is the one that is in an approved state and is within the applicable start and end dates. If there are multiple approved and active versions, then the most recently modified version is the current version.

Implicit version is also used in the subscriptions. Subscriptions are a relationship between a Managed System and one or more reference data sets. With subscriptions, you want to link to reference data in a particular state such as Approved or Test. To do this step, a lifecycle state attribute is available in the subscription, and which can be used to identify the version that a subscription is for. Updating subscriptions each time there is a new version of a Set is not a task you want to do; InfoSphere MDM Ref DM Hub can be configured to apply this information as a rule. To summarize, in implicit version, the system uses a business rule to compute the version that should be used.

The advantages of using implicit version are as follows:

- Relationships evolve automatically as the target of the relationship goes through its lifecycle.
- Data steward work is simplified and easier to maintain.

The disadvantages of using implicit version are as follows:

- Set must be approved before it can be used in a relationship.
- Determining which version you are looking at without examining all possibilities is difficult.
- Referential integrity rules are needed to prevent inconsistencies.

Explicit version selection

With explicit version selection, a user explicitly specifies which version is used in a relationship. In healthcare, there are mapping rules to map between various versions of the healthcare diagnostic coded ICD 9 and ICD 10. Updates to the standards and the mappings between them are released on a regular basis. These mappings must match the same version of the standard that they are released for and not automatically change to any new version that is created. Explicit version selection allows a user to specify the exact version to be used and does not change unless a user explicitly changes the version.

In RDM, explicit version selection is used in Reference Data Mappings to identify the source set version and the target set version that is used in the mapping. To summarize, in explicit version, the user explicitly specifies which version of the target set should be used.

The advantages of using explicit version are as follows:

- A specific version of a set is used for a relationship.
- Relationships do not automatically evolve to new version of the target entity.

The disadvantages of using explicit version are as follows:

- Extra information is needed in the model to store the select version.
- There is extra work for data stewards to evolve relationships in source entities when changing the target entity. For example if there is a change to the Country set, then all explicit relationships to Country must be changed to reflect the new version.

5.2.3 Versioning mapping relationships

Two mechanisms are available to model a relationship in RDM:

- Value relationship properties are simple and are managed as part of the reference data; they provide one-to-many relationships; they have implicit version selection.
- Mappings are managed independently from the set data to which they refer. Mappings have their own lifecycle, and may be included in many-to-many relationships; they have their own properties on relationships and explicit version selection.

Based on your needs you can choose one model or a mix of both.

5.2.4 Version strategy state machines

With InfoSphere MDM Ref DM Hub, although you can create multiple versions of a set or mapping, you might not want to. In some cases, where the reference data is managed outside of the InfoSphere MDM Ref DM Hub, you might want to keep only a single version that contains the current reference data from the external source. InfoSphere MDM Ref DM Hub lifecycle governance models provide a way to have a single version for some sets while providing a richer lifecycle for authoring and approvals.

The Active Editable governance model is intented to be used when you want only one version of the reference data. The model allows you to edit it and provides no operations to move it to any other state. The data is active and available and can be updated by data stewards.

The Simple Approval lifecycle model allows different user roles to participate in an authoring process where data can be edited in draft state, and go through an approval step before becoming active. After the version is approved, it becomes read-only.

These processes are fully configurable and you can define additional processes as needed. Although any number of lifecycles or state machines can be defined, during design be aware that you cannot always go backwards (for example, to reopen them for editing or to move them out of retirement).

6

Integration

This chapter describes certain key reference data integration scenarios and various considerations for them, facilitating effective enablement. As important as the underlying integration pieces are, it is equally essential to have a proper understanding of relevant reference data management functionality and features that play a key role in plumbing those pieces together. In that context, this chapter provides an overview of several common import, transcoding, and export features that, with a closely-related permissions model, enable effective loading, semantic alignment, and distribution of reference data. Further, through a series of exemplary integration scenarios with products in the IBM InfoSphere suite and related master data management (MDM) domains, this chapter describes how IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) features can be used and extended to realize robust integration patterns.

6.1 Data loading and import

One of the key steps in any reference data integration scenario is to load data from source systems into the reference data management hub. A varied permutation of use cases and solution alternatives for performing data loads makes it a rather non-trivial step with no clear best answer in many cases. This section describes mechanisms, associated considerations, and trade-offs when loading data and performing imports.

6.1.1 Input transformation

In most cases, the source data tends to differ from the input expectation of a reference data management hub. In terms of expected data format, this difference can be quality, semantics, or any combination. Although quality and semantic differences usually are handled in the analysis phase (see Chapter 4, “Requirement analysis” on page 81), differences in expected format still must be resolved.

InfoSphere MDM Ref DM Hub allows the use of various file formats for doing imports. InfoSphere MDM Ref DM Hub allows character-separated value (CSV) or extensible markup language (XML) file imports. Any source data must be transformed into one of these formats before using any import interface. Figure 6-1 shows an example of an incompatible source with InfoSphere MDM Ref DM Hub. You can write custom adapters to perform such conversions. If the data source is unstructured, the adapter must parse it, induce appropriate structure, and then convert it to the expected character-separated or markup representation. If the data source is already structured but in a different format than what is expected by the InfoSphere MDM Ref DM Hub, the adapter converts it into one of the expected formats after augmenting and modifying it with expected delimitation or markups.

```
PL:Poland, PT:Portugal, SK:Slovak Republic, ES:Spain, SE:Sweden,
CH:Switzerland, TR:Turkey, GB:United Kingdom, US:United States,
SA:Saudi Arabia
```

Figure 6-1 Example source data format incompatible with InfoSphere MDM Ref DM Hub

Some of our clients use custom code to combine data from multiple sources into a single CSV file before performing a bulk load into the InfoSphere MDM Ref DM Hub. This combination is often non-trivial where data from different sources represents different aspects or attributes of the same reference data value. For instance, for importing data into a country code table in InfoSphere MDM Ref DM

Hub, part of the attributes (code, name) might come from a Location entity while description might be part of the Address field within a Person entity.

Figure 6-1 on page 112 demonstrates a file snippet containing reference data from a source system that is not compatible with InfoSphere MDM Ref DM Hub. Figure 6-2 shows the data after performing appropriate input transformation to make it compatible with RDM input expectations, where columns represent reference data value properties and rows represent a reference data value instance.

```
ISO Country Code ,Name
PL,Poland
PT,Portugal
SK,Slovak Republic
ES,Spain
SE,Sweden
CH,Switzerland
TR,Turkey
GB,United Kingdom
US,United States
SA,Saudi Arabia
```

Figure 6-2 Transformed data consumable by InfoSphere MDM Ref DM Hub

6.1.2 Import interfaces

InfoSphere MDM Ref DM Hub provides several ways to import reference data. The most “user-friendly” method is the user interface (UI) import that is mostly a manual process. Other import methods are batch and command-line based mechanisms. By linking the batch importer to a **`cron`** job that is defined on a fixed or variable time-window, entire import process can be bootstrapped. Typically, application programming interfaces are also exposed to achieve fine-tuned control of the import process.

The decision of whether to use a UI import rather than a batch process to perform bulk imports has several implications beyond simplicity of use. For instance, depending on the nature of a specific RDM project in question, letting the development team bulk-load all the reference data initially while establishing the environment and processes might be preferable. One of our early user groups built a bulk load capability that proved to be useful beyond the initial data load because they were able to use the bulk load job for periodic batch update of RDM tables from their back-end systems. However, if the project is constrained by initial setup time, a preferable approach might be to train an initial set of business users to import their own data when the initial environment and governance processes are established. This way can help in getting the initial RDM environment up and running sooner.

Another argument in favor of letting the development team load data (instead of business users) is that the data analysis and modeling for the loaded data can be well-aligned initially. Letting business users load their own data has implications about enforcing suitable governance processes and modeling constraints. Business users who are familiar with mixing up their data in ad-hoc tools (spreadsheets, plaintext, and so on) might not model the data appropriately in InfoSphere MDM Ref DM Hub. For instance, if party address data was maintained in a single spreadsheet earlier, a business user is not likely to separate Country and State reference data, which might be highly desirable in an InfoSphere MDM Ref DM Hub. As a preventive measure, the project management team can decide to closely link modeling with the workflow of reference data entities, however, that would then have implications on the time necessary to process each reference data entity from the draft state through approval.

In some scenarios, a middle path might be more suitable where the development team loads enterprise-level data with broader usage and applicability, while sets of business users load suborganization-wide data later. Overall, there is often no clear data loading mechanism that is best for every scenario. Consequently, giving clients the power to choose is critical to broad-enough applicability of a comprehensive reference data management hub implementation.

User interface import

After the source reference data is properly transformed and represented in a CSV or XML file, it can easily be imported into the RDM hub. For a step-by-step process overview of the user interface import function in InfoSphere MDM Ref DM Hub, see 7.3, “Implementing InfoSphere MDM Ref DM Hub model” on page 172. Here, our focus of discussion is on various implications and considerations with respect to integration scenarios.

Importing code values into reference data sets

A typical integration scenario, such as ETL warehousing, real-time ESB, invariably begins with importing the code values residing in code or lookup tables within source systems into InfoSphere MDM Ref DM Hub. The user interface import functionality can make importing data from external source systems easy; there are considerations when planning the import process:

- InfoSphere MDM Ref DM Hub requires the reference data set versions, into which the data must be imported, to be present before the import process can begin. For this reason, you must define a standard procedure to either use existing set version to load delta changes, or define a new version on each load. Depending on the requirement, you might need to retire the previous version before loading data into a new version. When this process is fixed, the reference data type definition must be linked with the import module to ensure

the import file format is aligned with the properties defined in the reference data type definition. All required properties must be specified in the import file.

- InfoSphere MDM Ref DM Hub user interface offers a wizard-based importer (Figure 6-3). With it, you have the flexibility to select the column name to attribute mapping, file format, separator used in the file, date and time stamp formats for enabling the importer to understand any potential date and time field conflicts, and so on. You must determine these configurations early in the process to streamline the entire import process.

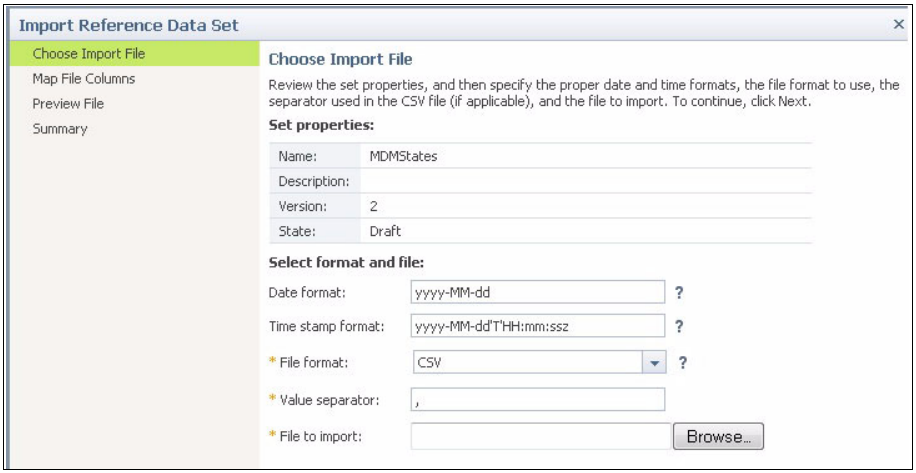


Figure 6-3 Flexible wizard-based import interface of InfoSphere MDM Ref DM Hub

- A common issue in reference data management is the representational gap between the data in source systems and what InfoSphere MDM Ref DM Hub can consume. Input transformation is then required. If using an XML file for importing data, you must ensure that the import file conforms to the published and bundled schema for the kind of reference data entity that is being imported (set values, value mappings, or hierarchies). When building custom adapters for creating XML import files, see the published schemas used in InfoSphere MDM Ref DM Hub at the following information center (by selecting **Reference** → **Data and configuration reference** → **Schemas for XML import**):

  http://pic.dhe.ibm.com/infocenter/mdm/v10r0m0/index.jsp?topic=%2Fcom.ibm.swg.im.mdmhs.rdm.ref.doc%2Ftopics%2FDataReferenceIntro.html

- If the entity is defined to contain a compound key (a key defined using multiple reference data value attributes), the import file must contain values that can map to each part of that compound key. For instance, if a reference data set for countries has a key defined on both code and name properties, then they become required fields during the import process.

- During the import process, any existing records (matched using the key) are updated while new ones are added. After the import finishes, the user can navigate back to the set and inspect the values. For reporting purposes, a summary page is presented after the import process finishes, letting the user know the number of successful imports (additions or updates) and failed cases (exceptions). The error records can be exported in a file and worked upon offline. After correction for any potential mistakes, they can be reimported back using the same procedure.

Defining level-based hierarchies while importing values in sets

The import procedure also allows importing rows for a data set that has a compound key formed by using the values from another set. By doing this step, you can establish relationships between more than one set, and define what are known as level-based hierarchies (Figure 6-4), where each level is defined by a unique reference data set.

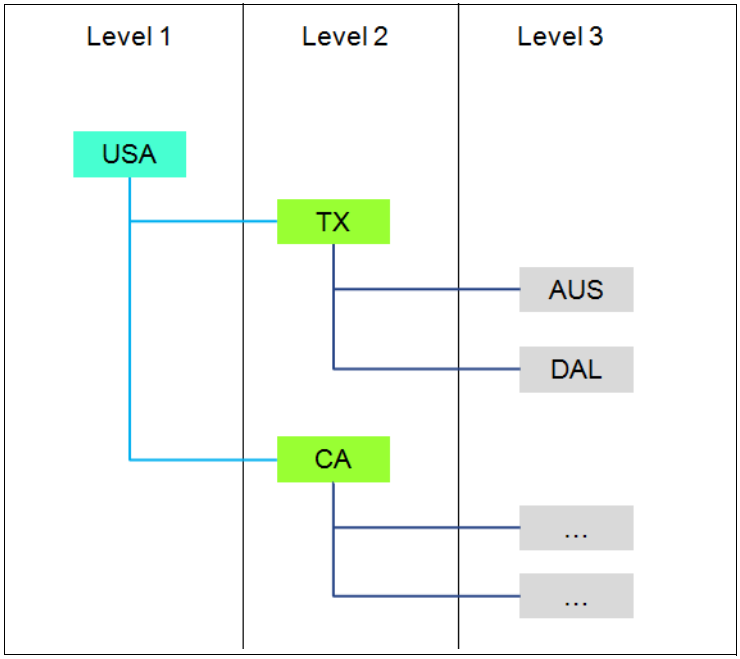


Figure 6-4 Level-based hierarchy example

To import such relationships, you must define the reference relationship to the other set (for example, States at Level 2 to Countries at Level 1 in Figure 6-4). InfoSphere MDM Ref DM Hub import wizard provides flexibility to do this step. For instance, as shown in Figure 6-5 on page 117, while importing reference

data values for a States set, you can decide to refer values from a set of Countries by referencing country codes in the compound keys.

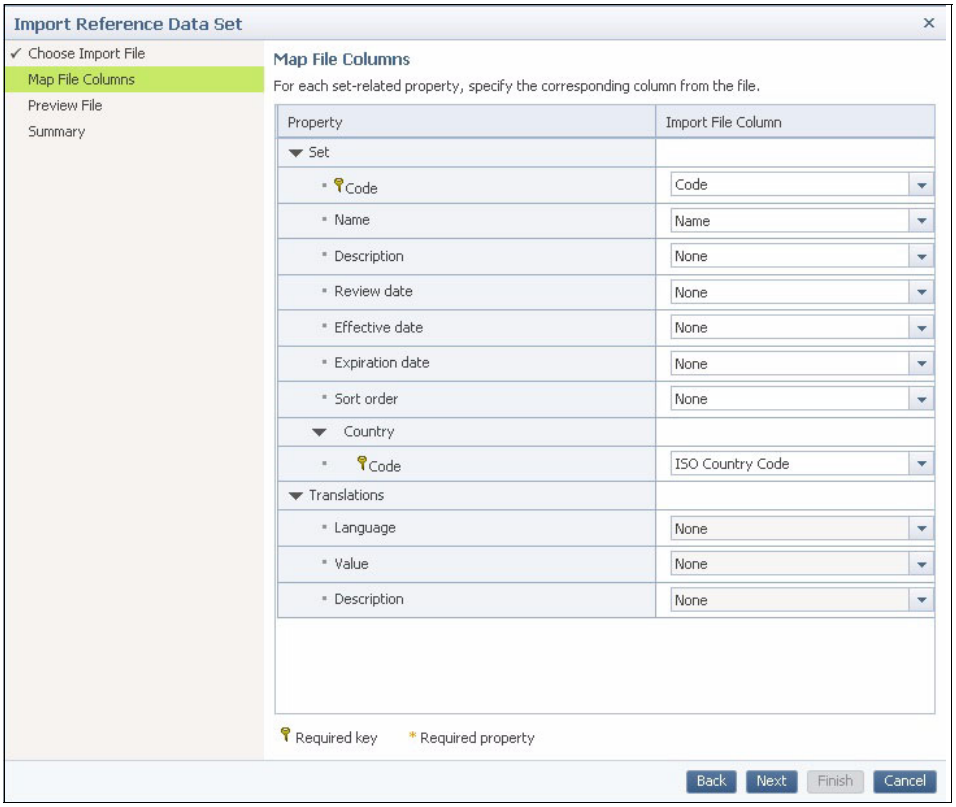


Figure 6-5 Defining level-based hierarchies while importing reference data values

Because level-based hierarchies are defined through relationships across multiple sets, a requirement is that the set being referenced in the set that is being imported must be marked as active. In most cases, being active requires the set version to be marked as being current using effective dating (effective date falling prior to the current data and expiry date set to a future date).

Importing set hierarchies

In addition to importing reference data values into reference data sets, you can import set hierarchies (Figure 6-6) that induce a classification structure over existing values within a single set (compared to level hierarchies that are defined across multiple sets).

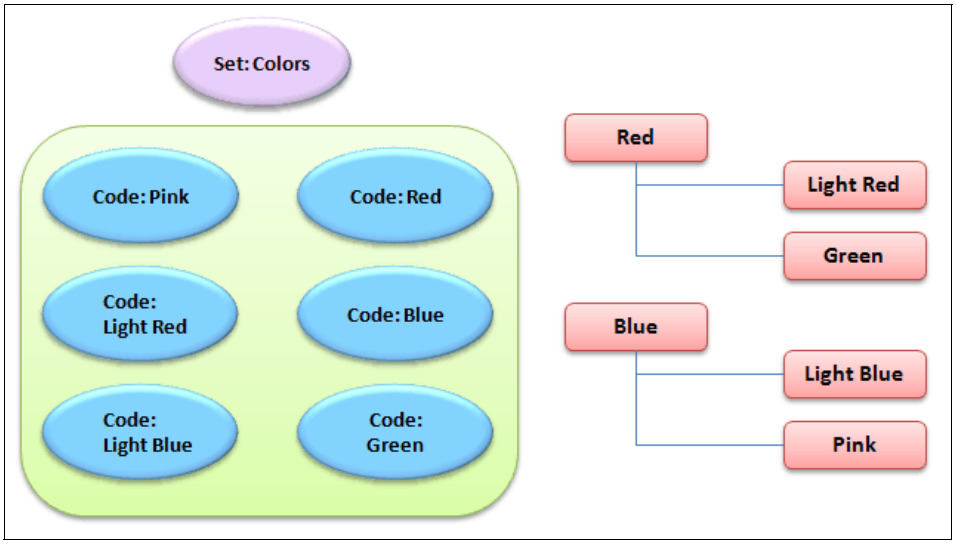


Figure 6-6 Set hierarchies defined on a single set of reference data values

These imports are separate from set imports and require a set with pre-existing values as a prerequisite. Therefore, if you want to bootstrap-import set hierarchies, you must define your import so that it occurs sequentially after the corresponding set import has finished.

We noted earlier in reference data value imports that an exception, which resulted from having an incorrect entry in the import file, can be exported and corrected from the summary page. This step is made easy because each row in the import file is imported independently of any other rows. However, for a set hierarchy, because every row is specifying a new relationship in a fully connected tree, any incorrect entry can potentially affect all the subsequent rows in the import file. For example, the first row in the import file is often used to specify the root. If this row itself is incorrect, all the remaining rows will result in errors because the root of the hierarchy is not defined.

Import using batch processor or service interfaces

A batch processing framework is bundled with InfoSphere MDM Ref DM Hub and can be used to build custom batch jobs to run RDM import transactions and support custom input and output data formats. For more information and examples about configuring, using, and customizing the batch interface, go to the information center:

http://bit.ly/YkMwBc

Select **InfoSphere MDM Server** → **Planning to use InfoSphere MDM Server** → **InfoSphere MDM Server platform technical features** → **Batch processing using MDMBatch**.

Certain reference data management implementations can expose a light-weight Representational State Transfer (REST) wrapper interface on top of the core services, which can ease the labor of determining which exact transactions to invoke for performing data loads through a batch processor.

When loading data in bulk, you must understand the pattern of the ongoing load of changed data from back-end systems. Depending on the load and frequency, loading a delta file that contains only the changes might be easier than loading the entire file for a set. Similarly, for large sets, loading changed data into a new (empty) version rather than an already full version might make more sense, especially, if nearly all the code values have changed.

In addition to imports that require an update to existing reference data or addition of new data, there could be a requirement to remove existing data while performing imports. InfoSphere MDM Ref DM Hub provides a few alternatives to do this step. One is to use effective dating to update the records that no longer need to be active with an expiry date that is before the current date or effective date. Alternatively, you can explicitly add a flag property to the reference data set type, which is set to Active or Inactive. This property then is updated during the import process for data that no longer needs to be active or vice-versa.

In addition to batch processor framework, programmatic approaches to directly call InfoSphere MDM Ref DM Hub services offer a flexible and highly customizable mechanism to load data into the hub. As an example, one form of reference data concerns taxonomies. We use taxonomies for classifying web content as part of Enterprise Web Taxonomy Tooling (TMT). We must coordinate these taxonomies across many content management systems (CMS) in such a manner that we are able to use the same taxonomies across all those CMS. However, those systems have different histories and heritages, and therefore different designs. A few examples of such taxonomies are industries (the industries in which customers conduct their business), geographic locations, and solutions areas. One of the challenges during the import process is to adhere the

input XML feed to a format that can be consumed by InfoSphere MDM Ref DM Hub. The scenario is illustrated in Figure 6-7.

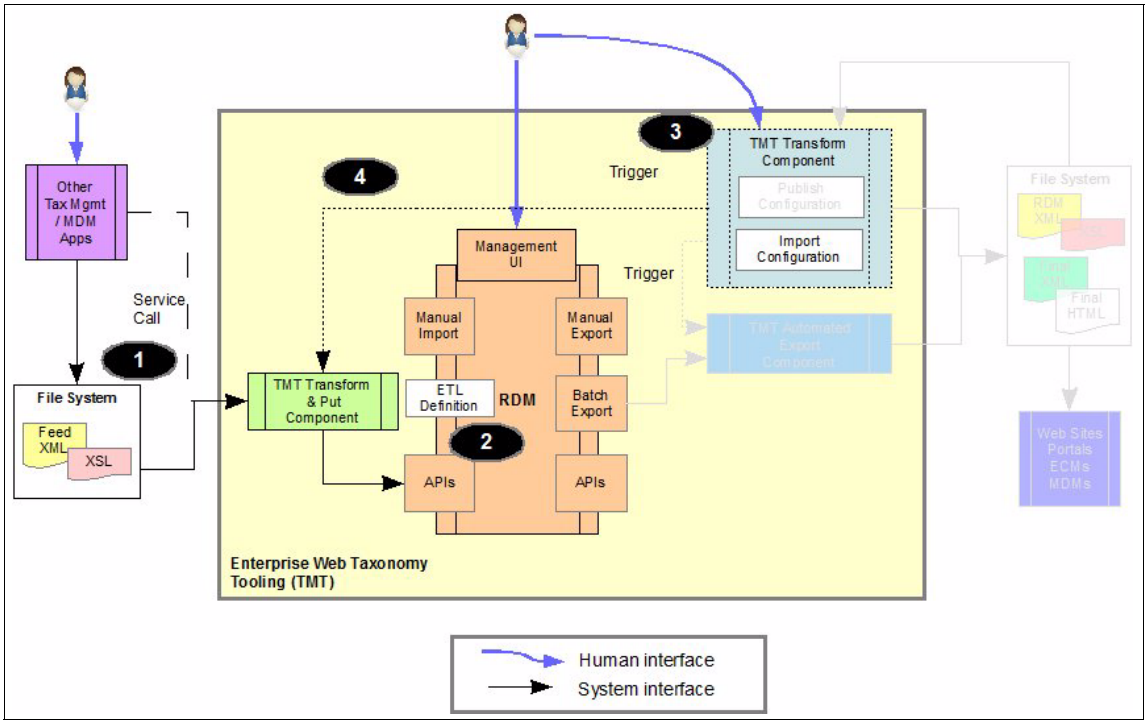


Figure 6-7 Customizing import interface: Enterprise Web Taxonomy Tooling

The TMT Transform and Put component uses configuration information, stored in an ETL definition in RDS, to call for XML over HTTP and apply an XML transform to generate one or more import files in the RDM-consumable format. It then uses the RDM service APIs to load the data that is contained in the import files into appropriate RDM entities. The Import Configuration in the TMT Transform Component simply passes the name of the ETL configuration to use when importing, and the source file to use when getting the information. In this way, the transform can be run against new versions of the source data without having to create a whole new ETL definition in RDM. Overall, the process of setting up a new import in the TMT solution is as follows:

1. Find the source and determine the method to be used to consume the reference data.
2. Configure the import:
   a. Create an XSL that build the required import file.
   b. Create a new ETL definition in the ETL reference data set in RDM.

3. Create a new import configuration document in the TMT Transform component.
4. Trigger the import.

This example scenario demonstrates how the various InfoSphere MDM Ref DM Hub import mechanisms can be used to achieve the level of customization you want according to the integration scenario under consideration.

6.2 Data distribution and export

Exporting data out of InfoSphere MDM Ref DM Hub for distribution to target systems and consumers is another important step in an integration scenario. Without flexible mechanisms and alternatives for both performing export and also allowing for easy transformation of exported data into a representation that can be consumed by a downstream system, the ability to enable seamless integrations can be severely limited. This section describes considerations and challenges involved in the export and data distribution procedure. It also presents available mechanisms in InfoSphere MDM Ref DM Hub for overcoming these issues.

6.2.1 Downstream system challenges consuming reference data

InfoSphere MDM Ref DM Hub provides several methods to export reference data sets and hierarchies for immediate use. The most common one is the user interface based manual export that requires no programming experience. Alternatives, such as batch export or using service interfaces and APIs, require basic programming knowledge. InfoSphere MDM Ref DM Hub also provides utilities to define and export reference data mappings that are used for transcoding.

Similar to data import, during data export, you face the issue of representational differences between the default export format that a typical reference data management hub implementation provides and one that is expected by target consumers of the exported data. In rare cases, the target system can directly consume exports from the product. However, in most scenarios, a transformation is required. In environments where multiple downstream systems must consume reference data from a typical reference data hub, it is impractical to assume that the format in which reference data is being exported exactly matches the format that is used by all those systems. This behavior is true whether you use manual or automated export processes.

In certain scenarios, suitable assembly of export information is required. For example, in one of our internal use cases, combining the reference data set and set hierarchy exports is required. Because the InfoSphere MDM Ref DM Hub functions that are for immediate use are based on separate schemas and exports for reference data sets and set-based hierarchies, special adapters had to be written for the scenario to work.

Similarly, if the export is from a newly created version of a reference data set to which a downstream system is subscribed, be sure to understand the extent of these changes to understand whether there is a need to consume them differently. In case of multiple consumers, there might be a need to batch and distribute multiple transformations, a subset of which might be equivalent, while others might be distinct, catering to a different subset of downstream systems.

Delta files

If the export is incremental, you might want to obtain a delta export for the changes since the last export for certain downstream systems rather than the entire set of reference data values. For enabling this approach, a reference data hub implementation must compare the current version of the reference data set with the version that was last exported, and allow export of only the changes. The output transformer can then only consume this delta file and transform it in a series of add or update jobs that can be directly run on the downstream systems.

for all these reasons, you must carefully consider and plan transformation of reference data exports into formats that can be consumed by downstream systems. In addition, pay close attention to situations where special assembly of output is wanted. Several of these considerations are described in 6.2.3, “Output transformation” on page 124 after a discussion of the various considerations that are involved in using the export interfaces that are available in InfoSphere MDM Ref DM Hub. See the step-by-step export scenarios with InfoSphere MDM Ref DM Hub in 7.3, “Implementing InfoSphere MDM Ref DM Hub model” on page 172.

6.2.2 Export interfaces

Similar to the import interfaces, reference data residing in InfoSphere MDM Ref DM Hub can be extracted in CSV or XML format by using the user interface export and batch export mechanisms. This exported data can be consumed by an adapter or transformer to output a format consumable by target systems. If using programmatic approach, you can code the transformer or adapter to call the services directly to eliminate one level of serialization or deserialization to or from a file (CSV or XML).

User interface export

When using user interface export, several considerations are required to ensure that the exported file is as close to the target system expectation as possible to minimize the transformation overhead.

Similar to the user interface import function, user interface export gives you the flexibility to pick the correct format for the target systems and to customize the amount of information you want exported. For instance, if the export is part of a continuous update process, you can skip some of the set and value properties if they are redundant across continuous updates.

The user interface export wizard also provides the flexibility to conform the export to the target system expectation. Using the Edit Properties to Export page (Figure 6-8), you can edit the names of the properties to decide how the data appears in the export file. This feature is useful in reducing the burden on the transformation stage later. You also have the option to save the format as a reusable template if data in the same export format will be distributed to the target systems periodically.

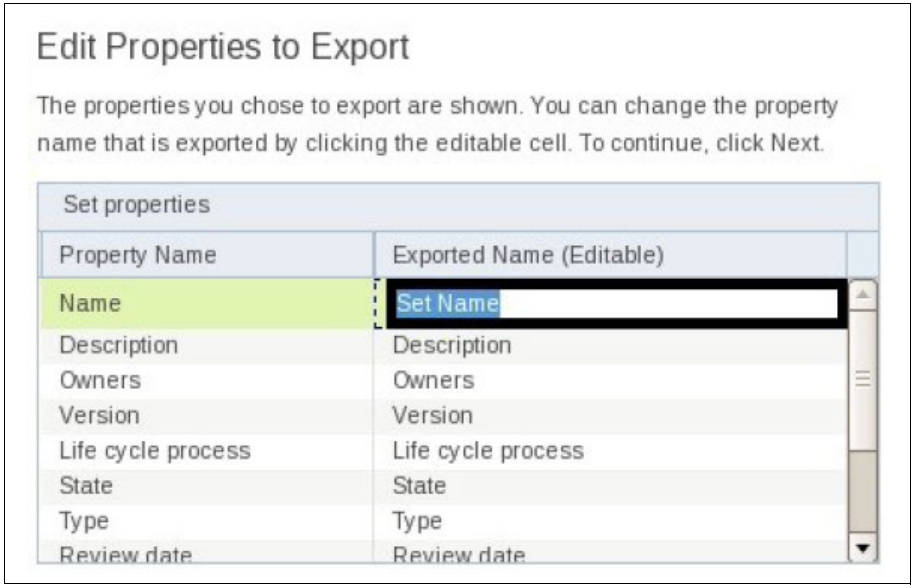


Figure 6-8 Flexible wizard-based export interface in InfoSphere MDM Ref DM Hub

Batch export and services interface

Batch export is a command-line API that allows you to export reference data from the InfoSphere MDM Ref DM Hub. The API is included with the product. It allows you to choose between XML or CSV formats with a wide choice of character delimiters. Depending on whether you want a reference data set, hierarchy, or

mapping export, you can set an appropriate configuration. You have the flexibility to provide the state and version of the specific reference data entity that must be exported.

Batch export is based on a J2SE client that is able to talk to RDM web services programatically. You can use the same client to programatically invoke RDM export services in the same way as the batch export does. With batch export, you can bootstrap the whole export process programatically instead of going through RDM's bundled batch export application.

6.2.3 Output transformation

Transforming the output from a reference data management hub is necessary in many situations because the challenges in consuming reference data in downstream systems (described in 6.2.1, "Downstream system challenges consuming reference data" on page 121).

Similarly to the import transformation process, the output transformers also must accept the export as input, process it to perform structural and semantic changes to finally result in something that is consumable by downstream systems. Revisiting the example scenario on Enterprise Web Taxonomy Tooling (TMT) from "Import using batch processor or service interfaces" on page 119, another challenge here is that some of these taxonomies are handled differently across these systems. That is, although several or all systems might be able to ingest an XML file, the format of that XML file is not the same in all those systems.

To get across this problem, we adopted the approach to export the reference data from InfoSphere MDM Ref DM Hub using the automated batch export capability. We produce an XML file in a standard format that mimics the structure of the data inside InfoSphere MDM Ref DM Hub. We then apply a standard transformation using XSL to produce the desired XML output files. This XML file has a structure that allows consuming applications to find similar information with the same tags across all taxonomies (reference data sets). In particular, because properties on the taxonomy nodes (reference data values) are important to our consuming applications and because the properties are quite different from taxonomy to taxonomy (reference data set to reference data set), we present those properties consistently no matter which taxonomy (reference data set) is being used. In this way, the consuming applications can more easily use these properties.

Figure 6-9 illustrates the components that were added to InfoSphere MDM Ref DM Hub to enable the taxonomy administrator to manage repeated publishing of both simple and complex transforms.

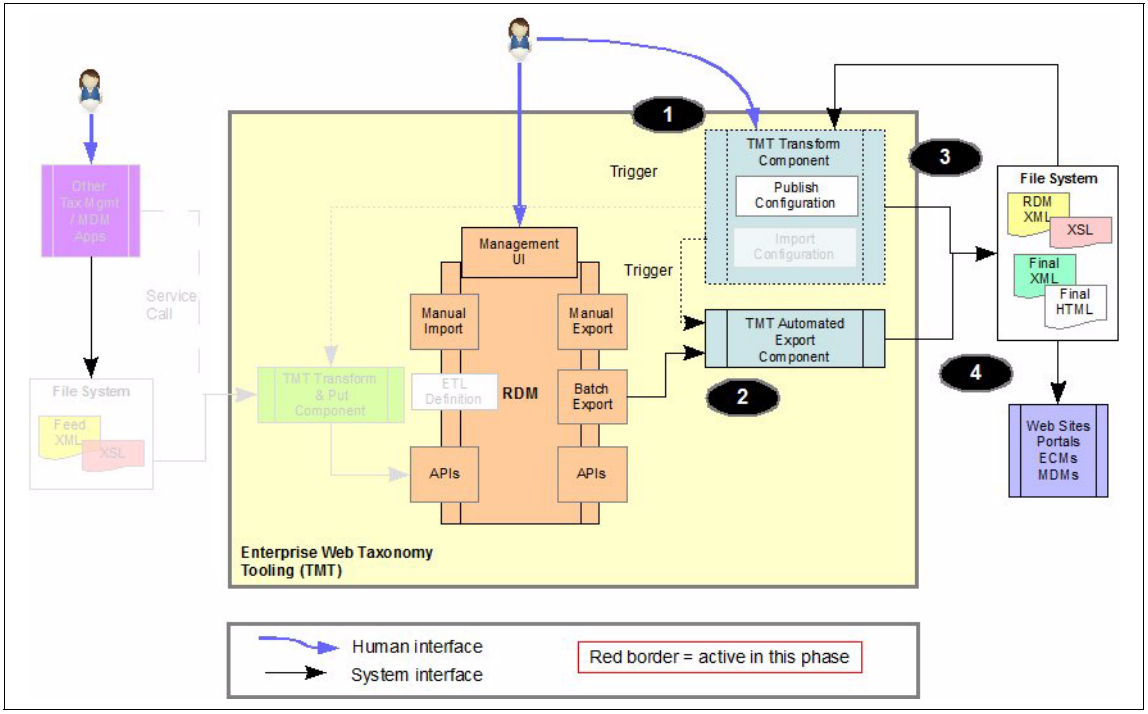


Figure 6-9 Customizing export interface: Enterprise Web Taxonomy Tooling

The key components are as follows:

- The TMT Transform component provides a UI and repository to capture the information needed to assemble a taxonomy.

  A Publishing Configuration document contains information about the source InfoSphere MDM Ref DM Hub objects that are required for the transform, a URI pointing to the XSL used for the transform, and the target location for the output. It provides the administrator with a way to preview the input data and the output result. The document also provides a way to trigger the export of InfoSphere MDM Ref DM Hub objects.

- The TMT Automated Export component provides a simple web interface to the InfoSphere MDM Ref DM Hub batch export jobs.

  The information that is required to configure the export is passed from the TMT Transform component. The taxonomy administrator then runs the batch job and gets log information back in a web browser.

- A corporate-wide file system is used to host the InfoSphere MDM Ref DM Hub XML file that is exported, the XSLs needed for the transform, and the final output files.

Assuming all the InfoSphere MDM Ref DM Hub objects are configured and that a standard transform (XSL) is available, the taxonomy administrator can take the following actions to create output:

1. Create the publishing configuration for a given input and output.
2. Trigger one or more exports and verify that the InfoSphere MDM Ref DM Hub XML was created.
3. Trigger the publishing transform and verify the results.

The taxonomy can then be consumed by multiple systems. Owners and users of the taxonomies can also use the reports created by the transforms.

Figure 6-10 shows an example of how a set of source taxonomies are transformed into a set of output taxonomies used to feed a search engine. The needed InfoSphere MDM Ref DM Hub object XML is exported, transformed, and published for reuse.

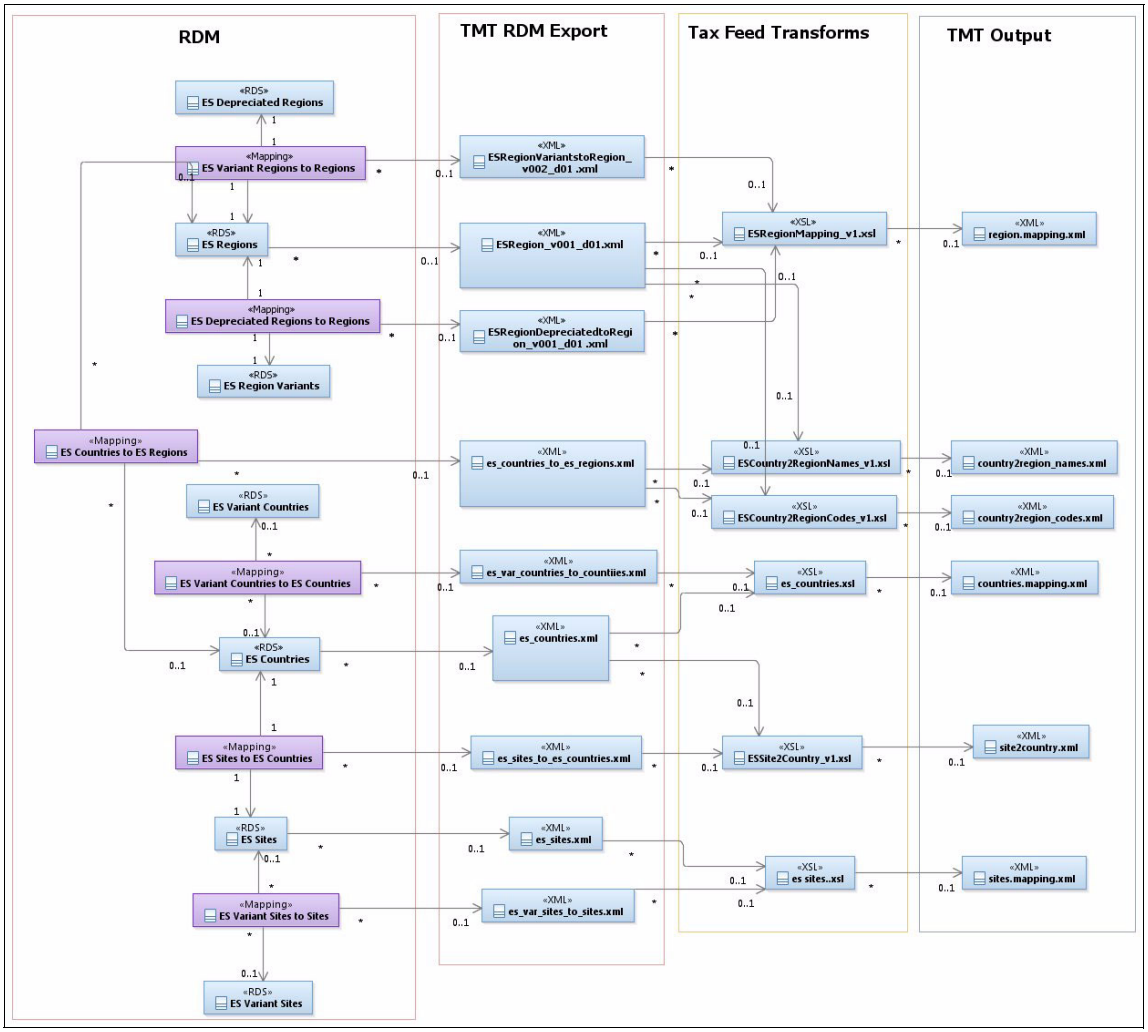


Figure 6-10 Conversion of source taxonomies into target output

Figure 6-11 through Figure 6-17 on page 134 are examples of the output of the InfoSphere MDM Ref DM Hub export for a reference data set and a mapping, the XSL that is used to create one of the output formats, and two examples of the output files.

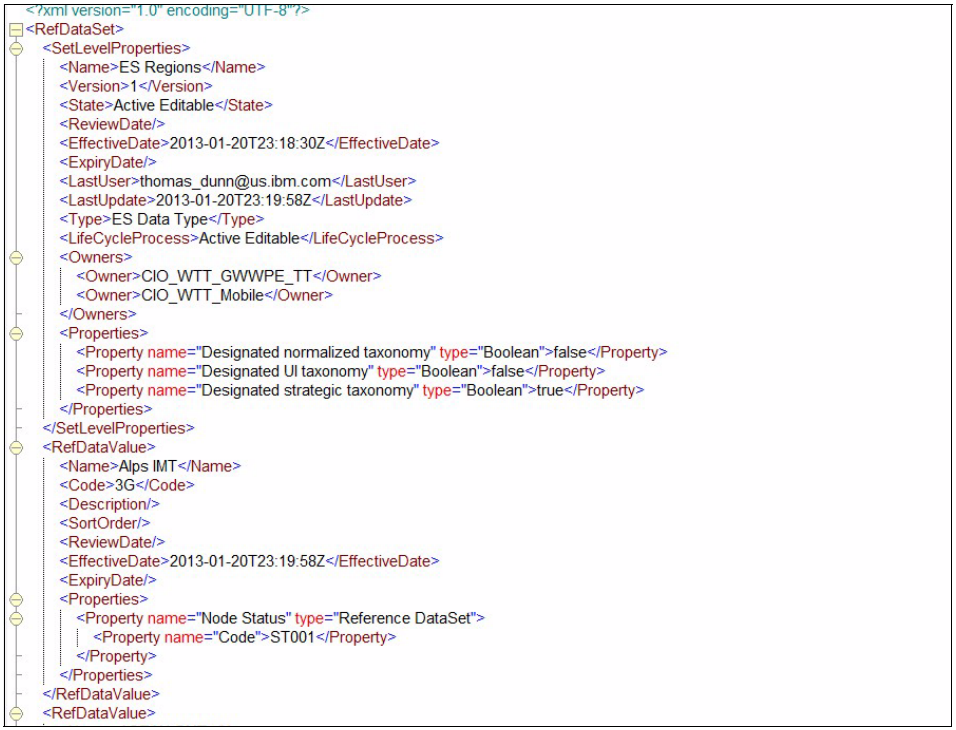


Figure 6-11 XML output from InfoSphere MDM Ref DM Hub reference data set

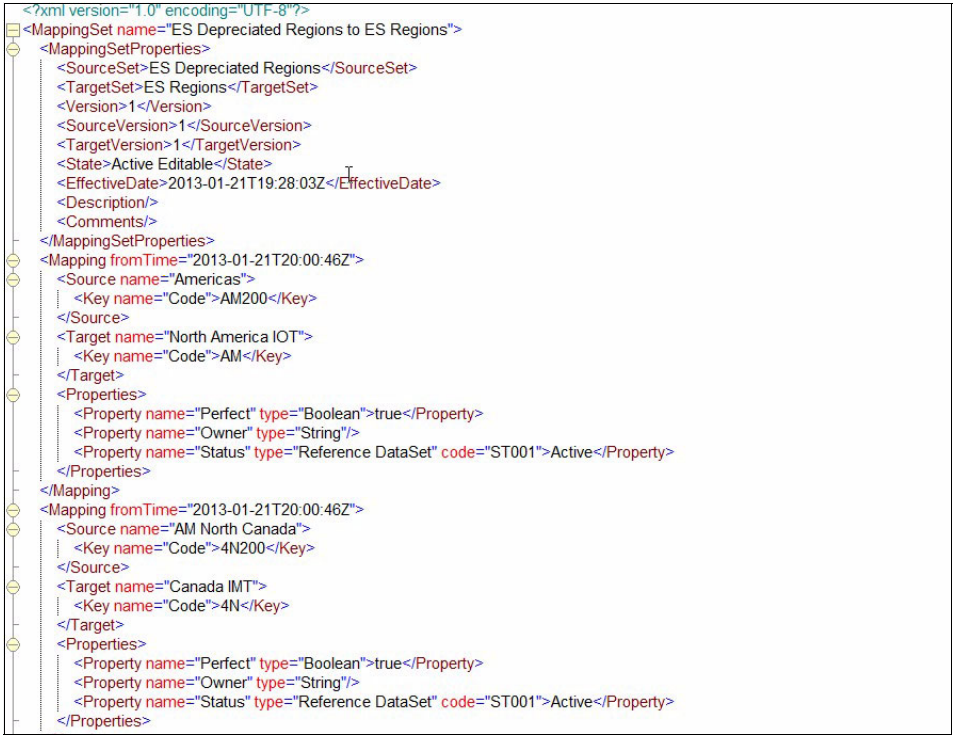
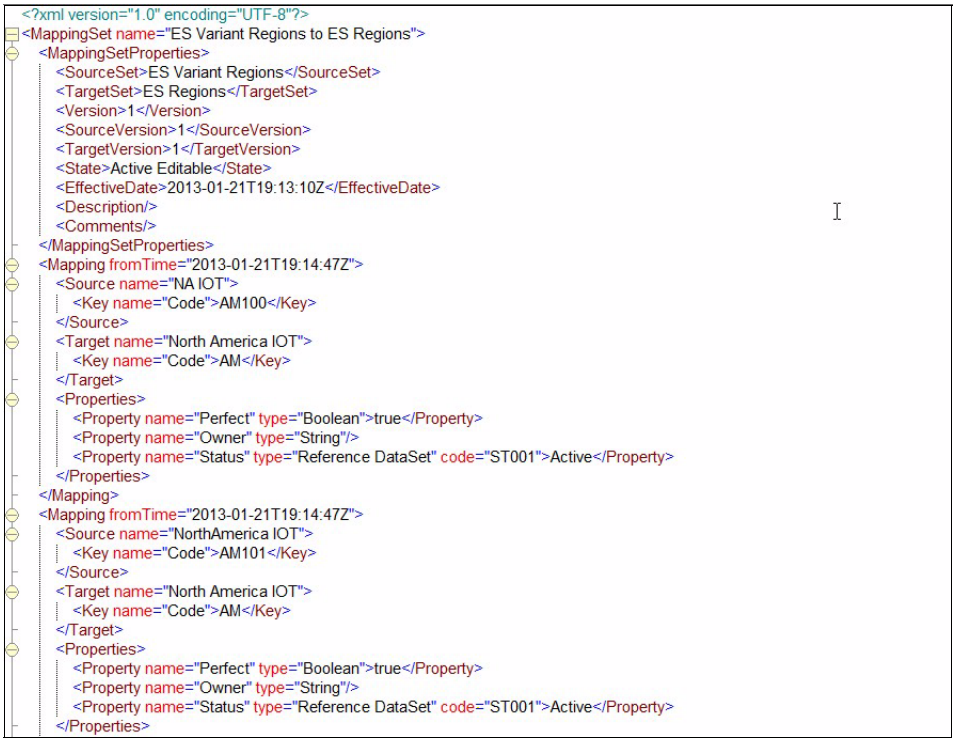


Figure 6-12 XML output from InfoSphere MDM Ref DM Hub mapping (Part 1 of 2)



```
<?xml version="1.0" encoding="UTF-8"?>
<MappingSet name="ES Variant Regions to ES Regions">
  <MappingSetProperties>
    <SourceSet>ES Variant Regions</SourceSet>
    <TargetSet>ES Regions</TargetSet>
    <Version>1</Version>
    <SourceVersion>1</SourceVersion>
    <TargetVersion>1</TargetVersion>
    <State>Active Editable</State>
    <EffectiveDate>2013-01-21T19:13:10Z</EffectiveDate>
    <Description/>
    <Comments/>
  </MappingSetProperties>
  <Mapping fromTime="2013-01-21T19:14:47Z">
    <Source name="NA IOT">
      <Key name="Code">AM100</Key>
    </Source>
    <Target name="North America IOT">
      <Key name="Code">AM</Key>
    </Target>
    <Properties>
      <Property name="Perfect" type="Boolean">true</Property>
      <Property name="Owner" type="String"/>
      <Property name="Status" type="Reference DataSet" code="ST001">Active</Property>
    </Properties>
  </Mapping>
  <Mapping fromTime="2013-01-21T19:14:47Z">
    <Source name="NorthAmerica IOT">
      <Key name="Code">AM101</Key>
    </Source>
    <Target name="North America IOT">
      <Key name="Code">AM</Key>
    </Target>
    <Properties>
      <Property name="Perfect" type="Boolean">true</Property>
      <Property name="Owner" type="String"/>
      <Property name="Status" type="Reference DataSet" code="ST001">Active</Property>
    </Properties>
```

Figure 6-13 XML output from InfoSphere MDM Ref DM Hub mapping (Part 2 of 2)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
    <!--
  9/25/2012 - TD - Fixed IE problem.
 -->
  <!-- Parameters passed in from the transform call -->
    <xsl:param name="wttEnv"></xsl:param>
    <xsl:param name="taxName"></xsl:param>
    <xsl:param name="dataPath"/>
    <xsl:param name="curversionnum"/>
    <xsl:param name="curstatus"/>
    <xsl:param name="inFilePrevious"/>
    <xsl:param name="preversionnum"/>
    <xsl:param name="prestatus"/>
    <xsl:param name="pubDateTime"/>
    <xsl:param name="mappingFile1"/>
    <xsl:param name="mappingFilePrevious1"/>
    <xsl:param name="mappingFile2"/>
    <xsl:param name="mappingFilePrevious2"/>
    <xsl:param name="mappingFile3"/>
    <xsl:param name="mappingFilePrevious3"/>
    <xsl:param name="tranName"/>
    <xsl:param name="tranID"/>
    <xsl:param name="tranDescription"/>
    <xsl:param name="tranStatus"/>
    <xsl:param name="mapDirection"/>
    <xsl:param name="taxNodeCustProps"/>

<!-- Variables created here in the xsl
    <xsl:variable name="map1" select="document($mappingFile1)"/>
    <xsl:variable name="map2" select="document($mappingFile2)"/>

<!--

    <xsl:template match="/RefDataSet">
      <xsl:text disable-output-escaping="yes">&lt;!DOCTYPE  html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;</xsl:text>
```

Figure 6-14 XSL used in TMT scenario to create an XML feed file

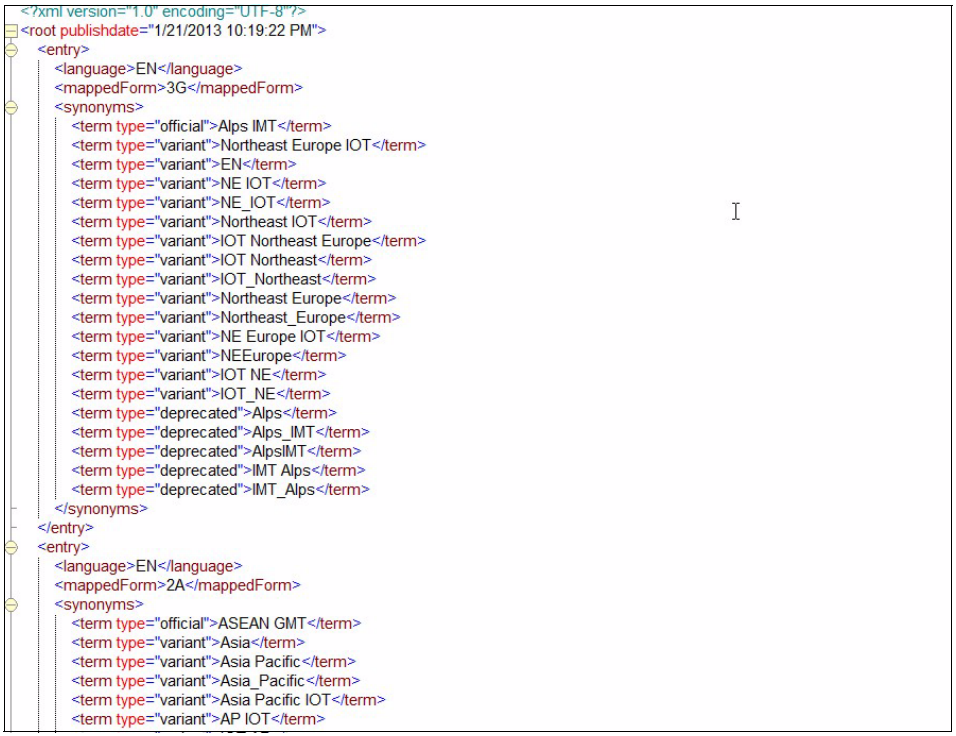


Figure 6-15 Final output XML after applying the XSL for the XML feed file

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>

<xsl:template match="*/text()[normalize-space()]">
    <xsl:value-of select="normalize-space()"/>
</xsl:template>

<xsl:template match="*/text()[not(normalize-space())]" />


<!-- This XSLT transforms RDM exported files into Enterprise Search backend dictionary format -->
<!-- Input files: regions.xml, variants2region.xml, deprecatedRegions2region.xml -->
<!-- Output file: region.mapping.xml -->

<!-- Parameters passed in from the transform call -->
   <xsl:param name="wttEnv">default - PoC</xsl:param>
   <xsl:param name="taxName">tax name</xsl:param>
   <xsl:param name="dataPath"/>
   <xsl:param name="curversionnum"></xsl:param>
   <xsl:param name="curdraftnum"></xsl:param>
   <xsl:param name="inFilePrevious"></xsl:param>
   <xsl:param name="preversionnum"></xsl:param>
   <xsl:param name="predraftnum"></xsl:param>
   <xsl:param name="pubDateTime"></xsl:param>
   <xsl:param name="mappingFile1"></xsl:param>
   <xsl:param name="mappingFilePrevious1"></xsl:param>
   <xsl:param name="mappingFile2"></xsl:param>
   <xsl:param name="mappingFilePrevious2"></xsl:param>
   <xsl:param name="mappingFile3"></xsl:param>
   <xsl:param name="mappingFilePrevious3"></xsl:param>
   <xsl:param name="tranName"></xsl:param>
   <xsl:param name="mapDirection"></xsl:param>


<xsl:template match="/">
<root>
 <xsl:attribute name="publishdate"><xsl:value-of select="$pubDateTime"/></xsl:attribute>
```

Figure 6-16 Final output XML after applying the XSL for the XML feed file

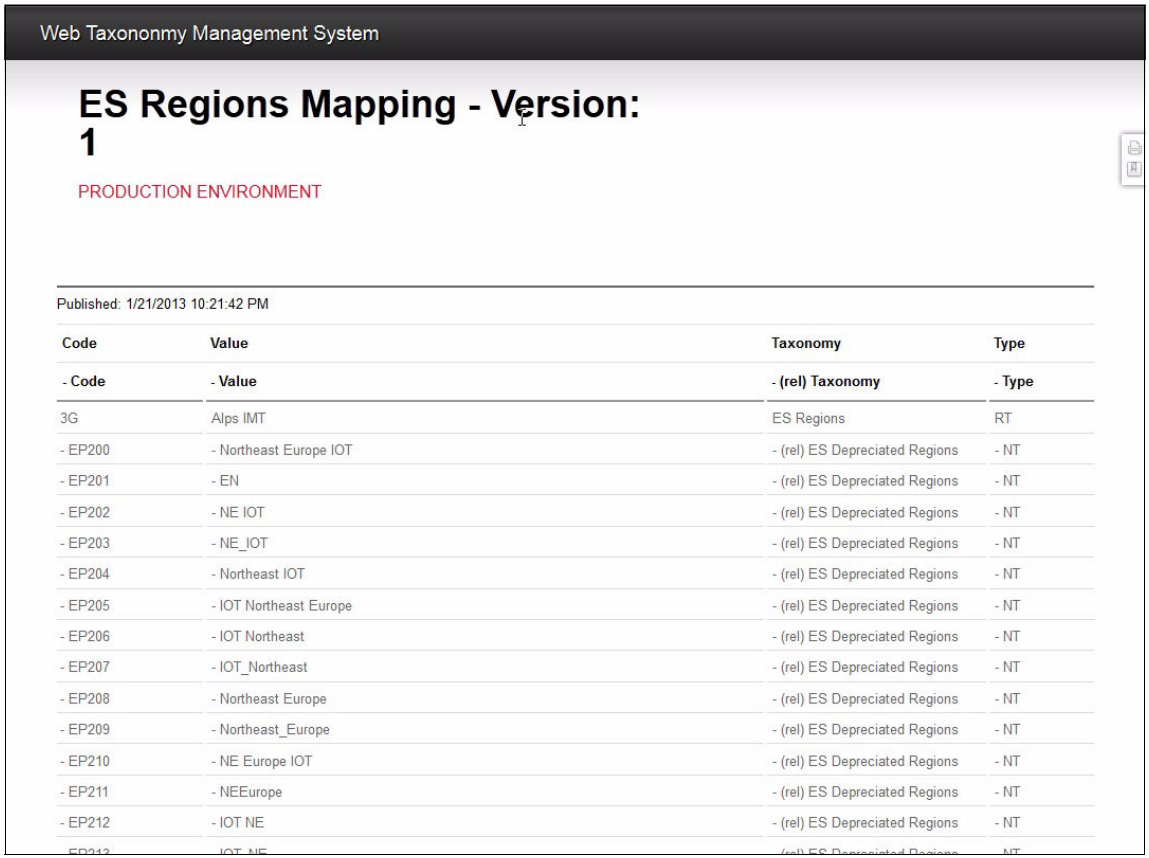


Figure 6-17 Final output XML after applying the XSL for the report

We provide application-specific transformations for some consuming applications and let certain consuming applications transform the standard XML to the specific XML format that the applications need.

6.3 Transcoding

Although enterprise reference data tends to be standardized before getting consumed by individual operational systems and applications, it often differs in representation or semantics across different, often siloed, applications. This semantic difference can be unavoidable because applications often require their local representations for improved processing.

For example, in Figure 6-18, fields for country codes in source and target represent the set of country codes, which is a type of reference data. However, as observed, source and target reference values have different code representations for the same countries. For this reason, before performing data integration or distribution, the source representation must be transcoded to one that the target can understand. This process is known as *reference data transcoding* and is a key step in many integration scenarios (for example, master data integration and distribution pipeline).

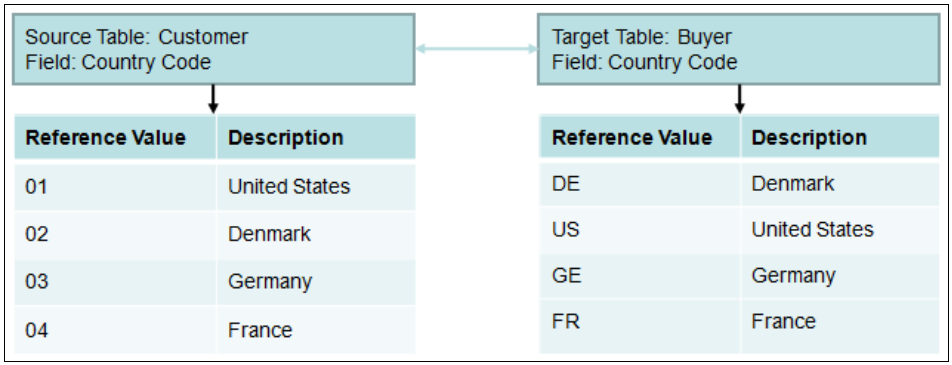


Figure 6-18 Representational differences across source and target tables

In typical reference data hub implementation, such as InfoSphere MDM Ref DM Hub, reference data transcoding is achieved through reference data mappings from a source reference data set to a target reference data set. Depending on the integration scenario, the transcoding might be simple (one-to-one) where a single source representation is mapped to a single target representation, or complex (one-to-many, many-to-one, or many-to-many), where one or more source representations are mapped to one or more target representations. An example of one-to-many scenario is where a single canonical representation is mapped to multiple application specific representations. Each scenario is described next.

6.3.1 Simple transcoding

Simple transcoding is applicable in integration or distribution scenarios where there is a single source system or a set of source systems, all backed up by the same code tables, and a single target system or a set of target systems, all backed up by the same code tables. For performing the required mappings, reference data from sources and targets is imported into InfoSphere MDM Ref DM Hub using one of the mechanisms described in 6.1, “Data loading and import” on page 112. After they are imported, a mapping set can be defined by using the UI, with source and targets specified as the respective sets into which reference data was imported. Finally, the values mappings are created either by

using the UI or through an import process. To perform import, a mapping file must be constructed with source and target specification and any properties you want for the mappings. Figure 6-19 illustrates the concept.

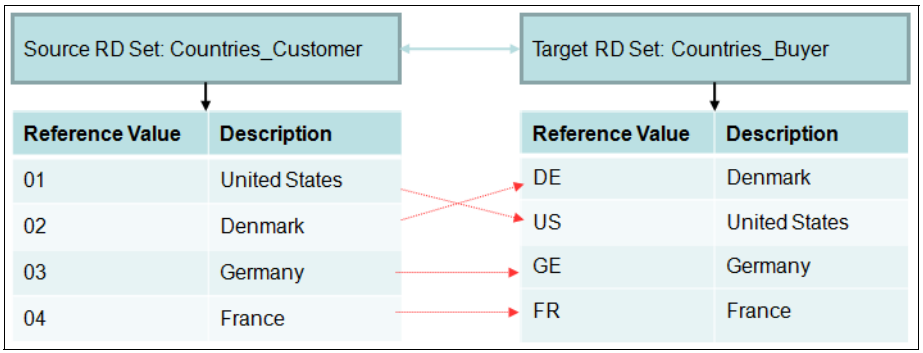


Figure 6-19 Defining reference data mappings from source to target set

As reference data goes through its lifecycle, stewards or business users make changes (additions, deletions, updates) to these mappings as necessary. Eventually, these mappings can be exported by using the UI or through a batch interface, and distributed to downstream systems for transcoding during integration scenarios.

For example, during master data integration and distribution process, the mappings are used to construct translation tables that help in semantic alignment of values during the alignment area. The alignment handles two types of transcoding:

- The source and target systems have the same code value but with different semantics on the description.
- The source and target systems have different code value sets for the same reference data domain.

In either type, without replacing the reference data values from the source MDM system with their semantic equivalent in the target, the semantic integrity of the records during data integration process cannot be guaranteed.

The mechanism for this replacement is a translation table similar to the one shown in Figure 6-20 which defines the rules to govern the replacement of reference values.

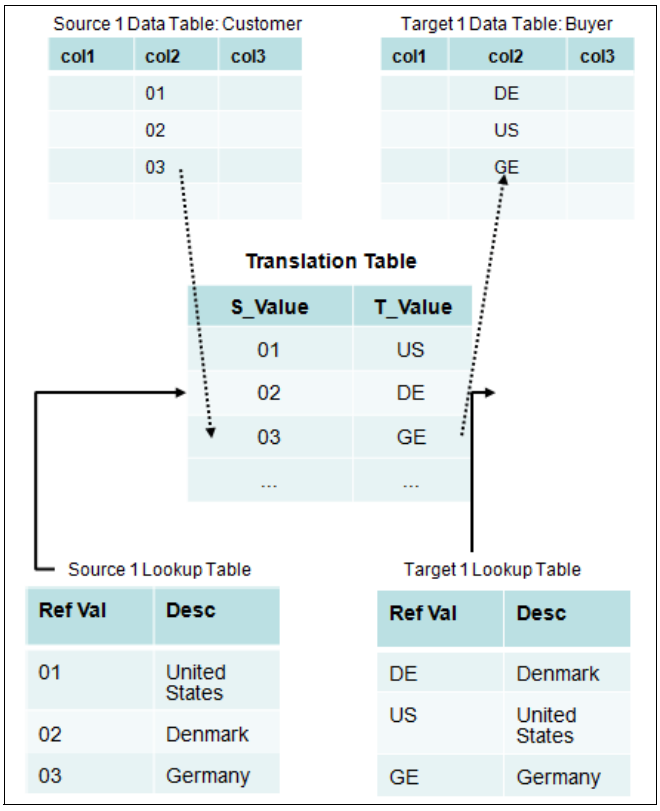


Figure 6-20 Translation tables for reference data alignment

6.3.2 Complex transcoding using multiple maps

Complex transcoding is applicable in scenarios where either multiple source systems are backed up by different code tables or multiple target systems are all backed up by the different code tables. For doing the required mappings, reference data from all sources and targets is imported into reference data management hub by using one of the mechanisms described in 6.1, “Data loading and import” on page 112. After they are imported, multiple mapping sets are defined by using the UI with each pair of source and target specified as the respective sets into which reference data was imported. Finally, the values mappings are created and undergo evolution as part of lifecycle activities.

When distributed, these mappings get utilized in a similar manner as simple transcoding, except the translation tables have entries for each pair of source and target systems (as shown in Figure 6-21).

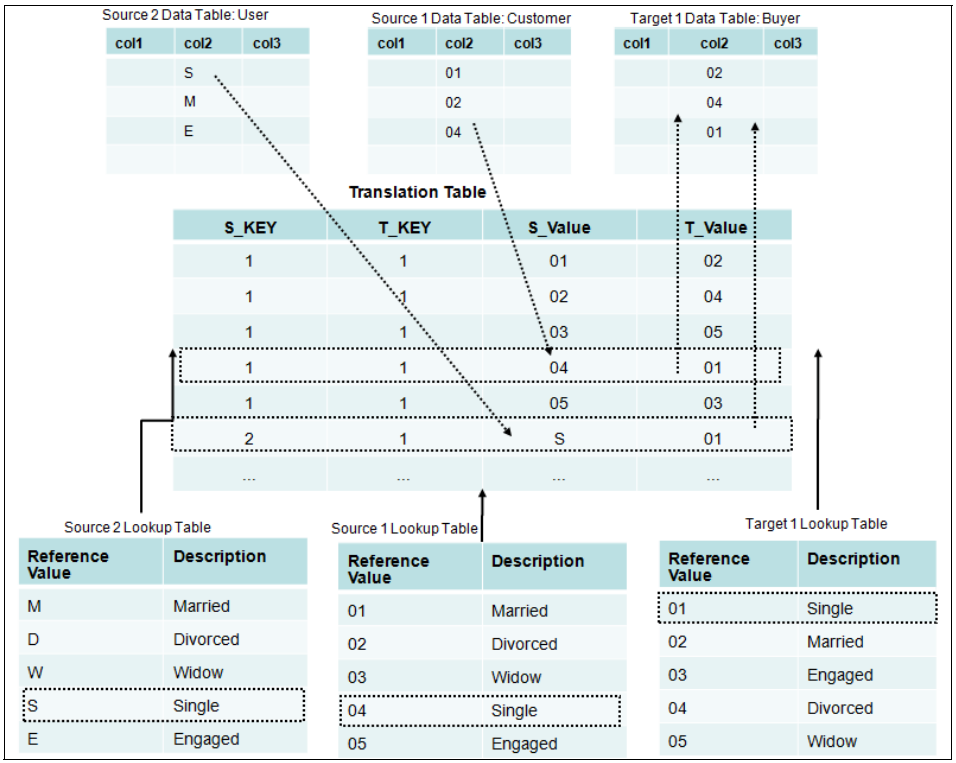


Figure 6-21 Complex transcoding across more than one source and target pair

6.4 Attribute-level permissions

A comprehensive reference data implementation, such as InfoSphere MDM Ref DM Hub, provides a fine-grained access control and security model that can be fine-tuned to appropriately define attribute-level permissions. Using this security feature, the author of a reference data entity (set, mapping or hierarchy) is able to restrict access to certain attributes of that entity depending on the role of the user who is accessing the entity. For example, one could configure a custom property defined on a reference data set to be read-only, hidden, or updatable depending on the role of the logged-in user who is trying to access that reference data set. This fine-grain attribute-level access control has direct implications on data import and export because, depending on the visibility of an attribute to a user,

the user might or might not be able to import data into it, or export data contained in it.

As seen in previous sections, in InfoSphere MDM Ref DM Hub, the UI presents an import wizard. In this wizard, an option is given to map the data columns (Figure 6-5 on page 117) in the (character-delimited) import file to the corresponding attributes of the entity into which the data is being imported. If attribute-level permissions are enabled, the attributes on the map screen are displayed appropriately, depending on the level of visibility that you have. For example, you are not presented with the “hidden” fields. Similarly, you are allowed to import data into the fields that are configured as “read-only” for his role. Similarly, while exporting data from a reference data entity, you are not allowed to export the attributes to which the access is restricted.

This security feature can be contrasted with the security that is based on ownership groups, where certain users (grouped by ownership groups) are not allowed to modify specific reference data sets that do not list those groups in their owners field. In such cases, the users are not able to modify any of the values in those sets. Whereas, with attribute-level security, users are able to modify the values; however, some attributes might be unmodifiable or hidden. This feature can be easily configured on a per set basis by using property files at the time of implementation of the scenario.

6.5 IBM InfoSphere Information Server

IBM InfoSphere Information Server is a data integration platform that helps you understand, cleanse, transform, and deliver trusted information to your critical business initiatives, such as big data, master data management, and point-of-impact analytics. Because reference data is prevalent in the information landscape, it plays a key role in enabling several integration scenarios across the IBM InfoSphere Information Server suite of technologies. This section discusses several of these integration scenarios.

6.5.1 InfoSphere Business Glossary

IBM InfoSphere Business Glossary (BG) is a tool that provides definitions for business terms across an enterprise for unified understanding. This section describes how InfoSphere Business Glossary can be integrated with InfoSphere MDM Ref DM Hub to enable seamless determination of range of valid values in InfoSphere MDM Ref DM Hub for the terms in Business Glossary.

Figure 6-22 describes the integration procedure at a high level.

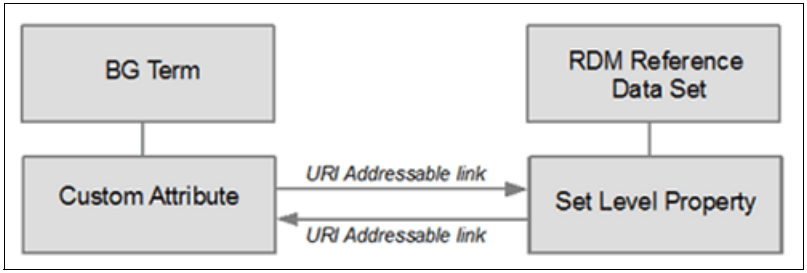


Figure 6-22 InfoSphere Business Glossary and InfoSphere MDM Ref DM Hub integration

The process is as follows:

1. Through the Administration tab of InfoSphere MDM Ref DM Hub user interface, create a new data type to be used as a base type for reference data sets, with a link to Business Glossary as shown in Figure 6-23. Create a new set level property for the base data type that has a URL type. This set property is not required, but it can have a default value of the base URL to Business Glossary.

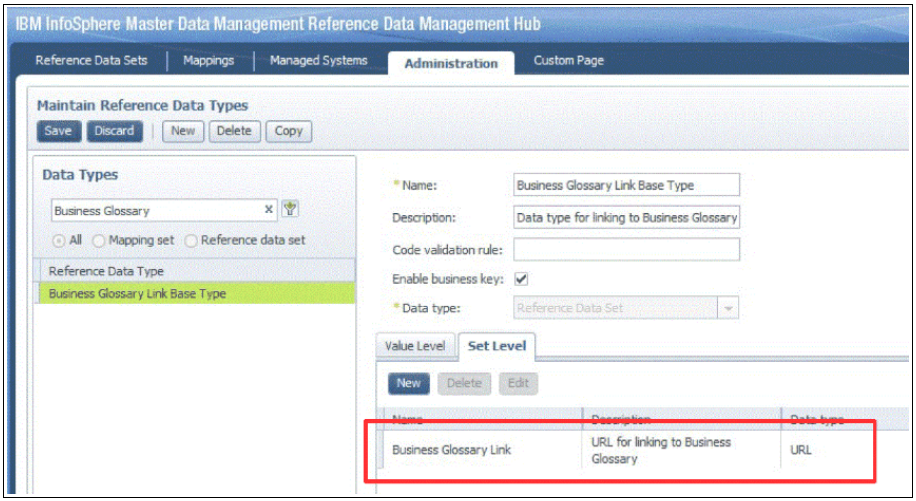


Figure 6-23 Creating a custom data type in InfoSphere MDM Ref DM Hub

2. In Business Glossary, a custom attribute must be created to contain the link to the reference data set that is created in InfoSphere MDM Ref DM Hub. For

this step, through the Administration tab of Business Glossary, create a custom attribute that applies only to terms and has a string type.

In Figure 6-24, the `Valid Values` attribute is created to link the country codes term to a reference data set representing the list of valid country codes for that term.

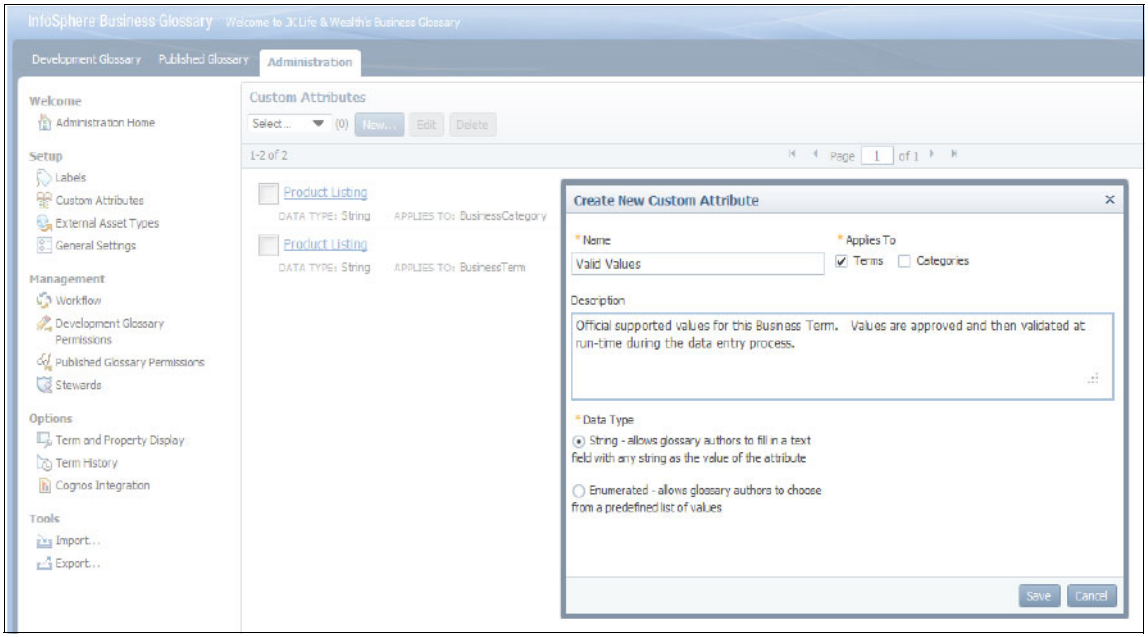


Figure 6-24 Creating custom attribute in InfoSphere Business Glossary

3. After completion of the initial setup of Business Glossary and InfoSphere MDM Ref DM Hub, you can create terms and reference data sets and establish cross-links between them. Links can be created by using URI addressable links, available in both InfoSphere MDM Ref DM Hub and Business Glossary. The linking steps are as follows:

   a. Create a new reference data set in InfoSphere MDM Ref DM Hub that uses the base type that was created with a URL set property.

   b. Create a new term in Business Glossary so that the created reference data set represents reference data for the term.

   c. Fill in the custom attribute of the team with a URL to the InfoSphere MDM Ref DM Hub set that was created. You can use the following special format of in the valid values attribute of Business Glossary to create a hyperlink to the specified URL that is displayed with the provided description text:

      `[<URL> | <Description>]`

InfoSphere MDM Ref DM Hub provides a URI-addressable link that can be used to link to a current summary of a reference data set. To use this link, create a URL in the following format:

```
<Base RDM URL>/RefDataClient/ShowRefDataSet.html?setName=<name of reference
data set>
```

In place of any spaces that occur in the name of the reference data set, you must use the following characters:

```
%20
```

For example, a reference data set named `Country Codes` results in the following URL:

```
<Base RDM URL>/RefDataClient/ShowRefDataSet.html?setName=Country%20Codes
```

This URL is then placed as the Valid Values attribute. See Figure 6-25.

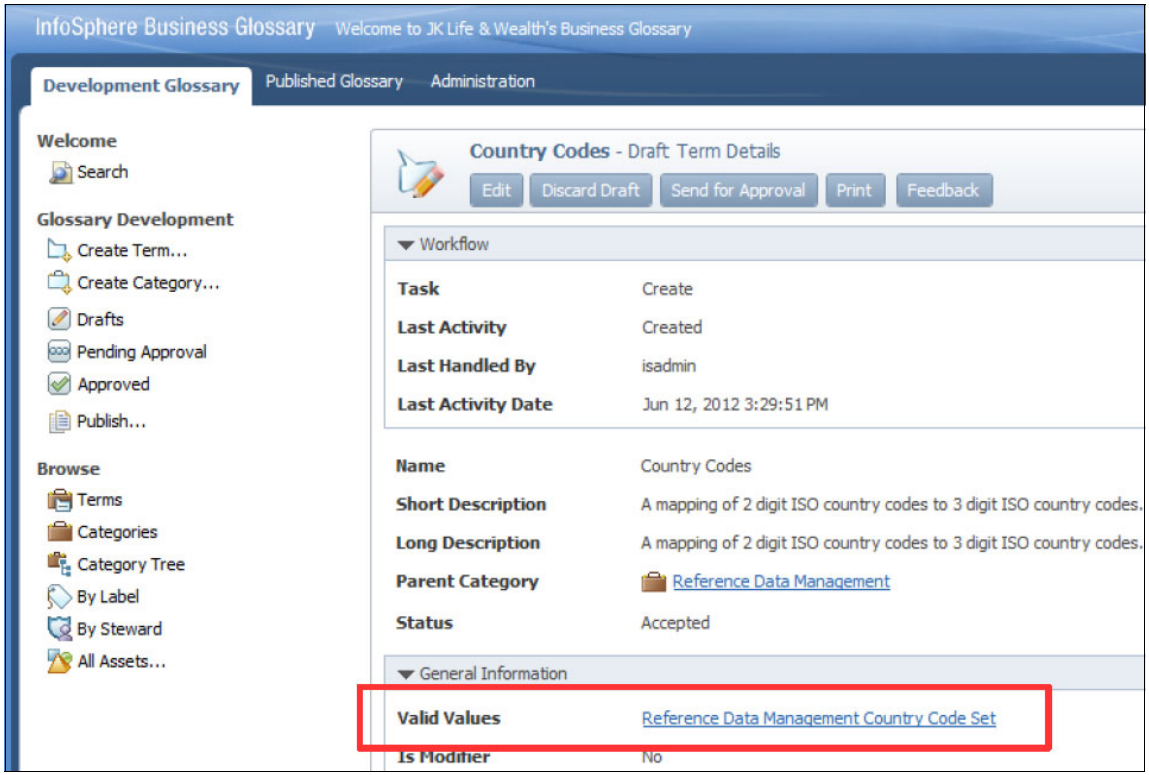


Figure 6-25 Linking InfoSphere MDM Ref DM Hub URL to terms in InfoSphere Business Glossary

6.5.2 InfoSphere Information Analyzer

In this section, we describe an integration scenario between InfoSphere MDM Ref DM Hub and InfoSphere Information Analyzer This integration comprises of the following main steps:

- Using InfoSphere Information Analyzer to create a reference table from a column analysis of some set of data
- Transferring that reference data into RDM to be managed and manipulated
- Transferring the updated reference data back into InfoSphere Information Analyzer

Figure 6-26 illustrates the high-level overview of the integration of InfoSphere MDM Ref DM Hub and InfoSphere Information Analyzer.

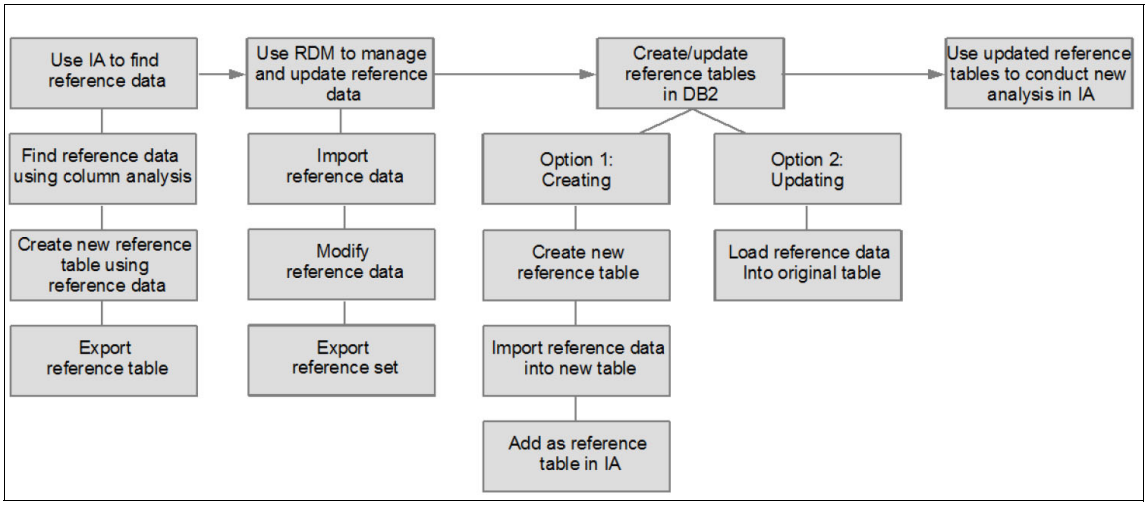


Figure 6-26 InfoSphere MDM Ref DM Hub and InfoSphere Information Analyzer integration overview

The reference data in both RDM and Information Analyzer must be kept synchronized to have consistent reference data across the two tools. This requirement provides accurate management and analysis of data. The integration process consists of four main steps:

1. Finding reference data in Information Analyzer
2. Using InfoSphere MDM Ref DM Hub to manage and update reference data
3. Creating or updating reference tables for the modified reference data in DB2
4. Updating the reference tables in Information Analyzer to conduct analysis with the new reference data set

Finding reference data in Information Analyzer

You can use Information Analyzer to find reference data by conducting a column analysis on a column of data from a table. The results can be saved in a reference table, which can then be exported for use by other tools, such as RDM.

To find reference data, use a column analysis on an existing table of data:

1. Open IBM InfoSphere Information Server Console to gain access to Information Analyzer services.
2. Open or create an Information Analyzer project.
3. Select **Investigate** → **Column Analysis**, select the column from which to find reference data, and then select to open or run a column analysis for that column of data.
4. After viewing an analysis, select **View Details** to view the details of that column analysis, such as the statistics determined by the analysis. See Figure 6-27.

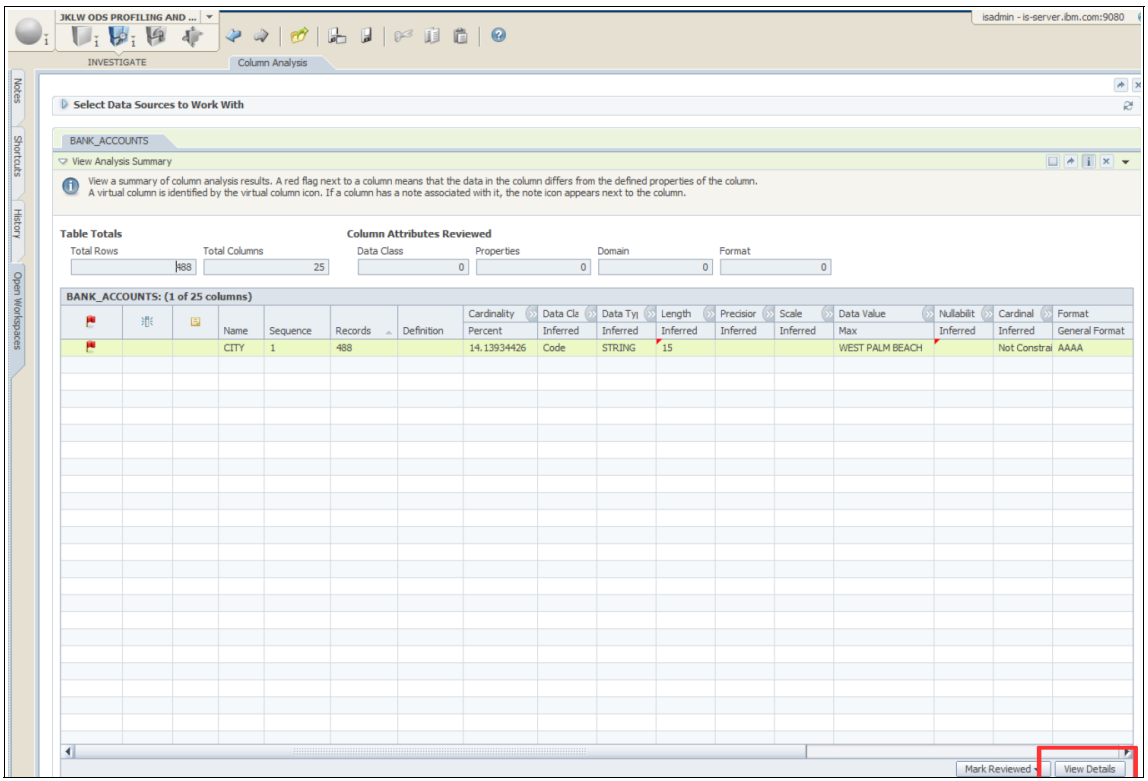


Figure 6-27 Results of running a column analysis on a candidate reference data column

To create a new reference table that contains reference data for the selected column, complete the following steps:

1. Create a new table by selecting **Reference Tables** → **New Reference Table**.
2. Choose a name for the reference table, select the type as **Valid** and save it. Currently, only the Valid type is supported in RDM for reference data.
3. View the new reference table by selecting **Reference Table** → **View Reference Table** from the column analysis view.
4. Export the reference table by using CSV format:
   a. Select I**nvestigate** → **Table Management**.
   b. Open the newly created reference table.
   c. Select **Export**. Select the path of where to save the file and keep the remaining options as the default values. Choose to save the reference table using CSV format (Figure 6-28).

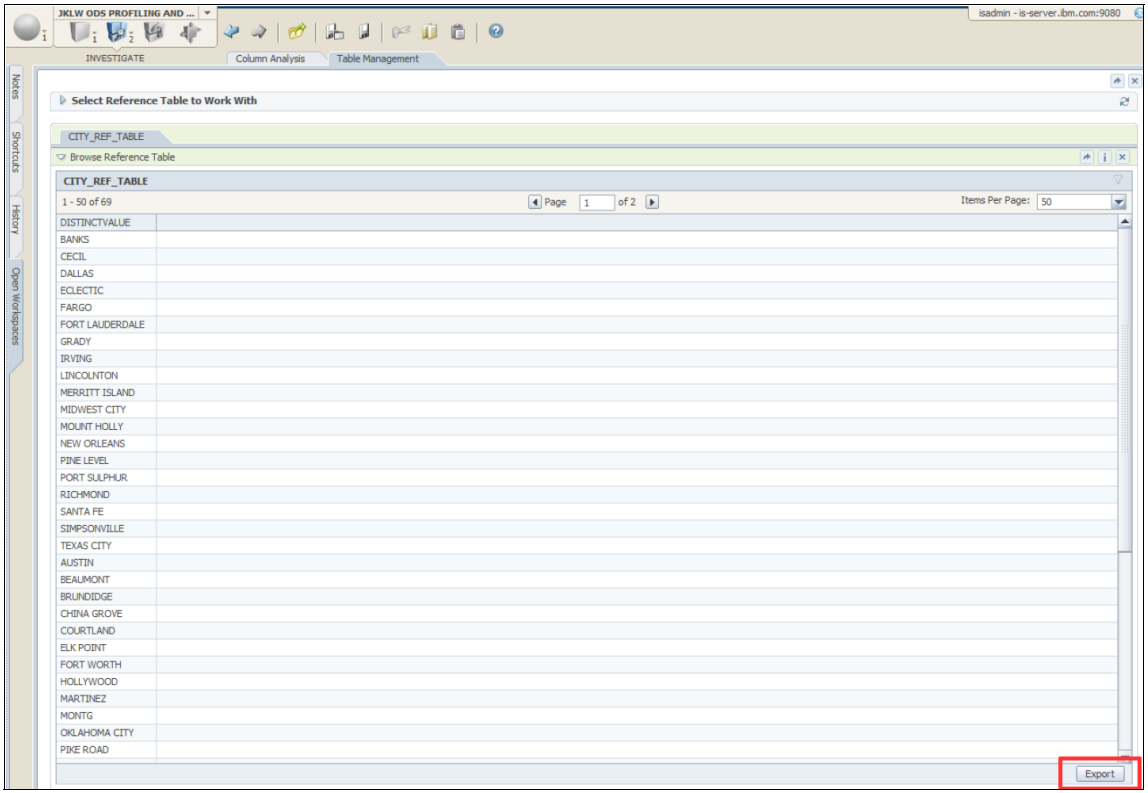


Figure 6-28 Exporting a reference table from Information Analyzer

Managing reference data in InfoSphere MDM Ref DM Hub

When reference data is discovered in Information Analyzer, it can be imported into InfoSphere MDM Ref DM Hub for management and manipulation. After reference data is modified in RDM, it can be exported so that Information Analyzer can be updated with those modifications.

Use the following steps to import reference data from a CSV file:

1. Determine which data type is to be used to create a reference set for the reference data. If the data type is not already created and if the user is not using the default reference data type, a custom data type must be created through the Administration tab in the InfoSphere MDM Ref DM Hub web interface.
2. Create a new reference set using the selected data type.
3. Select to import the reference table that was exported from Information Analyzer into the created reference set by right-clicking the new reference data set and selecting **Import**.
4. In the Import Reference Data Set window, select the CSV file that was exported from Information Analyzer as the file to import.
5. In the Map Columns tab, select **DISTINCT VALUE** as the import file column for both the Code and Name properties.
6. In the Preview File tab, verify that the data shown is the correct reference data, then click **Finish** to import the data.

If required, make any necessary changes to the reference data set, such as adding or deleting values in the set through the InfoSphere MDM Ref DM Hub web interface.

You can then export the reference data set in CSV format by using the following steps:

1. Right-click the reference set and select **Export** (Figure 6-29).

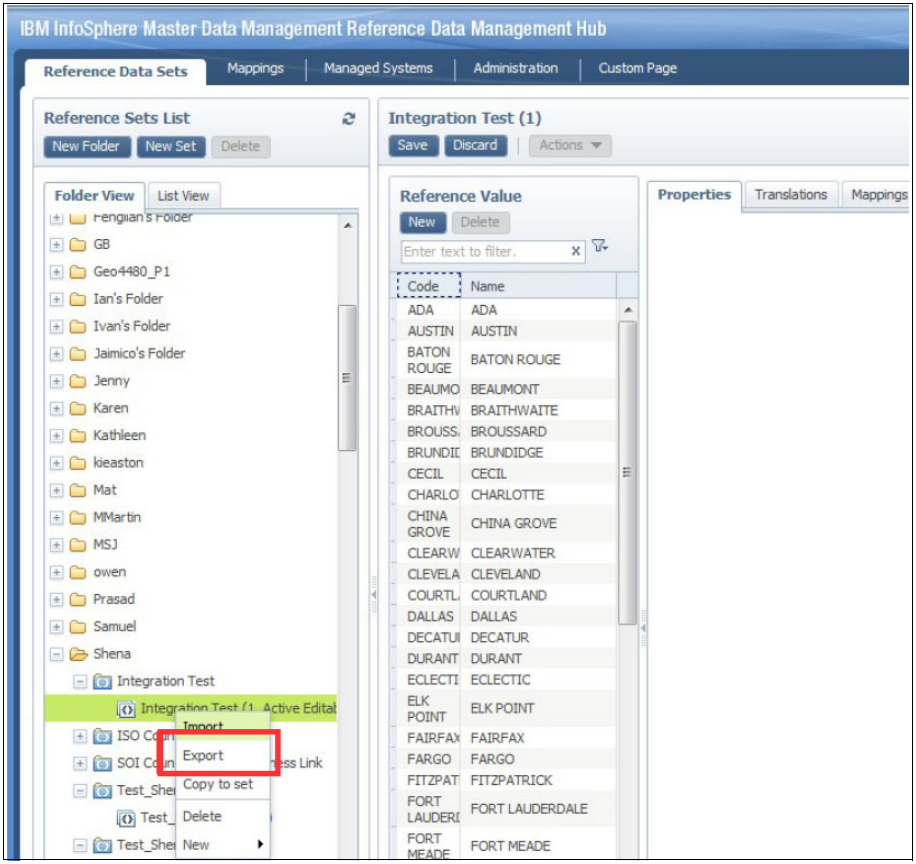


Figure 6-29 Exporting modified reference data set

2. Keep the default values of Code and Name for the Value properties being exported and click **Finish**.
3. Select to download the exported file.

Updating the reference tables running report in InfoSphere Information Analyzer

The exported reference data can be brought back into Information Analyzer by updating the original code table with the new data. To do this step, you create a JDBC connection to the original code table in Information Analyzer and update the original table data with the exported CSV file.

You can then run a new analysis in Information Analyzer with the updated reference table by using the following steps:

1. Select **Investigate** → **Column analysis**.
2. Select the column that the reference data applies to.
3. Select **View Details** to view the details of the last column analysis.
4. Navigate to the **Domain and Completeness** tab.
5. Set the domain type as **Reference table**.
6. Select **Table Name** as the name of the new reference table. This is the reference table to run the analysis against.
7. Save the selections.
8. Select **Rebuild inferences** and observe that the analysis statistics is changed (Figure 6-30).

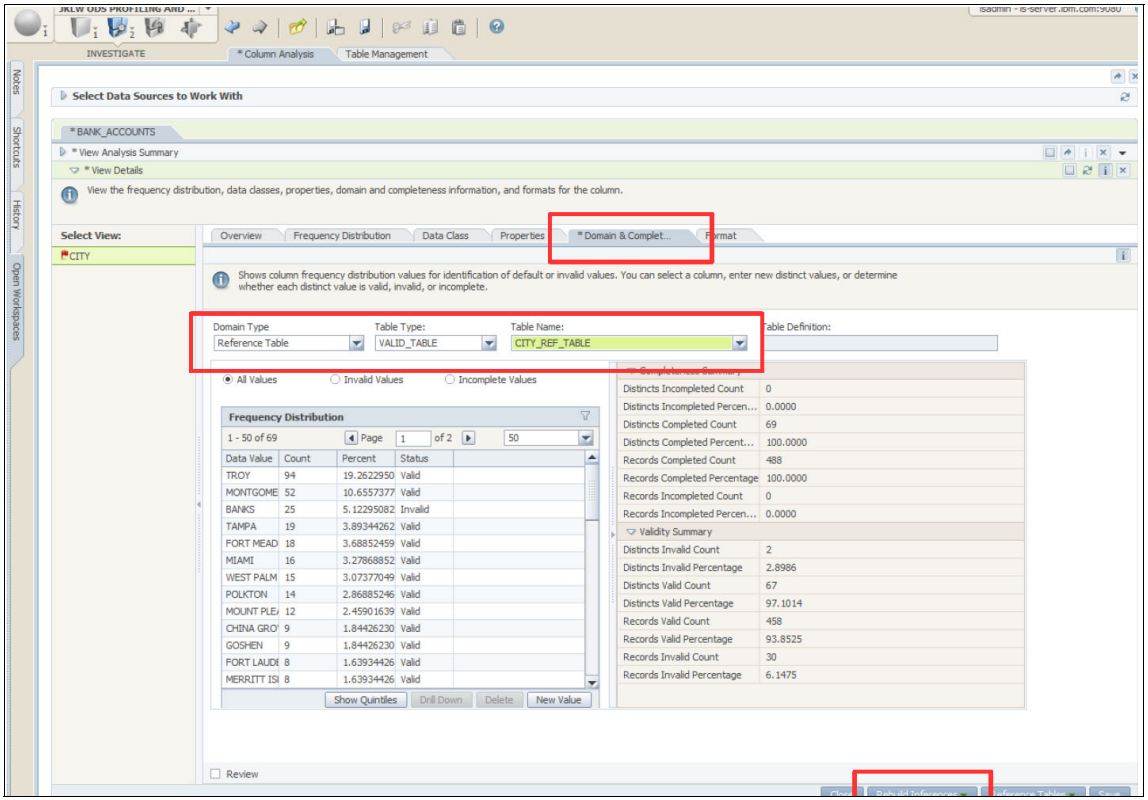


Figure 6-30 Running column analysis with the updated reference table

6.5.3 Conversion workbench

This section describes how InfoSphere MDM Ref DM Hub is used in SAP migrations and consolidations. Moving business data, for example customer objects and material objects, from one or more established systems to a target SAP system is the main process of such projects. While the data is moved from the source to the target system, the fields containing reference data must be transcoded from the values used in a source system into the values to be used in the target system. You can use InfoSphere MDM Ref DM Hub to provide the mapping rules required to perform the transcoding.

All SAP migration and consolidation projects are similarly structured into three phases:

- Discover: In the first phase, the data models of the various source systems are discovered, followed by an extraction of the data into a staging area.
- Prepare: In this second phase, the major steps of the prepare phase are structural alignment and data cleansing.
- Deliver: In this third phase, the data is transformed from the alignment data model to the target system data model and loaded into the target system.

The InfoSphere Conversion Workbench for SAP Application provides the blueprint that describes the necessary tasks to do in each phase and the appropriate tools to realize these tasks.

The blueprint snippet in Figure 6-31describes the basic sequence of structural alignment, transcoding, and data cleansing. The transcoding step is based upon transcoding tables providing the source to target value mappings.

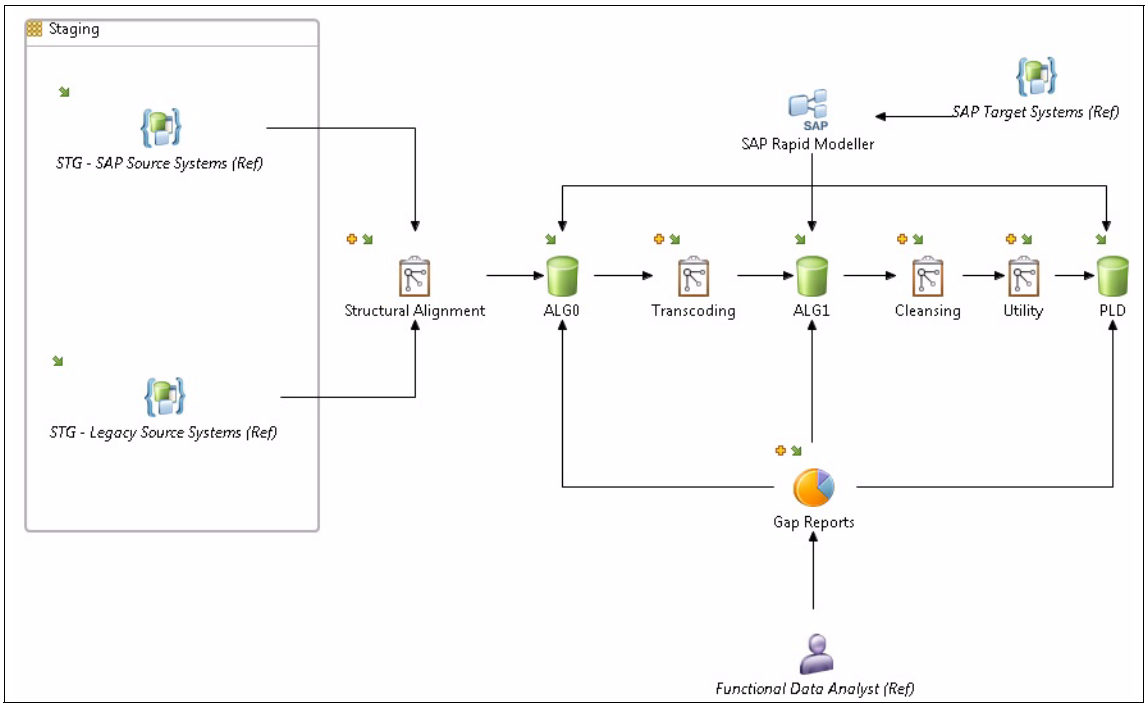


Figure 6-31 Transcoding in SAP migration and consolidation projects

The snippet in Figure 6-32 illustrates that the InfoSphere MDM Ref DM Hub is used by the Functional Data Analyst to define the value mappings. The Conversion Workbench application then populates the transcoding tables based on the value mappings defined in InfoSphere MDM Ref DM Hub.

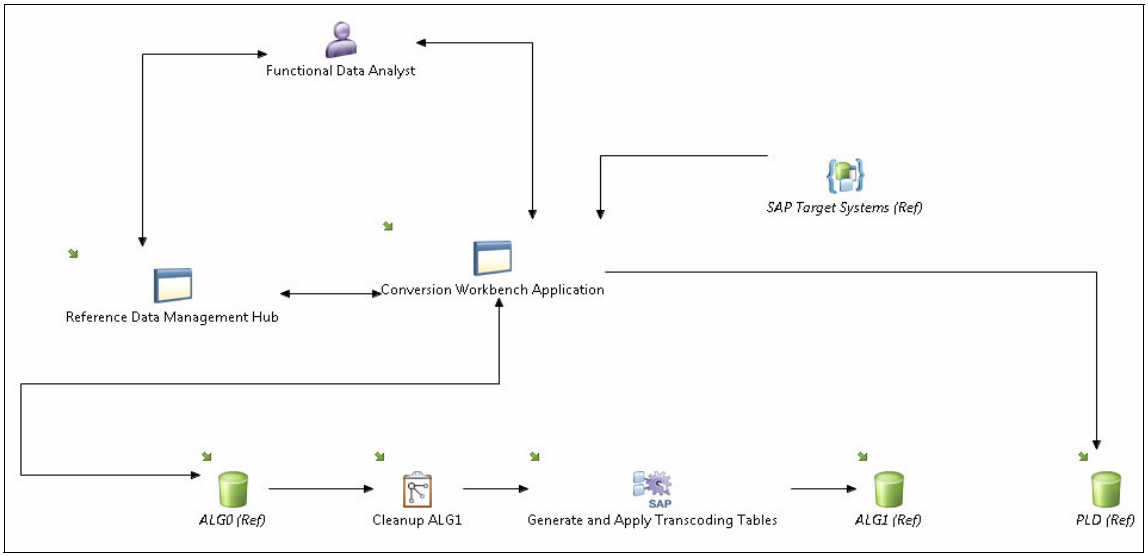


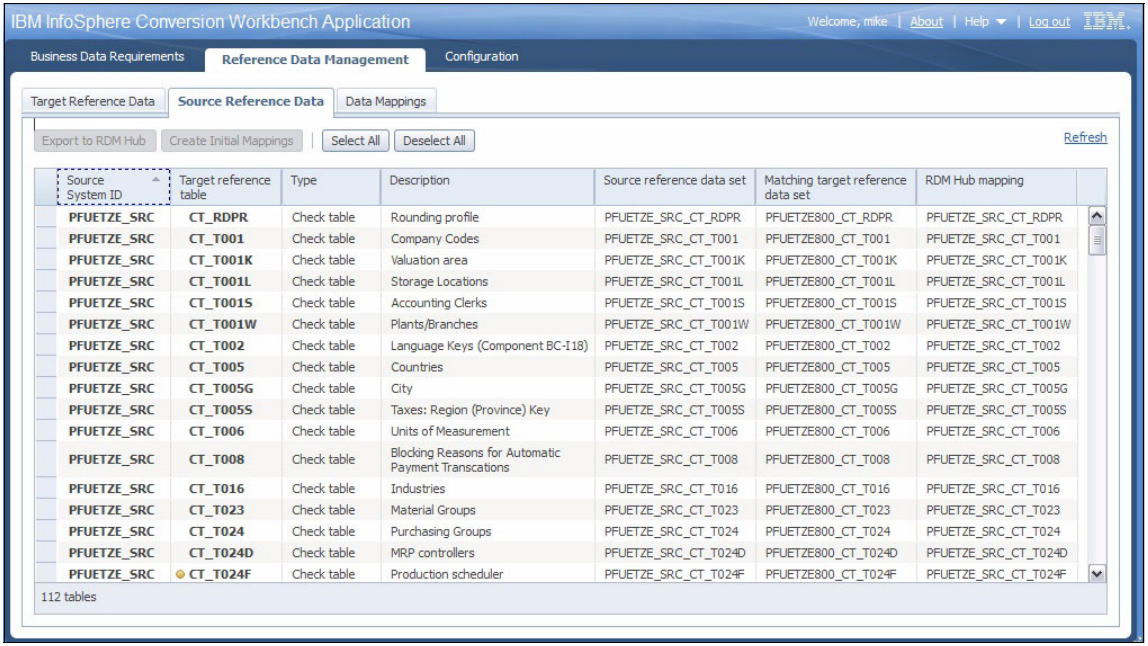
Figure 6-32 Function Data Analyst uses InfoSphere MDM Ref DM Hub and the Conversion Workbench Application to populate transcoding tables

To define value mappings and populate the transcoding tables, the Function Data Analyst (FDA) does the following tasks by using the Conversion Workbench Application and InfoSphere MDM Ref DM Hub:

1. The FDA uses the Conversion Workbench Application to extract the reference data of the target SAP system and all source systems into InfoSphere MDM Ref DM Hub.

   In addition, all required mapping objects that associate the appropriate source and target Reference Data Sets are created in InfoSphere MDM Ref DM Hub.

Figure 6-33 shows the created InfoSphere MDM Ref DM Hub source sets, matching target sets and the initially created mapping objects are listed in the Conversion Workbench Application.



Source System ID	Target reference table	Type	Description	Source reference data set	Matching target reference data set	RDM Hub mapping
PFUETZE_SRC	CT_RDPR	Check table	Rounding profile	PFUETZE_SRC_CT_RDPR	PFUETZE800_CT_RDPR	PFUETZE_SRC_CT_RDPR
PFUETZE_SRC	CT_T001	Check table	Company Codes	PFUETZE_SRC_CT_T001	PFUETZE800_CT_T001	PFUETZE_SRC_CT_T001
PFUETZE_SRC	CT_T001K	Check table	Valuation area	PFUETZE_SRC_CT_T001K	PFUETZE800_CT_T001K	PFUETZE_SRC_CT_T001K
PFUETZE_SRC	CT_T001L	Check table	Storage Locations	PFUETZE_SRC_CT_T001L	PFUETZE800_CT_T001L	PFUETZE_SRC_CT_T001L
PFUETZE_SRC	CT_T001S	Check table	Accounting Clerks	PFUETZE_SRC_CT_T001S	PFUETZE800_CT_T001S	PFUETZE_SRC_CT_T001S
PFUETZE_SRC	CT_T001W	Check table	Plants/Branches	PFUETZE_SRC_CT_T001W	PFUETZE800_CT_T001W	PFUETZE_SRC_CT_T001W
PFUETZE_SRC	CT_T002	Check table	Language Keys (Component BC-I18)	PFUETZE_SRC_CT_T002	PFUETZE800_CT_T002	PFUETZE_SRC_CT_T002
PFUETZE_SRC	CT_T005	Check table	Countries	PFUETZE_SRC_CT_T005	PFUETZE800_CT_T005	PFUETZE_SRC_CT_T005
PFUETZE_SRC	CT_T005G	Check table	City	PFUETZE_SRC_CT_T005G	PFUETZE800_CT_T005G	PFUETZE_SRC_CT_T005G
PFUETZE_SRC	CT_T005S	Check table	Taxes: Region (Province) Key	PFUETZE_SRC_CT_T005S	PFUETZE800_CT_T005S	PFUETZE_SRC_CT_T005S
PFUETZE_SRC	CT_T006	Check table	Units of Measurement	PFUETZE_SRC_CT_T006	PFUETZE800_CT_T006	PFUETZE_SRC_CT_T006
PFUETZE_SRC	CT_T008	Check table	Blocking Reasons for Automatic Payment Transcations	PFUETZE_SRC_CT_T008	PFUETZE800_CT_T008	PFUETZE_SRC_CT_T008
PFUETZE_SRC	CT_T016	Check table	Industries	PFUETZE_SRC_CT_T016	PFUETZE800_CT_T016	PFUETZE_SRC_CT_T016
PFUETZE_SRC	CT_T023	Check table	Material Groups	PFUETZE_SRC_CT_T023	PFUETZE800_CT_T023	PFUETZE_SRC_CT_T023
PFUETZE_SRC	CT_T024	Check table	Purchasing Groups	PFUETZE_SRC_CT_T024	PFUETZE800_CT_T024	PFUETZE_SRC_CT_T024
PFUETZE_SRC	CT_T024D	Check table	MRP controllers	PFUETZE_SRC_CT_T024D	PFUETZE800_CT_T024D	PFUETZE_SRC_CT_T024D
PFUETZE_SRC	CT_T024F	Check table	Production scheduler	PFUETZE_SRC_CT_T024F	PFUETZE800_CT_T024F	PFUETZE_SRC_CT_T024F

Figure 6-33 FDA creates InfoSphere MDM Ref DM Hub source and target reference data sets and initial mapping objects

2. After all sets and mapping objects are created in InfoSphere MDM Ref DM Hub, the FDA defines the value mappings in InfoSphere MDM Ref DM Hub as in Figure 6-34.

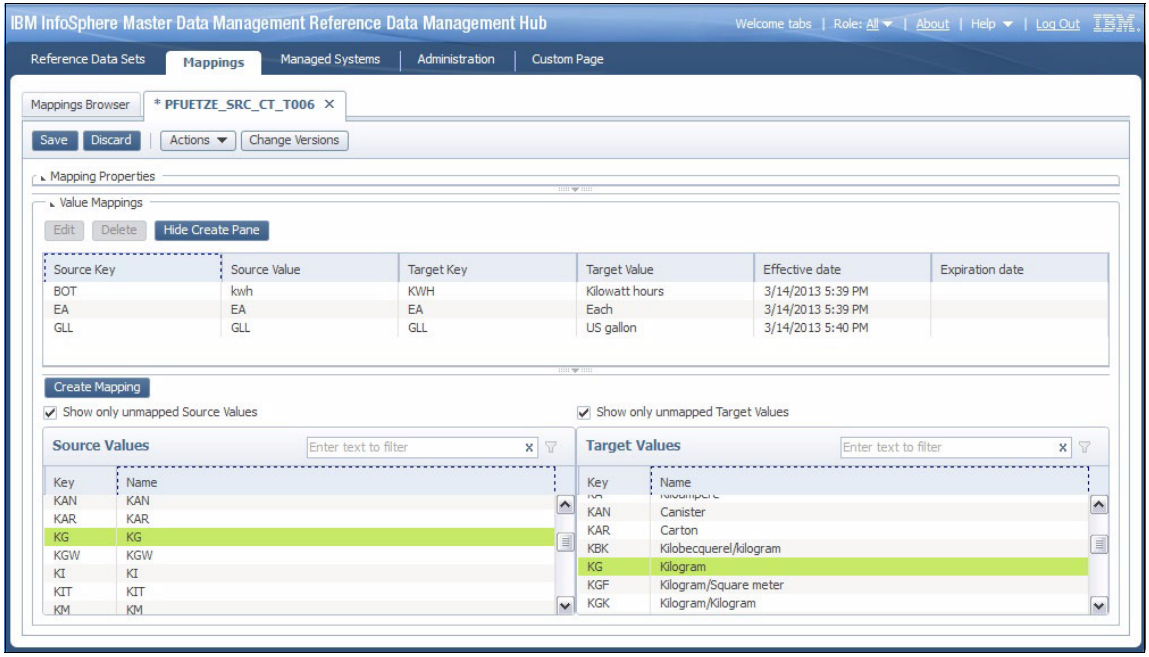
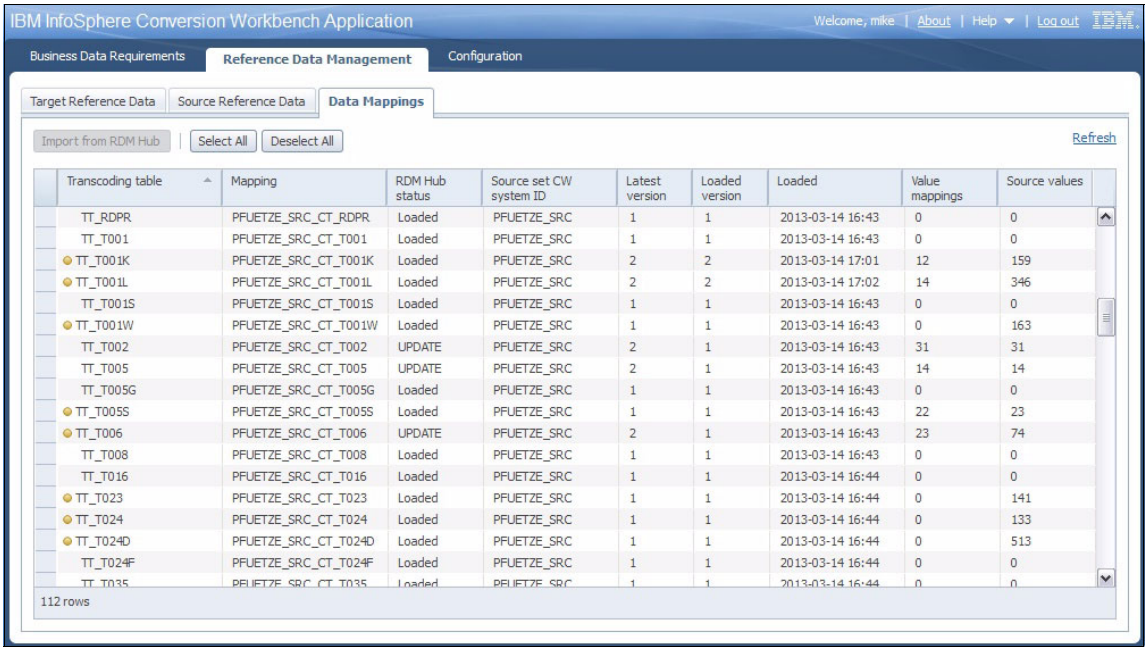


Figure 6-34 FDA defines value mappings with InfoSphere MDM Ref DM Hub

3. After the value mappings are defined in InfoSphere MDM Ref DM Hub, the FDA uses the Conversion Workbench Application to import the InfoSphere MDM Ref DM Hub mappings into the Conversion Workbench transcoding tables (Figure 6-35).



Transcoding table	Mapping	RDM Hub status	Source set CW system ID	Latest version	Loaded version	Loaded	Value mappings	Source values
TT_RDPR	PFUETZE_SRC_CT_RDPR	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	0
TT_T001	PFUETZE_SRC_CT_T001	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	0
TT_T001K	PFUETZE_SRC_CT_T001K	Loaded	PFUETZE_SRC	2	2	2013-03-14 17:01	12	159
TT_T001L	PFUETZE_SRC_CT_T001L	Loaded	PFUETZE_SRC	2	2	2013-03-14 17:02	14	346
TT_T001S	PFUETZE_SRC_CT_T001S	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	0
TT_T001W	PFUETZE_SRC_CT_T001W	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	163
TT_T002	PFUETZE_SRC_CT_T002	UPDATE	PFUETZE_SRC	2	1	2013-03-14 16:43	31	31
TT_T005	PFUETZE_SRC_CT_T005	UPDATE	PFUETZE_SRC	2	1	2013-03-14 16:43	14	14
TT_T005G	PFUETZE_SRC_CT_T005G	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	0
TT_T005S	PFUETZE_SRC_CT_T005S	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	22	23
TT_T006	PFUETZE_SRC_CT_T006	UPDATE	PFUETZE_SRC	2	1	2013-03-14 16:43	23	74
TT_T008	PFUETZE_SRC_CT_T008	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:43	0	0
TT_T016	PFUETZE_SRC_CT_T016	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	0
TT_T023	PFUETZE_SRC_CT_T023	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	141
TT_T024	PFUETZE_SRC_CT_T024	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	133
TT_T024D	PFUETZE_SRC_CT_T024D	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	513
TT_T024F	PFUETZE_SRC_CT_T024F	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	0
TT_T035	PFUETZE_SRC_CT_T035	Loaded	PFUETZE_SRC	1	1	2013-03-14 16:44	0	0

Figure 6-35 FDA imports InfoSphere MDM Ref DM Hub value mappings into transcoding tables

4. The Application Developer generates the data movement ETL jobs that make use of the populated transcoding tables. The data movement jobs are generated with the Conversion Workbench Rapid Generator tool and IBM InfoSphere DataStage®.

Figure 6-36 shows a generated data movement ETL job.

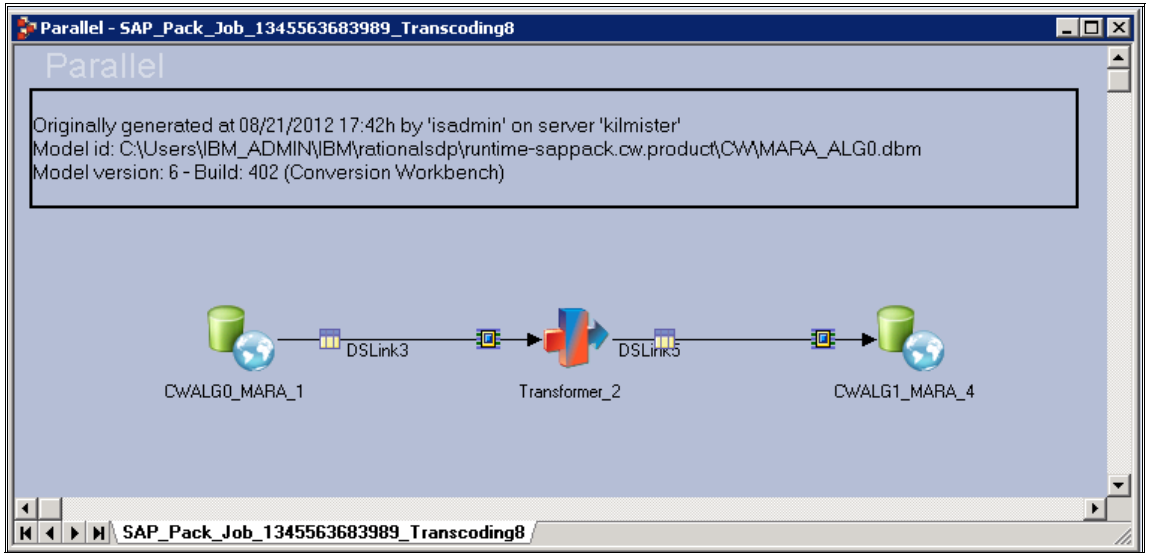


Figure 6-36 Data movement job using transcoding tables

6.6 Workflow

A key aspect of a typical InfoSphere MDM Ref DM Hub implementation is imposing standard governance and procedural guidelines for seamless management of reference data. To that effect, an essential goal becomes having a well laid-out process workflow that can determine and manage the steps, tasks, assignments, and operations to enable smooth transition of reference data from assignment to approval, and eventually distribution. This section introduces several mechanisms to achieve this gaol in InfoSphere MDM Ref DM Hub.

6.6.1 Basic workflow using lifecycle processes

Every resource (reference data set, mapping, or hierarchy) has an associated lifecycle process that determines the states that the particular resource can go through. This lifecycle is used by data stewards to control specific versions of the resources that are in use. Although InfoSphere MDM Ref DM Hub provides a way to define your own custom lifecycle processes, it also provides a set of ready-to use lifecycle process definitions, each providing a different set of states.

InfoSphere MDM Ref DM Hub provides the following lifecycle process definitions:

- Simple Approval process
- State Machine - 2
- Active Editable
- Two Step Approval

See “Lifecycles and states” on page 22 for more detail of these descriptions.

Also see the details in the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/index.jsp?topic=%2Fcom.ibm.swg.im.mdmhs.rdm.usage.doc%2Ftopics%2FLifecyclesAndStates.html

6.6.2 Advanced workflow using integration with external tooling

Although the InfoSphere MDM Ref DM Hub lifecycle processes provide a set of states that can be assigned to resources, in a business process (such as an approval business process), you usually have multiple participants (people and systems) changing the state of resources and sharing information through various transport channels (email, web portal, and so on). To realize these scenarios, you need a workflow engine such as IBM Business Process Manager that provides the means to interact with these people and systems (for instance, with worklists and notification capabilities). In addition to design and execution, a workflow engine (like BPM) also provides the means to simulate and determine the costs of a workflow.

6.6.3 Reference data workflow example

Figure 6-37 on page 157 illustrates a business process combining the BPM and InfoSphere MDM Ref DM Hub lifecycle capabilities. The overall workflow and user-data steward interaction can be realized with BPM. Some of the process steps require that the data stewards interact with InfoSphere MDM Ref DM Hub to modify sets and change the resource state using the Simple Approval Lifecycle associated with the set.

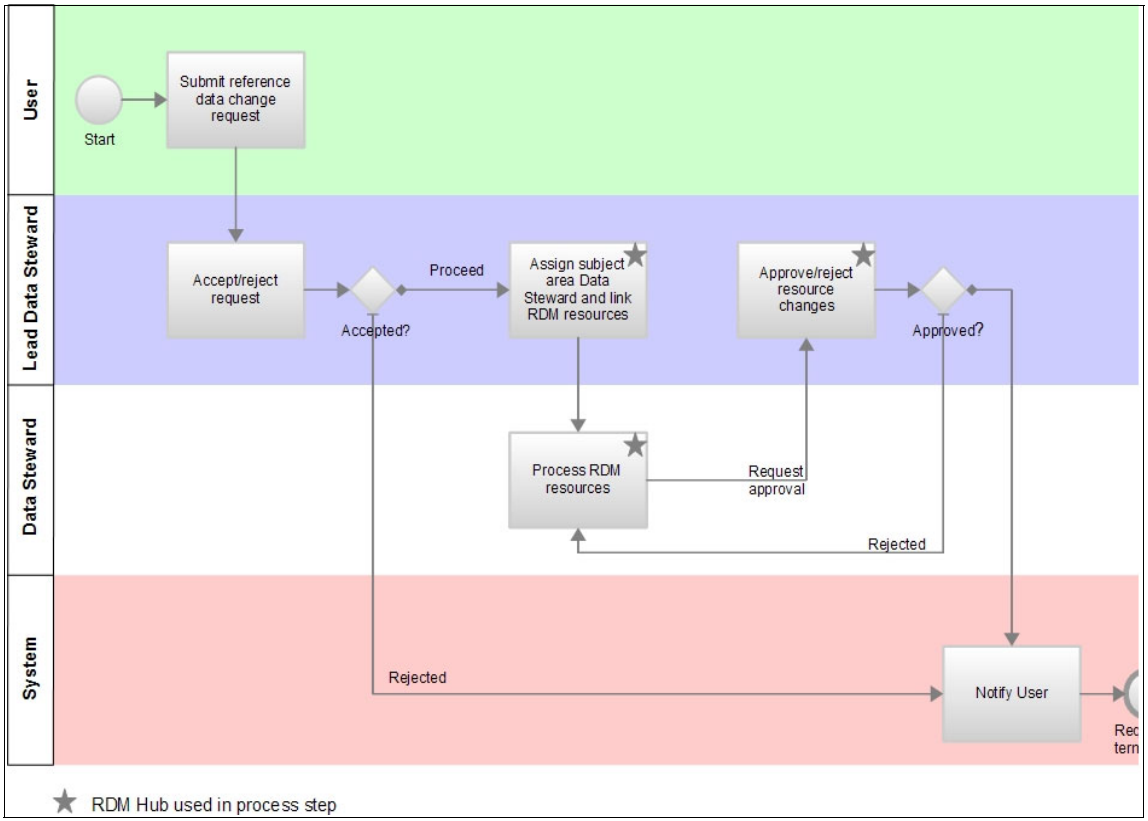


Figure 6-37 Business process

In this example scenario, a business user starts the process by submitting a reference data change request on a web form. This request might be for a new code that represents a product type. The request is added to the worklist of the lead data steward who verifies the information that is provided through the web form and, based on this information, either rejects the request or proceeds processing the request. Based on the provided information, the lead data steward assigns a data steward who is responsible for the requested subject area (for example, product area) to the request and, optionally, also links InfoSphere MDM Ref DM Hub sets and mappings to be processed. The data steward receives the request in his worklist, applies the necessary changes, and sets the state of the modified sets and mappings to Pending Approval. The approval request is added to the worklist of the lead data steward who then verifies the changes and sets the state of the modified sets and mappings to Approved or Rejected. The user is notified if the submitted change request is rejected or the change has been approved.

7

Implementation

After analyzing your reference data and completing the data modeling, you can proceed to implement the reference data model. This chapter describes the process of implementing reference data with IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub). The chapter includes the following topics:

- Installation
- Configuration
- Implementing data resources using InfoSphere MDM Ref DM Hub Client web user interface (UI), batch jobs, and application programming interfaces (APIs)
- Customization

7.1 Installation

InfoSphere MDM Ref DM Hub has two types of entitlements, IBM InfoSphere MDM Reference Data Management Hub license and Custom Data Hub license. Both have the same InfoSphere MDM Ref DM Hub contents.

Installing InfoSphere MDM Ref DM Hub is a multi-step process that includes the following tasks:

1. Prepare for installation:
   a. Check the specific hardware and software requirements that must be met to properly install and run the InfoSphere MDM Ref DM Hub.
   b. install the prerequisite applications, including the InfoSphere MDM Custom Domain Hub.
2. Install InfoSphere MDM Ref DM Hub:
   a. Install the database entities.
   b. Install the web application.

   You can use IBM Installation or run command-line processes to install the product.
3. Configure security and authorization in the web application server.

   InfoSphere MDM Ref DM Hub uses a role-based security concept. Activities and states entities are related to roles, which are linked to groups that you define in the application server. Users are members of these groups.
4. Configure the components.

   After installation is finished, you can configure additional components:

   – Adjusting transaction time-outs
   – Configuring batch export
   – Configuring application defaults
   – Customizing the custom page tab

This section demonstrates the InfoSphere MDM Ref DM Hub installation with IBM Installation Manager in a Windows environment.

For detailed installation and configuration information, see the information center:

http://pic.dhe.ibm.com/infocenter/mdm/v10r0m0/index.jsp?topic=%2Fcom.ibm.swg.im.mdmhs.rdm.install.doc%2Ftopics%2FTechnicalRequirements.html

7.1.1 Product download

You can download InfoSphere MDM Reference Data Management Hub V10.1.0 though the IBM Passport Advantage® website by using the following information:

- Part number: CIB2UML
- Product name: IBM InfoSphere Master Data Management Reference Data Management Hub V10.1.0 Multiplatform Multilingual
- File name: `IS_MDM_RDMH_V10.1_MLTP_ML.tar.gz`

7.1.2 Installation with IBM Installation Manager

InfoSphere MDM Ref DM Hub installation with IBM Installation Manager can accomplish two tasks:

- Reading and accepting the license agreement during the installation stage.
- Installing application files to a target directory.

If you do not have IBM Installation Manager, you can download it:

`http://www.ibm.com/support/docview.wss?uid=swg24030152`

Complete the following steps to install InfoSphere MDM Ref DM Hub to a target file system:

1. Extract `IS_MDM_RDMH_V10.1_MLTP_ML.tar.gz` file to a temporary directory on your local file system, as in the following example:

   ```
   C:\temp
   ```

2. Extract the `RDMRepos.zip` file to a temporary directory, as in the following example:

   ```
   C:\temp\RDMINST-V10.1.0-2012-08-16
   ```

3. Open IBM installation manager and configure the installation manager to point to the InfoSphere MDM Ref DM Hub repository configuration file, as in the following example:

   ```
   C:\Projects\IM\RDM\RDMINST-V10.1.0-2012-08-16\output\repository.config
   ```

4. Click **Install** to install InfoSphere MDM Ref DM Hub and provide an installation path for the installation. We use `C:\Tools` as the path.

In a successful installation, you see the folders and files (shown in Figure 7-1), which are created under the following path:

`C:\Tools\IBM\InfoSphere MDM Reference Data Management`

Figure 7-1 Extracted files and folders by the installation manager

The folders and files are as follows:

- `BatchExport` folder contains all files that are related to the InfoSphere MDM Ref DM Hub Batch Export.
- `License` folder contains all InfoSphere MDM Ref DM Hub supported language licenses and notices files.
- `Scripts` folder contains all SQL script files for setting up InfoSphere MDM Ref DM Hub database and the rollback scripts. InfoSphere MDM Ref DM Hub supports both DB2 and Oracle databases and the scripts files are similar for these two database systems. Figure 7-2 shows the DB2 SQL file names.

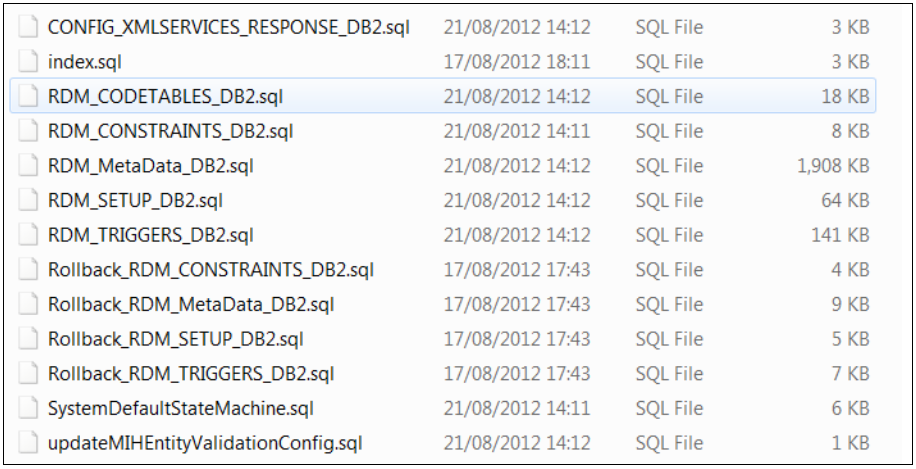


Figure 7-2 DB2 SQL script files

- MIH101-App.ear file is the InfoSphere MDM Ref DM Hub server component.
- RDMClientEAR.ear file is the InfoSphere MDM Ref DM Hub client component.
- README_DB2.txt and README_Oracle.txt files include the instructions on which SQL scripts must run for a fresh InfoSphere MDM Ref DM Hub V10.1.0 installation.

You must manually install both the InfoSphere MDM Ref DM Hub server component and the client component.

7.1.3 InfoSphere MDM Ref DM Hub server installation

Before installing the InfoSphere MDM Ref DM Hub server component, you must run the SQL setup scripts to install the database entities. The scripts are in the following directory, where <InstallDir> is where the InfoSphere MDM Ref DM Hub is installed:

<InstallDir>/scripts/sql/<db_type>

Follow the instructions in the readme file for the database type:

- DB2: <InstallDir>/README_DB2.txt
- Oracle: <InstallDir>/README_Oracle.txt

When running the database scripts, if it is a fresh installation, you do not need to run any of the Rollback scripts, just the scripts other than roll back are sufficient. Example 7-1shows the command to run DB2 script to set up DB2 tables.

Example 7-1 Example of set up InfoSphere MDM Ref DM Hub tables for DB2

```
db2 -tvf RDM_SETUP_DB2.sql
db2 -v -td@ -f RDM_TRIGGERS_DB2.sql
db2 -tvf RDM_CONSTRAINTS_DB2.sql
db2 -tvf RDM_MetaData_DB2.sql
db2 -tvf RDM_CODETABLES_DB2.sql
db2 -tvf CONFIG_XMLSERVICES_RESPONSE_DB2.sql
db2 -tvf index.sql
db2 -tvf SystemDefaultStateMachine.sql
db2 -tvf updateMIHEntityValidationConfig.sql
```

In the scripts in Example 7-1, the default schema is db2admin. It must be replaced accordingly if the schema name that is used differs.

The MIH-App.ear file is the InfoSphere MDM Ref DM Hub server component. Before installing the MIH-App.ear to the server, you must modify the EAR file with certain configuration changes, and turn off the Entitlement Engine.

Follow the preparation instructions, which are available at the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.install.doc/topics/PreparingToInstallRDM.html

After you complete the preparation tasks, use the following steps to install the `MIH-App.ear` file:

1. Make sure WebSphere Application Server is running.
2. From WebSphere Application Server, using the administration console, select **Applications** → **New Application** → **New Enterprise Applications**.
3. Click **Install** and browse to the `MIH-App.ear` file.
4. Select the **Fast Path installation** option, accept the defaults, and click **Finish**.

7.1.4 RDM client installation

InfoSphere MDM Ref DM Hub client runs on a WebSphere Application Server. Complete the following steps to install InfoSphere MDM Ref DM Hub client:

- Make sure WebSphere Application Server is running.
- From WebSphere Application Server using administration console, choose **Applications** → **New Application** → **New Enterprise Applications**.
- Click **Install** and browse to the `RDMClientEAR.ear` file.
- Select the **Fast Path installation** option, accept the defaults, and click **Finish**.

If InfoSphere MDM Ref DM Hub is not installed with the default port (9080), the configuration of the web service clients must be changed. You can find the configuration in the following location:

`RDMClientEAR/RestAPI.war/WEB-INF/classes/client.properties`

If the correct host aliases are not defined in WebSphere Application Server, you might see an error message when you log in to InfoSphere MDM Ref DM Hub; the message states that virtual hosts are not defined.

One solution you can try is as follows:

1. Click **Servers** → **WebSphere application servers** → **<server_name>** → **Ports**.
2. Note the values for the following items:
   - SIP_DEFAULTHOST
   - SIP_DEFAULTHOST_SECURE
   - WC_defaulthost
   - WC_defaulthost_secure
3. Click **Environment** → **Virtual Hosts** → **<default_host>** → **Host Aliases**.
4. Create new aliases based on the four values you noted.

7.2 Configuration

This section describes InfoSphere MDM Ref DM Hub configuration tasks regarding the following items:

- Security and authorization
- Lifecycle and state machine
- Regular expression validation

7.2.1 Configure security and authorization

InfoSphere MDM Ref DM Hub uses a role-based security model. Activities and state entities are related to roles, which are linked to groups that are defined in the application server. Users are members of these groups.

To log in to the InfoSphere MDM Ref DM Hub web user interface (UI) application, you must enable WebSphere Application Server global security, create users and groups, and then associate users to groups to enable them to performing certain activities.

This section shows how to do the following tasks:

- Configure the web application server security.
- Create groups on WebSphere Application Server.
- Create owner groups in InfoSphere MDM Ref DM Hub.

- Create users in WebSphere Application Server or LDAP and assign them to groups.
- Add roles, if needed.
- Associate the roles with the WebSphere or LDAP groups.

Configuring WebSphere Application Server and application security

Configure WebSphere Application Server global security as follows:

1. Start and log in the WebSphere Application Server administrative console.
2. Browse to **Security** → **Global Security**.
3. In the User Account Repository, select **Federated Repositories**.
4. Click **Configure**.
5. In the General Properties section of Federated Repositories, select the following options:
   - **Automatically generated server identity**
   - **Ignore case for authorization**
6. On the Administrative User Password page, provide a password for your administrative user. This user is the one to access the administrative console.
7. On the Global Security page, click **Save**.

Figure 7-3 shows the result of this configuration. You see that both administrative security and application security are enabled.

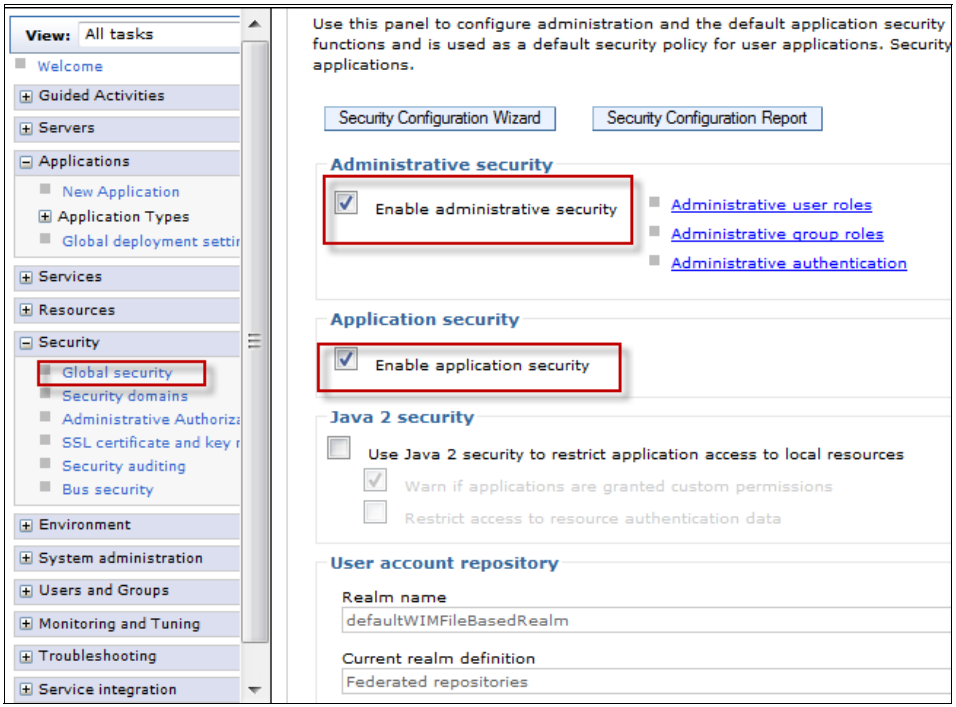


Figure 7-3 Configured application security

Creating groups in the WebSphere Application Server

The `RDMRole_*` groups are used for the user interface. The group MDM is associated with the service provider and consumer role. If you want to access server-side functions, you must become a member of the MDM group.

Complete the following steps to create a group:

1. Start the WebSphere Application Server administrative console and log in with a user name and password that you set up for your WebSphere Application Server administrative security.
2. Browse to **Users and Groups** → **Manage Groups**.
3. Click **Search** to show the existing groups.

4. In the Manage Groups section, click **Create**.
5. On the Create a Group page, add the following groups. Click **Create** after adding each group.
   - RDMRole_Administrators
   - RDMRole_All
   - RDMRole_Approvers
   - RDMRole_Approvers2
   - RDMRole_Custom
   - RDMRole_Integrators
   - RDMRole_Stewards
   - crm
   - enterprise
   - mdm

Creating and assigning users

You create users and assign them to groups to give them access to the functions through WebSphere Application Server.

Complete the following steps to create and assign users:

1. Log in the WebSphere Application Server administration console.
1. Expand the Users and Groups menu and select **Manage Users**.
2. Click **Search** to show all current users.
3. On the Manage Users panel, click **Create**.
4. In the Create a User page, type a User ID as listed in Table 7-1 on page 169.
5. Click **Group Membership** and add the group to which you want each user to belong. See the examples in Table 7-1 on page 169.
6. Provide a password for the user.

Table 7-1 lists users and groups, supported by InfoSphere MDM Ref DM Hub.

Table 7-1 Example of system users to add

User ID	Group	Password
admin	RDMRole_Administrators, enterprise	passw0rd
approver	RDMRole_Approvers, enterprise	passw0rd
approver2	RDMRole_Approvers2,mdm	passw0rd
integrator	RDMRole_Integrators, enterprise	passw0rd
mdm	RDMRole_All	passw0rd
mih	None	passw0rd
steward	RDMRole_Stewards,enterprise	passw0rd
steward2	RDMRole_Stewards,mdm	passw0rd
super	RDMRole_Administrators, RDMRole_Approvers, RDMRole_Integrators, RDMRole_Stewards, crm, enterprise, mdm	passw0rd
tabs	RDMRole_All, crm, enterprise, mdm	passw0rd

For more details about InfoSphere MDM Ref DM Hub security, see the “Creating roles” topic in the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.install.doc/topics/CreatingRoles.html

To associate groups and roles, see the following page in the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.install.doc/topics/AssociatingGroupsAndRoles.html

7.2.2 Lifecycle and state machine

Reference data lifecycle management is about managing the states of a reference data set throughout its lifecycle: from creation and initial storage to the time when it becomes obsolete and is deleted. The core InfoSphere MDM Ref DM Hub objects can have a well-defined lifecycle that specifies the governance over the state of a reference data object.

The lifecycle management is supported with the use of a combination of property files and data base table configuration. For the details of lifecycles and their possible states, see the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.usage.doc/topics/LifecyclesAndStates.html?resultof=%22lifecycle%22%20%22lifecycl%22

For the details of the various database tables involved in the governance of the lifecycle of an InfoSphere MDM Ref DM Hub object, see the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.install.doc/topics/CustomizingLifecycles.html?resultof=%22lifecycle%22%20%22lifecycl%22

It also has an example of the steps to follow to create a custom lifecycle process.

7.2.3 Regular expression validation

Regular expression validation checks a property value against a validation rule. If the value does not match the validation rule, an error is displayed on the widget in the InfoSphere MDM Ref DM Hub UI.

A validation rule is defined for a string type of a property. In InfoSphere MDM Ref DM Hub, you can define a validation rule for the following properties:

- The Code property for a reference data set
- The Custom properties with a string type for a reference data set for both set level and value level.

You can define a validation rule when you define or edit a reference data type or a custom property. The validation rule is an optional field so you can place a constraint on some set values.

Validation expression syntax is based on the Java regular expression syntax. InfoSphere MDM Ref DM Hub matches only when the entire input string explicitly matches.

Table 7-2 lists the characters for validation expression.

Table 7-2 List of characters for the regular expression

Characters	Meaning of the character
\	Escape character. Forces any special character to be a normal character.
[	Signifies the beginning of a character class.
]	Signifies the end of a character class.
\d	Any digit and equivalent to [0-9].
\D	Any non-digit.
\s	White space characters.
\S	Non-whitespace characters.
^	Signifies not. For example, [^A-Z] matches a character if it is not a capital letter.
?	Matches zero or one time.
*	Matches zero or more times.
+	Matches one or more times.
{n}	Matches n times.
{n,}	Matches n or more times.
{n, m}	Matches between n and m times, inclusive.
\w	Word characters and equivalent to [A-Za-z0-9].

Example 7-2 through Example 7-5 on page 172 show rules.

Example 7-2 Code format

```
Rule: The code for any entered value must be two letters followed by three
digits. Validation rule:

[A-Za-z]{2}[\d]{3}

valid value: Abc123
InValid value:adcd12
```

Example 7-3 Alphanumeric code

```
Rule: Any combination of numbers and letters, with a length of at least 1.
Validation rule:

[A-Za-z0-9]+
```

Example 7-4 Part number

```
Rule: A six to eight digit number, optionally followed by a hyphen and another
five digit number. Validation rule:

[\d]{6,8}(-\d{5}])?
```

Example 7-5 Email address

```
Rule: An alphanumeric handle followed by an @ symbol, followed by the domain
name. Validation rule:

\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z]{2,4}\b

\b The beginning of a word boundary.
[A-Aa-z0-9._%+-]+ Matches any combination of uppercase and lowercase letters,
digits, period, underscore, percent sign, plus sign, or minus sign characters.
@ The next character must be an at sign(@).
[A-Za-z0-9.-]+ Matches any combination of uppercase and lowercase letters,
digits, period and minus sign characters.
\. Literally matches the peirod character.
[A-Za-z]{2,4} The next 2-4 characters must be letters.
\b The end of a word boundary.
```

7.3 Implementing InfoSphere MDM Ref DM Hub model

This section shows how to create reference data types, sets, and mappings, by using the import web user interface, and the MDM Ref DM Hub web service API.

The section then demonstrate how to distribute this data to external applications or other file systems.

7.3.1 Creating reference data sets

With InfoSphere MDM Reference Data Management Hub web UI, you can group your reference data sets in folders. You can have subfolders in a folder. Each reference data set is based on a reference data type that can be a default data

type or a custom data type. If you create a reference data set with custom properties, you must define the custom properties in a data type.

This section shows you how to do the following tasks:

- Create a default reference data set.
- Define a data type that has a reference data set property.
- Create a reference data set that has a related reference data set.
- Create a reference data set value manually.
- Create multiple reference data set values by importing a file.

In the example, we create a root folder named `Chap08`.

A default reference data type has a set of core properties including Code and Name. You must log into InfoSphere MDM Ref DM Hub and select a role that permits that access to the Administrator tab.

Creating a default reference data set

Complete the following steps to create a reference data set with a default data type in Web UI:

1. Select the **Chap08** folder.
2. Create a new set by either clicking **New Set** or right-clicking to open the context menu and then selecting **New** → **Set**. See Figure 7-4 on page 174.

   The Create a Set window opens, with the values of the following properties:

   – Name: CountryCode
   – Version: 1, by default (You can change it to another version.)
   – Type: Default Reference Data Type
   – Lifecycle Process: Simple Approval Process
   – Effective Date: current date and time, by default

3. Click **OK** to create a new set.

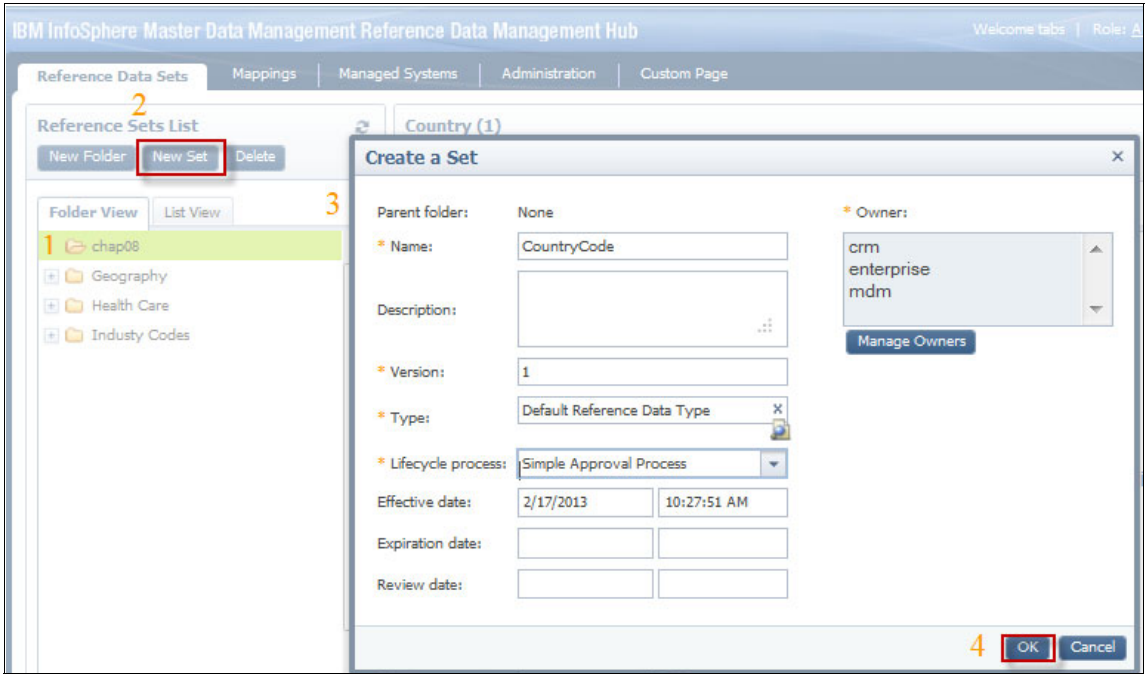


Figure 7-4 Create a reference data set

The new set named CountryCode is created under the `chap08` folder and an empty set in the set view is displayed on the right panel next to the Reference Sets List.

After a reference data set is created, you can either manually create reference values or import reference data set values from a comma-separated values (CSV) file or an XML file.

Creating a reference data set value manually

You can manually create a reference data set by using the following steps:

1. Click **New** under the Reference Value as shown in Figure 7-5.

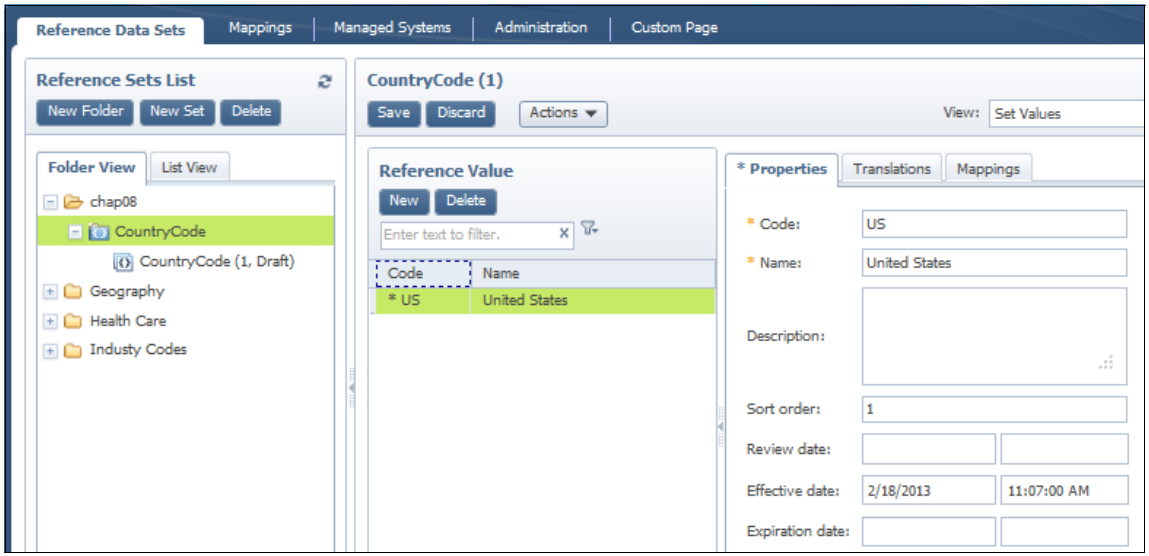


Figure 7-5 Create a reference data set value

2. Tabbed panels are displayed on the right pane, under the View, for creating a reference data set value. Enter **US** for the Code property; enter **United States** for the Name property.
3. Click **Save** under the CountryCode(1) to save the properties.
4. A new reference value is created, which you can see under the Reference Value pane list, as in Figure 7-6.

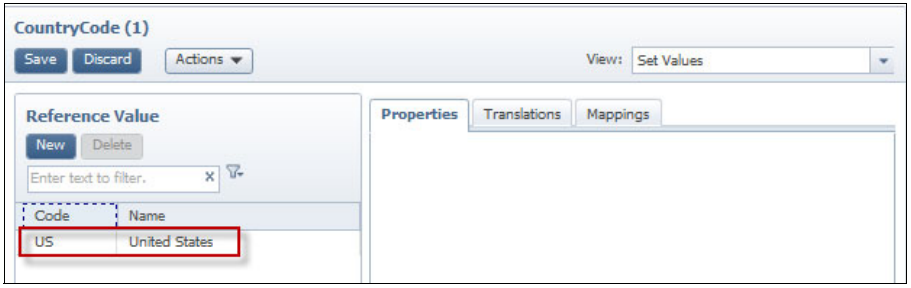


Figure 7-6 A newly created reference data set value

You can also edit or delete a reference data set value manually.

Creating reference data set values by importing a CSV file

The best practice for creating a large number of reference data set values is to import a CSV or an XML file into a pre-created empty set.

Before you import a CSV file, look at the column names and values that are in the CSV file. The first row of the CSV file lists the column names that are mapped to the properties of a reference data set in InfoSphere MDM Ref DM Hub. Figure 7-7 shows a snippet of our sample CSV file with two column names: ISO Country Code and Name. They are mapped to the Code and Name of a reference data set during importing of a CSV file. This task is done through the import wizard.

	A	B
1	ISO Country Code	Name
2	AU	Australia
3	AT	Austria
4	BE	Belgium
5	CA	Canada
6	CZ	Czech Republic
7	DK	Denmark
8	FI	Finland
9	FR	France
10	DE	Germany
11	GR	Greece
12	HU	Hungary

Figure 7-7 CSV file

Complete the following steps to import reference data set values:

1. Right-click the CountryCode set in the Folder View to open a context menu, as shown in Figure 7-8. Click **Import** to import a reference data set.

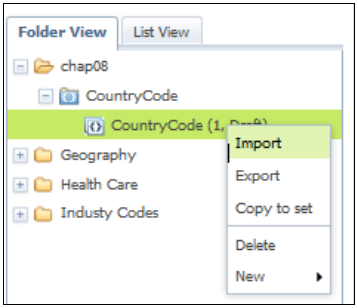


Figure 7-8 A set context menu: Right-click the CountryCode

2. Browse to the `CountryCode.csv` file and click **Next** (Figure 7-9).

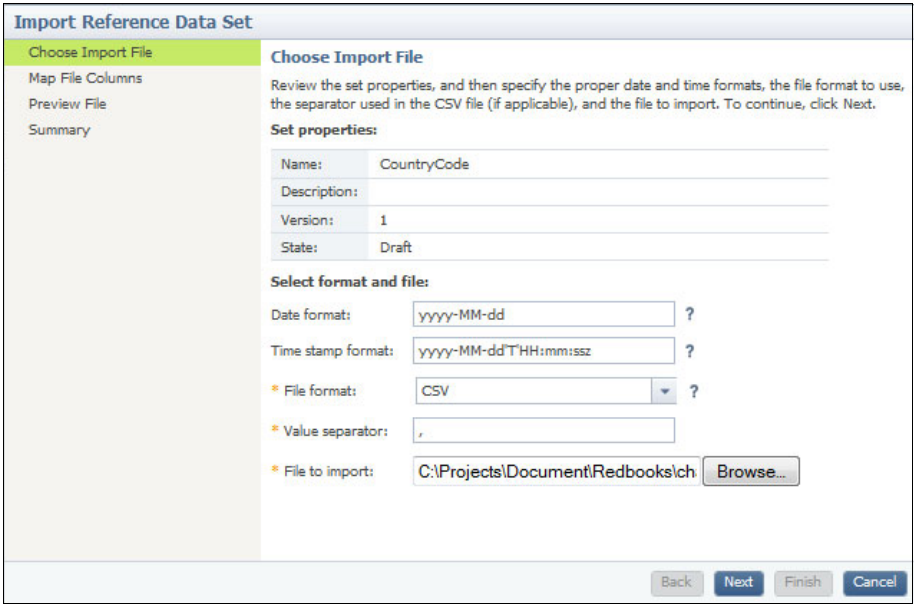


Figure 7-9 Choose import file wizard

3. Both Code and Name properties are mapped automatically in Map File Columns pane (Figure 7-10).

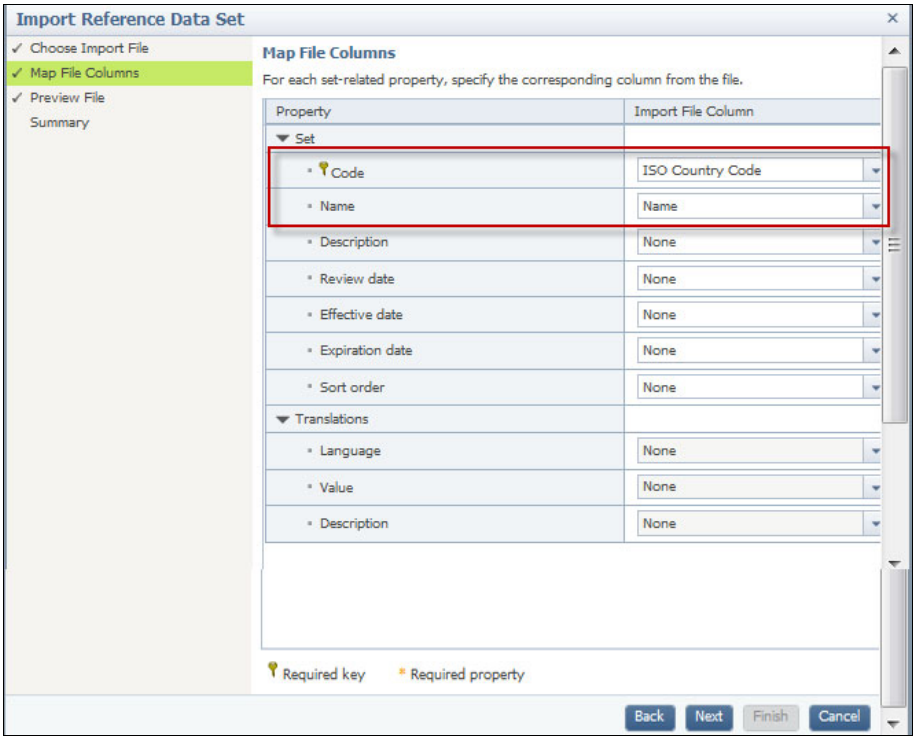


Figure 7-10 Map file columns

In the Preview File pane(Figure 7-11), a list of reference data set values are displayed for preview.

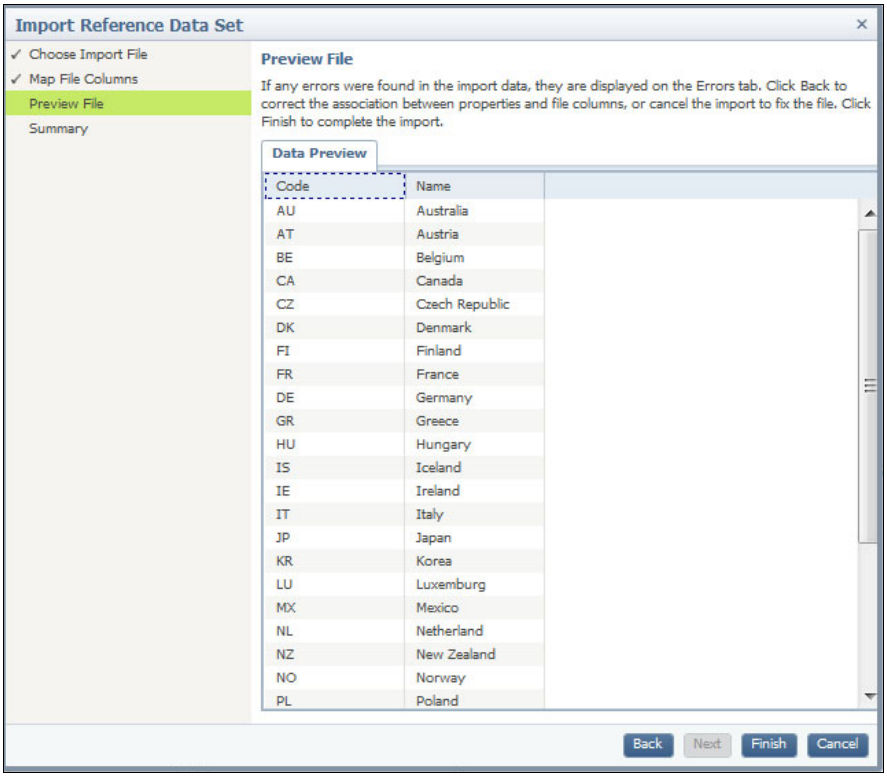


Figure 7-11 Preview the CSV file content

4. In the Summary panel (Figure 7-12), the wizard displays the number of values imported. Any row that failed to be imported is displayed with an error message. Review the error message carefully and fix the problem in the CSV file and import it again.

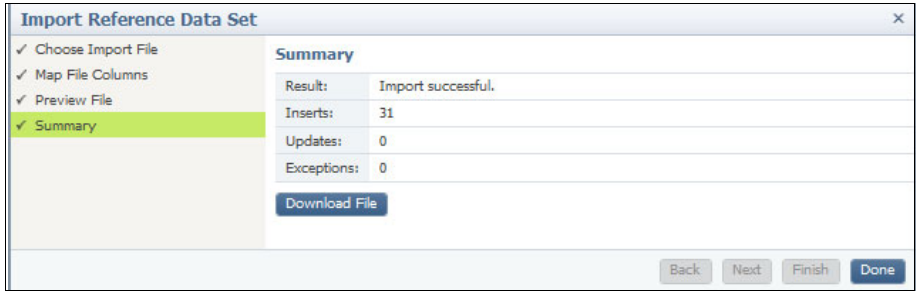


Figure 7-12 Summary of importing a CSV file

Tip: If any date or time values are in the CSV file, be sure that the value matches the date and time format that is specified in the import file wizard. Any mismatch results in an error and the entire row will be skipped.

Figure 7-13 shows that the reference data set values are imported.

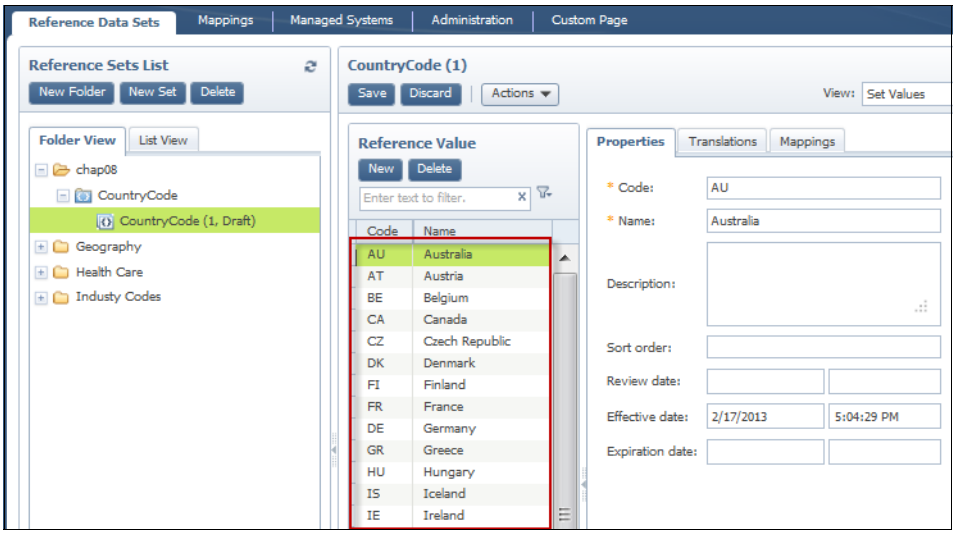


Figure 7-13 Reference data set values by importing a CSV file

Importing an XML file is similar to importing a CSV file. Select an XML file format in the Choose Import File step (Figure 7-9 on page 177). Then, follow the import wizard to complete the import steps.

Creating a reference data set with custom properties

This section describes how to create a reference data set with custom properties. At the data analysis stage, you must decide whether your data needs to be modeled as a custom data type. In general, if your CSV file, XML file, or a database table contains extra columns other than those found in the default properties of InfoSphere MDM Ref DM Hub, you must consider whether to include the columns. If they are to be included. then a custom data type must be defined and created for each column.

Defining a custom data type

With InfoSphere MDM Ref DM Hub, you can define custom types in the reference data set type. When you create a custom data type that has a reference data set property, you must ensure that the reference data set was created and approved in InfoSphere MDM Ref DM Hub. You must log in to

InfoSphere MDM Ref DM Hub with an account that has permission to create a data type with custom properties.

This section uses a pre-created reference data set named Country in the Geography folder and this set is in an approved status. Complete the following steps to create a reference data set type in InfoSphere MDM Ref DM Hub:

1. On the Administration tab, click **New**. The Create Data Type dialog box opens. See Figure 7-14.

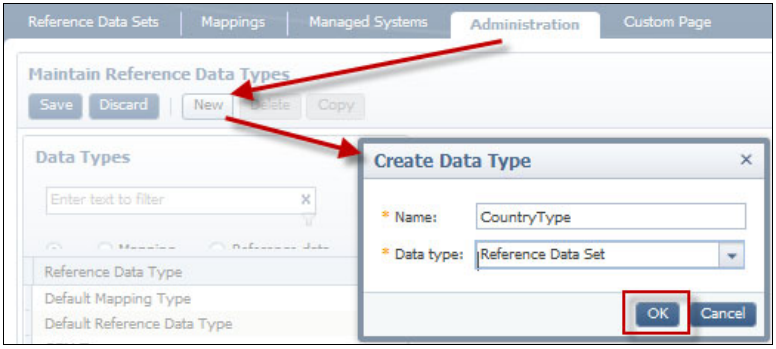


Figure 7-14 Create a data type

2. Enter the following information:
   - In the Name file, type `CountryType`.
   - In the Data types field, select **Reference Data Set**.
   - Click **OK**.

3. Both Value Level and Set Level are created as shown in Figure 7-15. Select **Value Level** → **New** to define a custom property at Value Level.

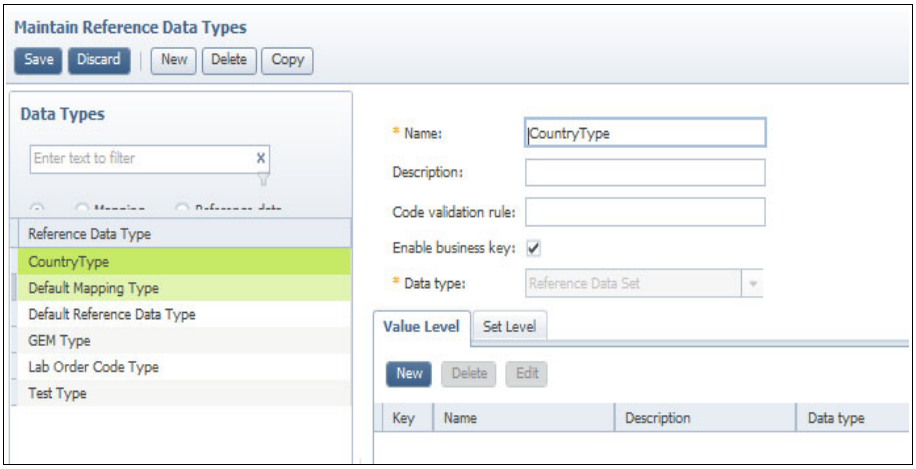


Figure 7-15 An empty CountryCodeType

4. In the Data Type Property window (Figure 7-16), do the following steps:
   a. In the Name field, enter `ISOCountry`.
   b. In the Type field, select **Reference Data Set**.

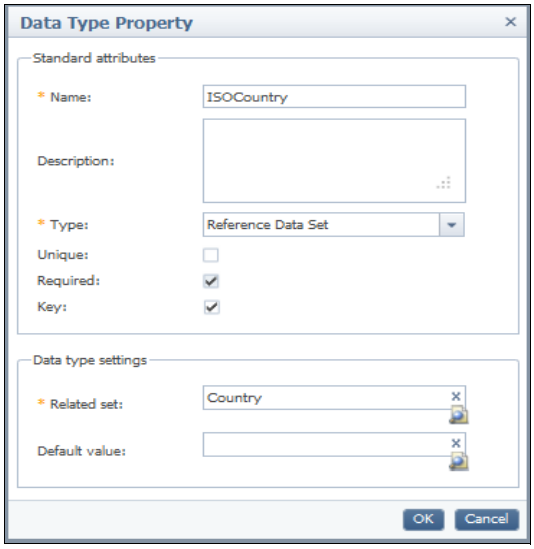


Figure 7-16 Data Type Property window

c. In the Related set field, click the search icon to start a widget, which has a filter.

d. Select a reference data set. See Figure 7-17.

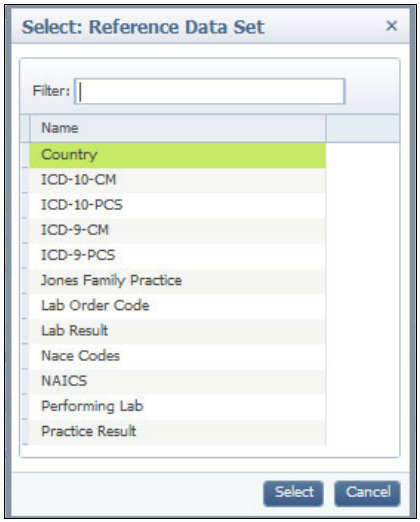


Figure 7-17 Select a reference data set from InfoSphere MDM Ref DM Hub

In the example, we enter **Country** and select it from the list. Click **Select**. The wizard brings you back to the Data type property dialog window.

e. Click **OK** to create the CountryCode property.

Figure 7-18 shows that the custom property is created.

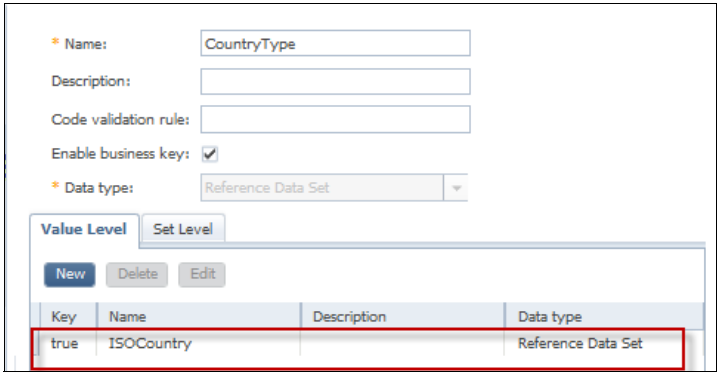


Figure 7-18 A custom property in the defined data type

You can define as many custom properties as you like. A custom property can be any supported data type as shown in Figure 7-19. You can select a custom property to edit or delete before the custom data type is used.

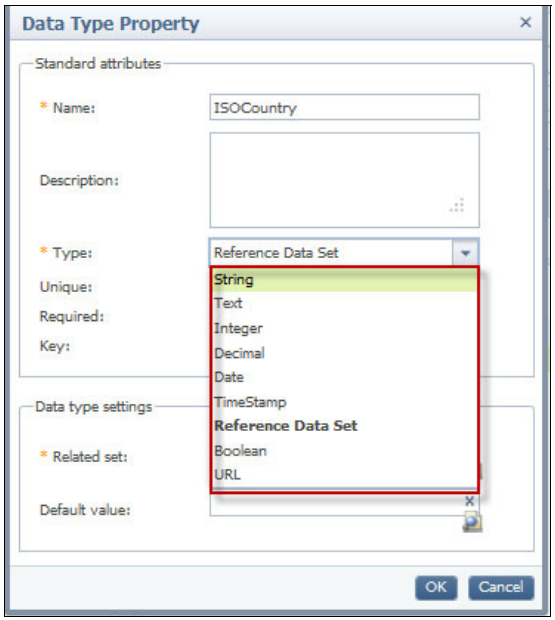


Figure 7-19 RDM supported custom property data types

Creating a reference data set with a custom data type

This section shows how to create a reference data set with a custom property, with an example. This example uses the custom data type, CountryType, that was created in the previous steps.

Use the following steps to create a reference data set with a custom data type:

1. Log in to InfoSphere MDM Ref DM Hub with the user ID and password that has the proper authorization.
2. Right-click the **Chap08** folder and then select **New** → **Set**.
3. Enter the set properties. We created a set by specifying the following information (see Figure 7-20 on page 185):
   - Type `MDMStates` in the Name field.
   - Select **CountryType** by clicking the search icon for the Type.
   - Select **Simple Approval Process** from the Lifecycle process list.
   - Use default values for every other field.

   Click **OK** to create the set.

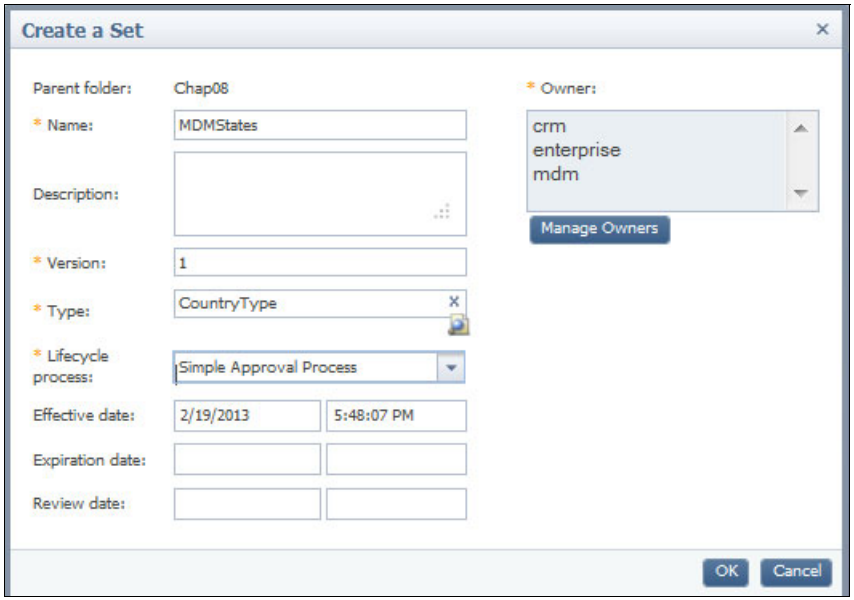


Figure 7-20 Create a Set dialog

After the reference data set is created, we use the import steps described in “Creating reference data set values by importing a CSV file” on page 176 to import values from a CSV file to the new created set MDMStates.

Figure 7-21 shows that the MDMStates set has the value populated.

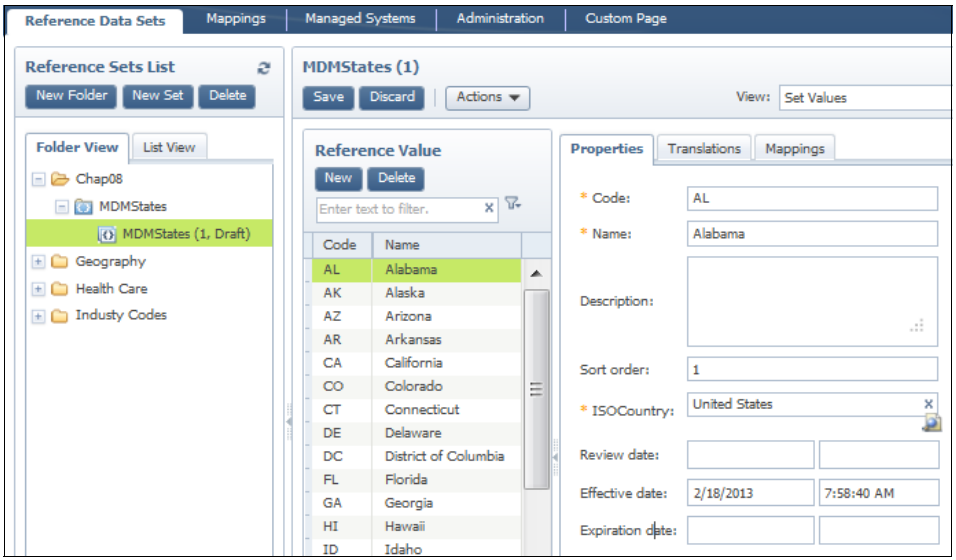


Figure 7-21 Imported MDMStates set values

Creating a set with the InfoSphere MDM Ref DM Hub web service APIs

The InfoSphere MDM Ref DM Hub exposes the `RDMPort.wsdl` file so that the InfoSphere MDM Ref DM Hub Rest services can use the web service APIs to make the web service call. With the InfoSphere MDM Ref DM Hub web service APIs, you can send SOAP messages to the web service.

Example 7-6 shows code for creating new reference data set.

Example 7-6 API to create new reference data set

```
public RDValueSetResponse addRDValueSet(Control control, RDValueSet request)
throws java.rmi.RemoteException, com.ibm.wcc.service.intf.ProcessingException;
public RDValueSetResponse updateRDValueSet(Control control, RDValueSet request)
throws java.rmi.RemoteException, com.ibm.wcc.service.intf.ProcessingException;
```

You must use addRDValueSet to create a new set in InfoSphere MDM Ref DM Hub and you can then use updateRDValueSet to add, update, and delete the set values.

Example 7-7 shows a code snippet for creating a new set.

Example 7-7 Creating a new set

```
// 1. create RDMService port type
RDMService_PortType RDMService = new ServiceFatory().newRDMService();

// 2. create a Control object from the HttpRequest object,set reqester name and
role and the paging information etc.
Control control = getNewControl(HttpServletRequest request);

//3. get a set type from a set type ID and get a set state machine from initial
//type, such as Default Set type and Simple Process Approval
RDValueType type = null;
RDStateMachine stateMachine = null;
try {
type = RDMService.getRDValueType(control, typeID,1).getResponse();
stateMachine = RDMService.getRDStateMachine(control,
stateMacineID).getResponse();
} catch (ProcessingException e1){...}

//4. get the set value property Type
RDValuePropertyType[] rdValuePropertyType = null;
try {
rdValuePropertyType = RDMService.getRDValuePropertyTypeByValueTypeID(control,
type.getIdPK().get_value()).getResponse();
} catch (ProcessingException pe) {...}

//5. create a RDValueSet object with an initial RDState
// and a set name
RDValueSet updatedValueSet = new RDValueSet();
updatedValueSet.setRdState(stateMachine.getInitialState());
updatedValueSet.setRdValueSetName("CRMStates");

// 6. make web service call to update or create the value set
try {
updatedValueSet = RDMService.addRDValueSet(control,
updatedValueSet).getResponse();
long newSetID = updatedValueSet.getBaseID();
} catch (ProcessingException ex) {...}
```

Example 7-8 shows a code snippet for adding a set value to an existing set. The `valuesList` is a list of new created set values.

Example 7-8 Adding set values using updateRDValueSet API

```
// setID is the give set ID
RDValueSet set = null;
try {
set = RDMService.getRDValueSet(control, setID, 0).getResponse();
}catch (ProcessingException ex) {...}

RDValue[] modifiedValues =
valuesList.toArray(new RDValue[valuesList.size()]);
set.setRdValue(modifiedValues);
...
RDValueSet updatedSet = RDMService.updateRDValueSet(control,
set).getResponse();
```

7.3.2 Creating mappings

Mappings are designed for maintaining a relationship between two sets that can be similar or different versions of a set, or relationships between two set values. Creating a mapping is similar to creating a set. Each mapping has a type that defines a set of core properties of a mapping. You can define a mapping with a default mapping type or a custom mapping type.

This section shows how to create a mapping, based on the default mapping type, between two sets. In this example, we use a source set MDMStates and a target set CRMStates. See Figure 7-22 on page 189.

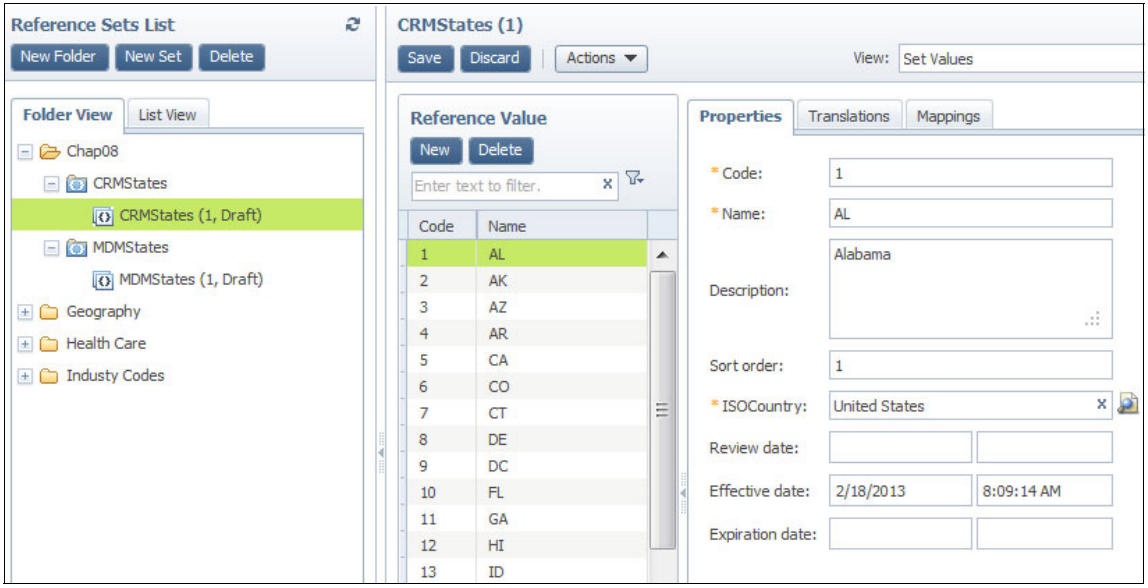


Figure 7-22 CRMStates set

The MDMStates and CRMStates use separate codes and names to represent a Country code and thus need a code mapping between these two sets.

Use the following steps to create a mapping:

1. Log in to InfoSphere MDM Ref DM Hub with a user name and password that has the proper authorization, and click the **Mappings** tab.
2. Click **New** on the Mapping Browser tab to create a new mapping (Figure 7-23 on page 190). Enter the following data for the required fields:
   - Source set: Type `MDM` in the filter widget and select **MDMStates(1, Draft)**.
   - Target set: Type `CRM` in the filter widget and select **RDMStates(1, Draft)**.
   - Name: Enter `MDM To CRM States`.
   - Type: Use **Default Mapping Type**.
   - Life cycle process: Select **Simple Approval Process**.

Keep the defaults on all other fields. Click **OK** to create this mapping.

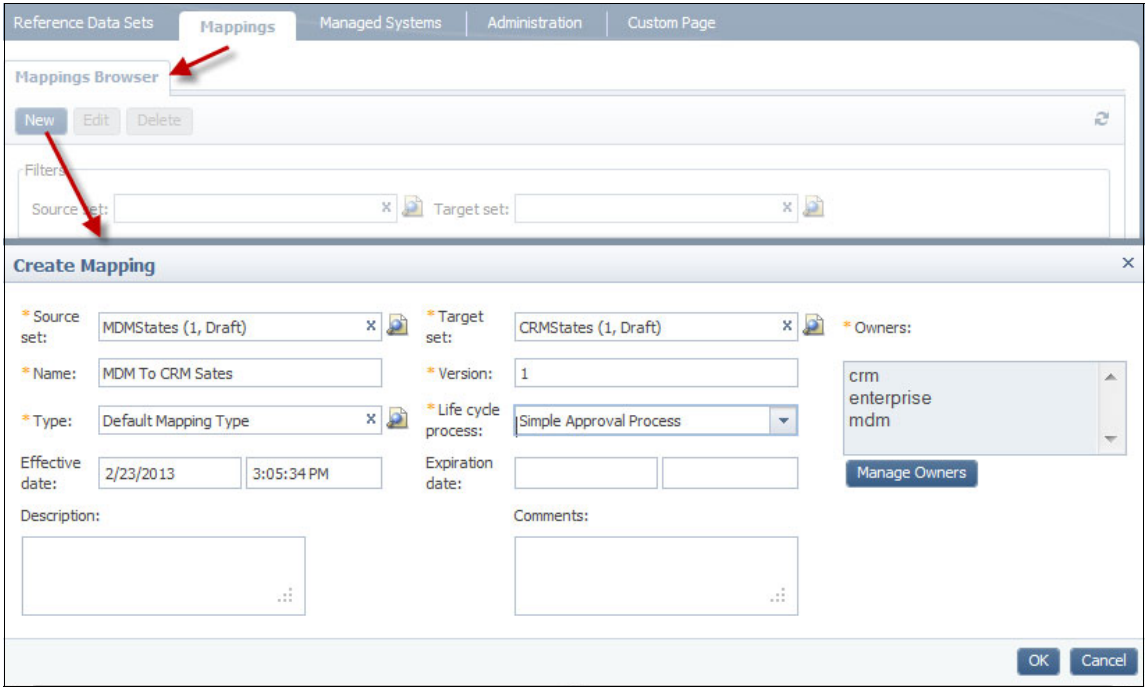


Figure 7-23 Creating a mapping

After a mapping is created, you can create value mappings either manually or by importing.

Creating value mappings manually

To map the values between two sets, complete the following steps:

1. Click **New** on the Create value mappings pane. You then see a new pane with two panes: Value Mappings and Create Mapping, as Figure 7-24 shows.

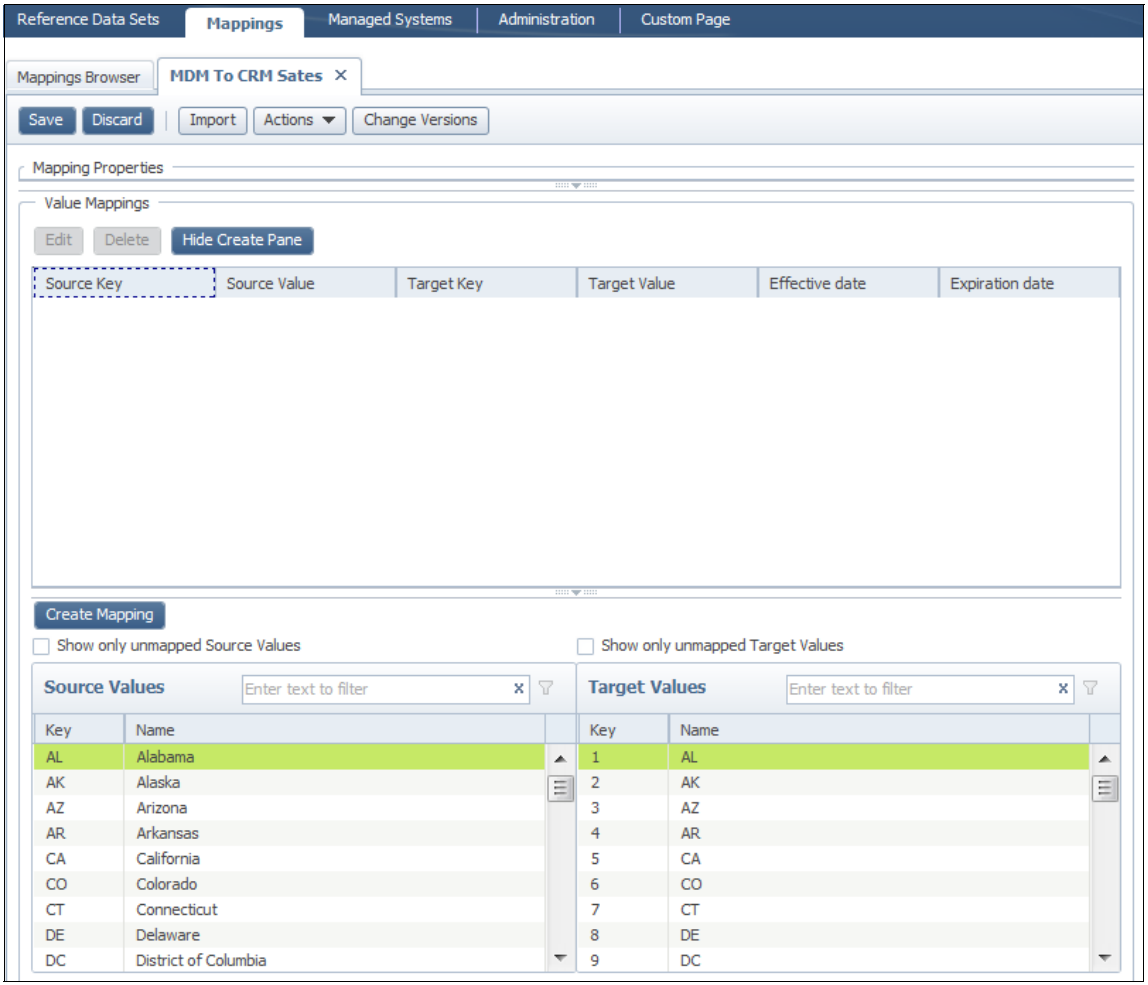


Figure 7-24 Creating a value mapping

2. Select the appropriate lines from both Source Values and Target Values to enable the Create Mapping button. In the example, it is the first line of both sets.

3. Click **Create Mapping** to create a value mapping, as shown in Figure 7-25:
   - Enter the following source values:
     - Name: Alabama
     - Code: AL
     - ISOCountry: [US] United States
   - Enter the following target values:
     - Name: AL
     - Code: 1
     - ISOCountry: [US] United States
   - Leave the custom property empty because the default type mapping has no custom properties.

   Click **OK** to create the value mapping.

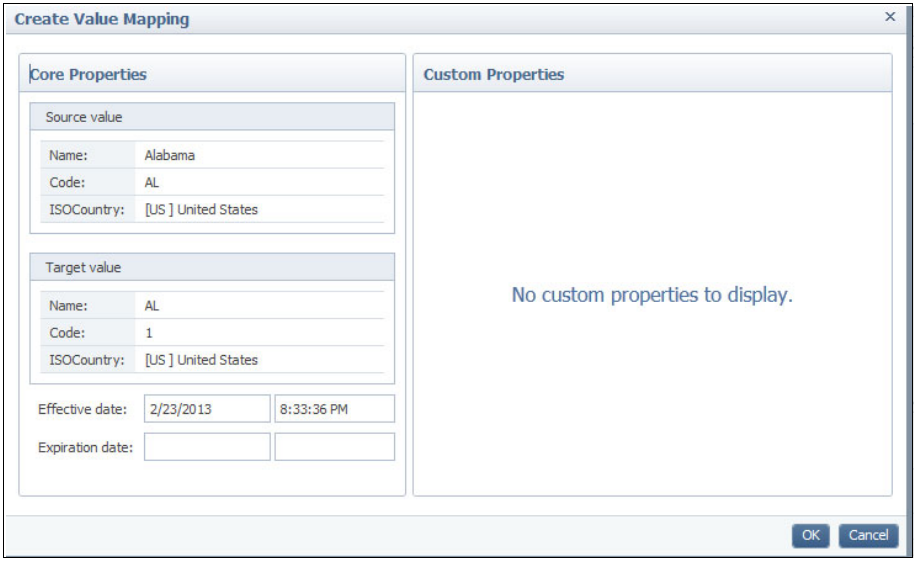


Figure 7-25 Creating a value mapping

4. Click **Save** to save the value mapping as in Figure 7-24 on page 191.
5. As shown in Figure 7-26 on page 193, a new value mapping (AL → 1) is displayed in the value mappings data grid. You may also select both of the following check boxes in the bottom pane, to show only unmapped values:
   - Show only unmapped Source values
   - Show only unmapped Target values

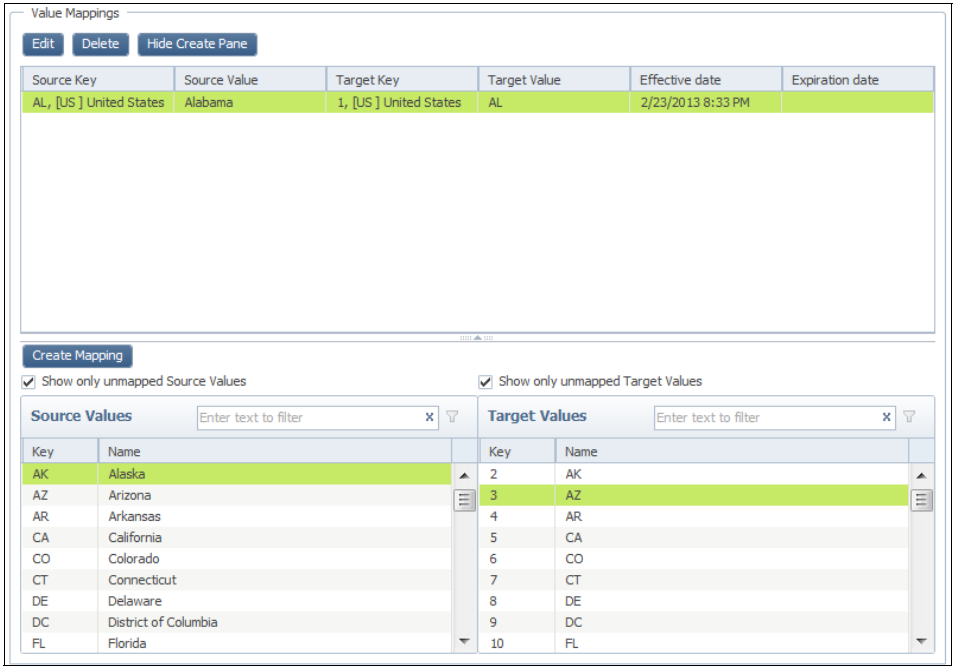


Figure 7-26 Manually created a value mapping

You can repeat these steps to create more value mappings. If you create a large number of value mappings, use import value mappings from a CSV file.

Importing a mapping from a CSV file

A mapping CSV file contains a list of the source code and the target code values. Table 7-3 shows content of a typical CSV file.

Table 7-3 MDM to CRM.csv content

MDM Code	MDM.ISOCountry.Code	CRM Code	CRM.ISOCountry.Code
AL	US	1	US
AK	US	2	US
AZ	US	3	US
AR	US	4	US
...			
WY	US	51	US

Unsupported: Importing an XML file to a mapping is not supported in InfoSphere MDM Ref DM Hub V10.1.*.

Use the following steps to import a mapping from a CSV file:

1. Log in to InfoSphere MDM Ref DM Hub with a user name and password that has a proper authorization. Click the **Mappings** tab on the navigation bar. A list of created mappings opens (Figure 7-27). If you cannot see the mapping on the list, you can type a source name or a target name in the Filters section to search for an existing mapping. Note that the MDM to CRM States mapping is highlighted.

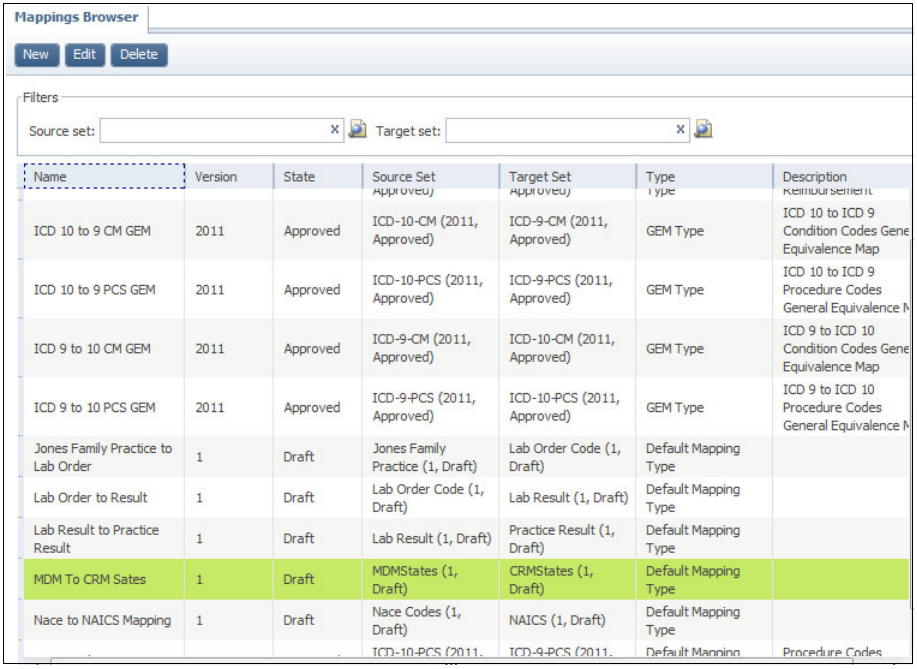


Figure 7-27 A mapping list

2. Right-click **MDM To CRM States** and select **Import** to start the import wizard.

3. In the Import mapping wizard, click **Browse** to select the **MDM to CRM States.csv** file to import, as shown in Figure 7-28.

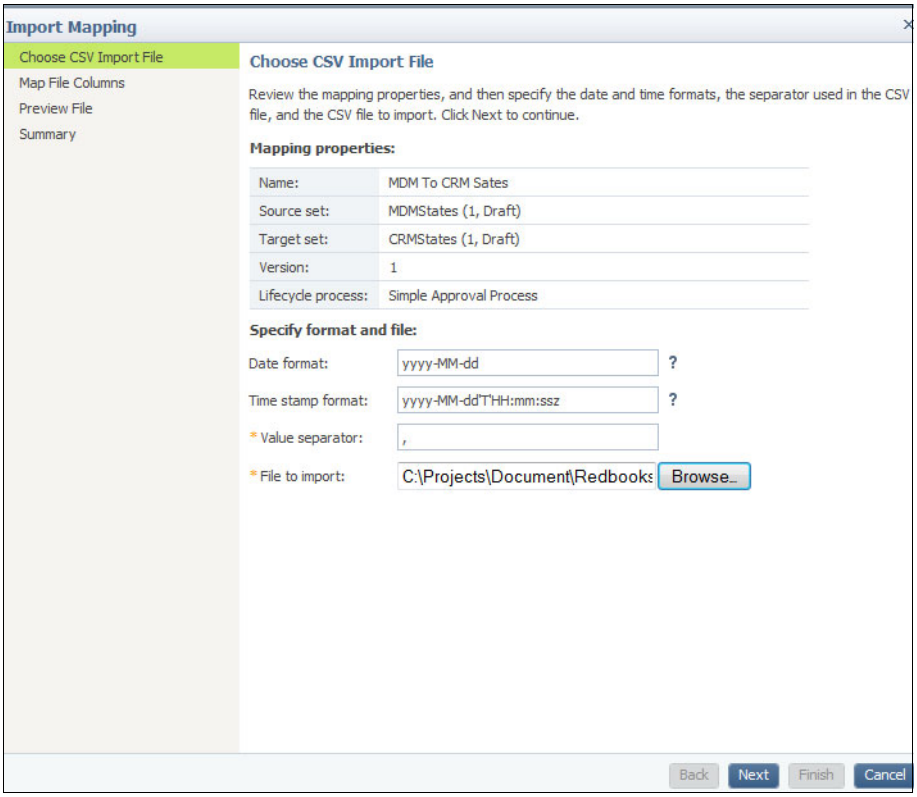


Figure 7-28 Choosing a CSV file to import

4. In Map File Columns section (Figure 7-29), select the appropriate column name to its corresponding property.

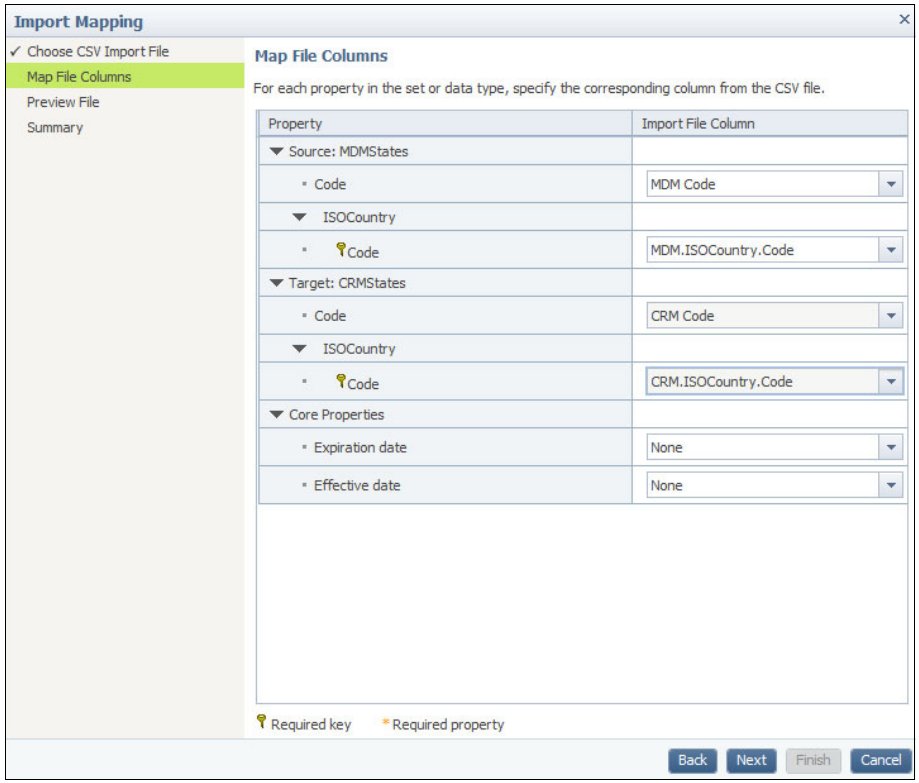


Figure 7-29 Map file column names to the mapping properties

5. In the Preview File section (Figure 7-30), carefully review the data in Data Preview tab. Click **Finish**.

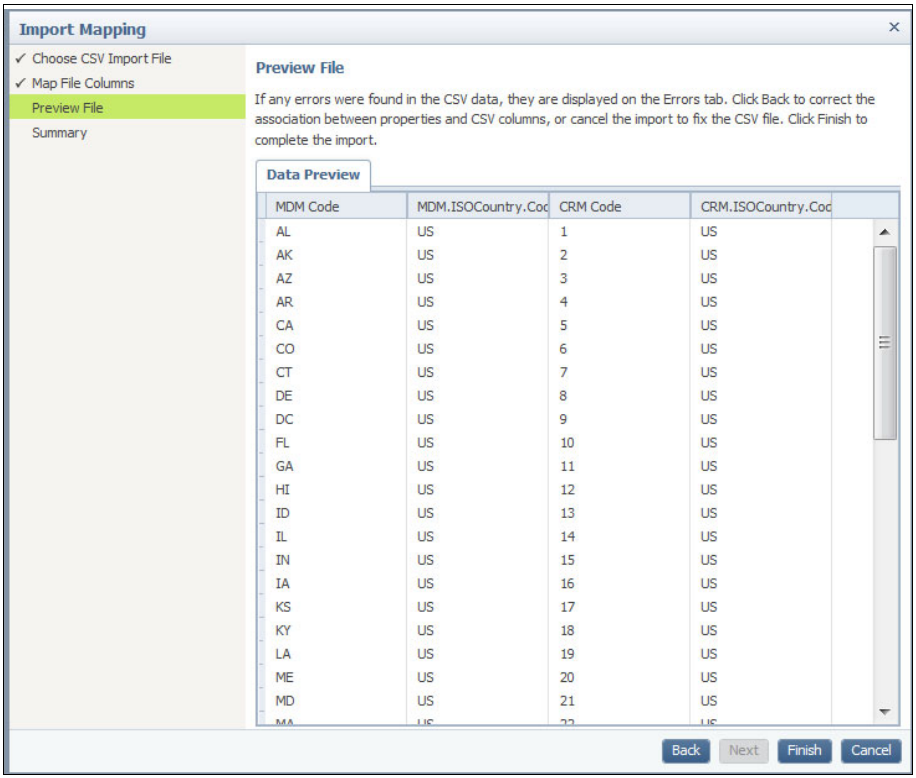


Figure 7-30 Data preview

6. The Summary section (Figure 7-31) shows you a summary with the number of inserts, exceptions, and updates. Then, click **Done**.

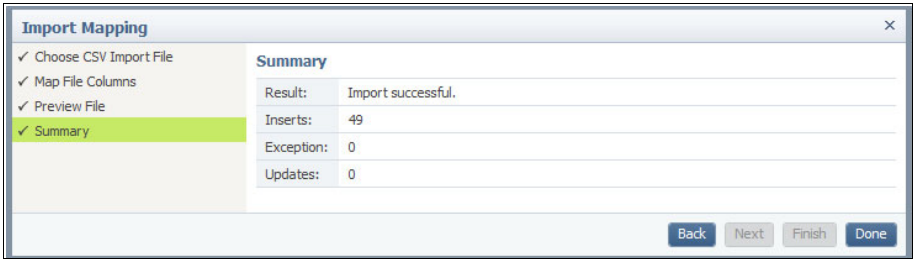


Figure 7-31 Importing mapping summary

A list of value mappings is created (Figure 7-32).

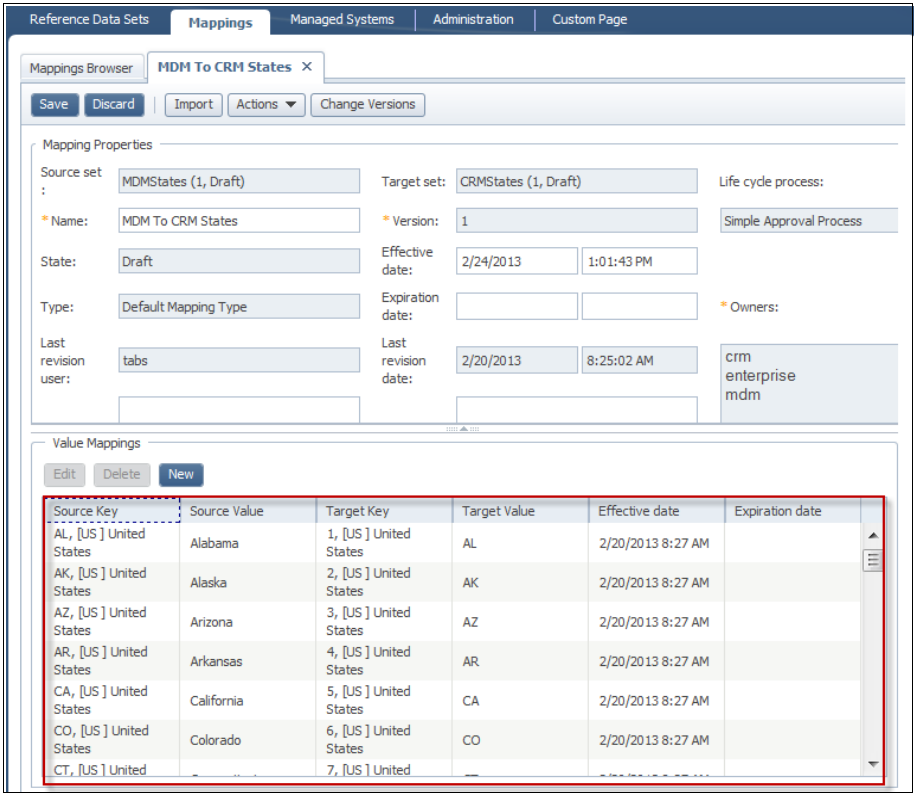


Figure 7-32 Imported MDM To CRM value mappings

Creating mapping with InfoSphere MDM Ref DM Hub web service APIs

In InfoSphere MDM Ref DM Hub V10.1.*, the REST services use the Web Service RPC client to create a mapping with the APIs.

To create a mapping with web service API, you must create the following items:

- Create a mapping with the `addRDValueSetRelationship` API. See Example 7-9.

Example 7-9 The addRDSetRelationship API

```
public RDValueSetRelationshipResponse addRDValueSetRelationship(Control
control, RDValueSetRelationship request) throws java.rmi.RemoteException,
com.ibm.wcc.service.intf.ProcessingException;
```

- Create a control object with HttpServletRequest as follows:

  ```
  Control control = ControlUtil.getNewControl(httpServletRequest);
  ```

- Create an RDValueSetRelationship object as `wsRel`.

Example 7-10 shows an API code snippet for creating new mappings.

Example 7-10 Code snippet to create a new mapping

```
RDMService_PortType RDMService = new ServiceFactory().newRDMService();
// The RDM web service port type
RDValueSetRelationship wsRel = new RDValueSetRelationship();
Long stateMachineID = 0L; // the id for the life cycle type for this mapping
RDStateMachine stateMachine = null;
RDMService.getRDStateMachine(control, stateMachineID).getResponse();
RDValueType type = null;
RDMService.getRDValueType(control, typeID, 1).getResponse();
String name = "MDM To CRM States";
wsRel.setRdState(stateMachine.getInitialState());
wsRel.setRdValueSetRelationshpName((String)name);

RDValueSet source = null;
RDValueSet target = null;
try {// Get the source and target base id
long sourceSetIDPK = wsRel.getSourceSetVersionIdPk();
long targetSEtIDPK = wsRel.getTargetSetVersionIdPk();
source = RDMService.getRDValueSet(control,sourceSetIDPK, 0).getResponse();
target = RDMService.getRDValueSet(control,targetSEtIDPK, 0).getResponse();
wsRel.setSourceSetID(source.getBaseID());
wsRel.setTargetSetID(target.getBaseID());
} catch (ProcessingException e1){
...}
create a mapping with the wsRel object.
RDValueSetRelationship mapping;
try {
mapping = RDMService.addRDValueSetRelationship(
control, wsRel).getResponse();
}
catch (NumberFormatException e) {...}
catch (ProcessingException e) {...}
catch (RemoteException e) {...}
```

Importing a mapping using the InfoSphere MDM Ref DM Hub web service API

A mapping is a relationship between two sets and it has a list of relationships that are associated between values from the two sets.

Importing a mapping is to update a mapping at the set relationship level. You can use the web service API (Example 7-11) to import a mapping.

Example 7-11 The `updateRDValueSetRelationship` *API*

```
public RDValueSetRelationshipResponse updateRDValueSetRelationship(Control
control, RDValueSetRelationship request) throws java.rmi.RemoteException,
com.ibm.wcc.service.intf.ProcessingException;
```

Example 7-12 shows a code snippet to import a mapping with API.

Example 7-12 Importing with the updateRDValueSetRelationship API

```
// both control and RDMService object are created with the same sample above
List<RDValueRelationship> newMappings;
// it is created from a list of source and target code values
// you must create the newMappings before using it
mapping.setRdValueRel(newMappings.toArray(new
RDValueRelationship[newMappings.size()]));
RDValueSetRelationship response =
RDMService.updateRDValueSetRelationship(control, mapping).getResponse();
```

7.3.3 Distributing reference data sets

InfoSphere MDM Ref DM Hub reference data sets can be distributed by manually exporting them using the web UI, by calling the web service API, or by using batch export.

Exporting reference data sets with InfoSphere MDM Ref DM Hub web UI

This section shows an example of exporting the reference data set MDMStates to an XML file by using the InfoSphere MDM Ref DM Hub web UI.

The steps are as follows:

1. Log in to the InfoSphere MDM Ref DM Hub web UI with a user name and password that has proper authorization through the following URL:

   `https://rdmdemo:9443/RefDataClient`

2. Navigate to the MDMStates set (created in 7.3.1, “Creating reference data sets” on page 172) under the `Chap08` folder.

3. Right-click **MDMStates(1, Draft)** and select **Export** to open the Export wizard, as shown in Figure 7-33.

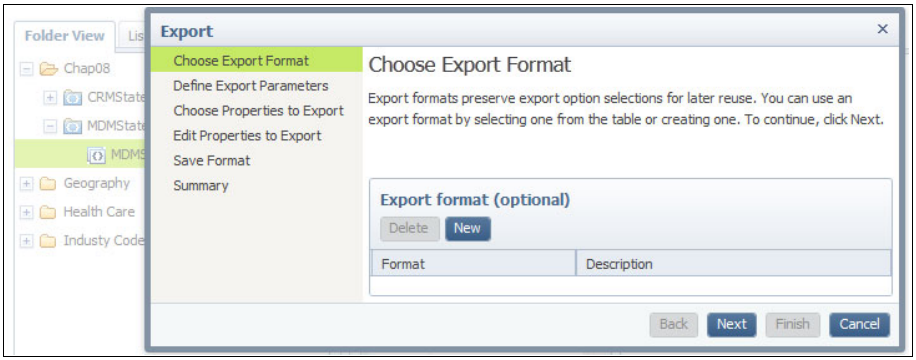


Figure 7-33 Choosing export format

4. In the Define Export Parameters section (Figure 7-34), select the **XML** file format.

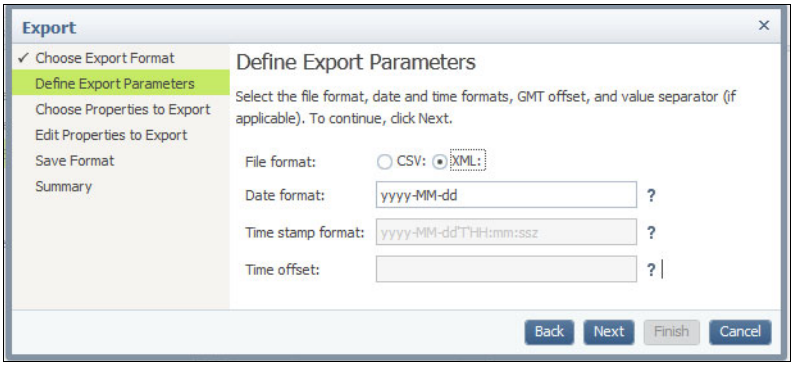


Figure 7-34 Defining export parameters

5. In the Choose Properties to Export section (Figure 7-35 on page 202), do the following steps:

   a. For the Set properties, click the double right-angle brackets button (**>>**) to move all available properties to the selected properties.

   b. For the Value properties, click the **>>** button to move all available properties to the selected properties.

   c. Select the **Export the value translations** check box.

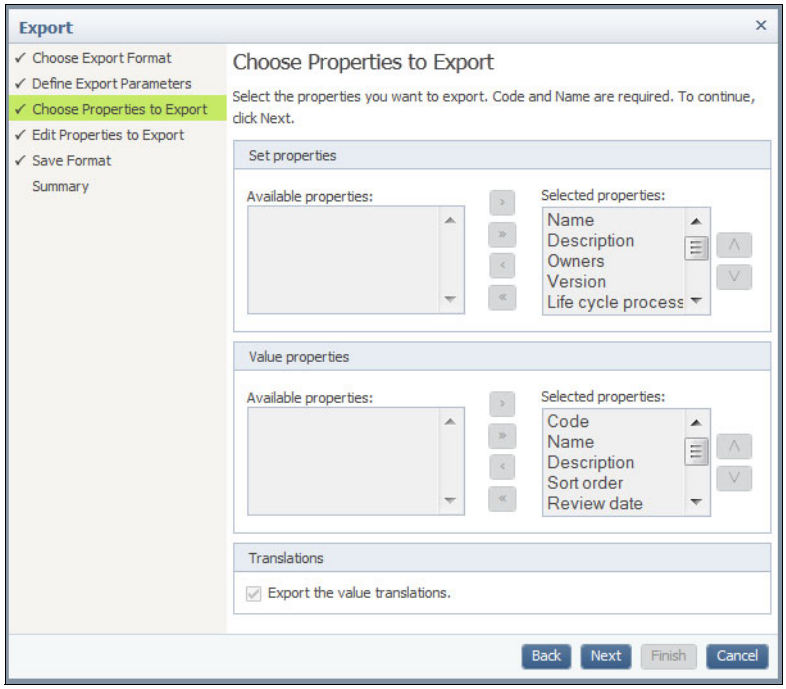


Figure 7-35 Choosing properties to export

6. In Edit Properties to Export (Figure 7-36 on page 203), do the following steps:
   - Select the column property to edit the name as your preference.
   - Click **Back** if you do not want to export a specific property.
   - Click **Next** to save the export format for reuse.

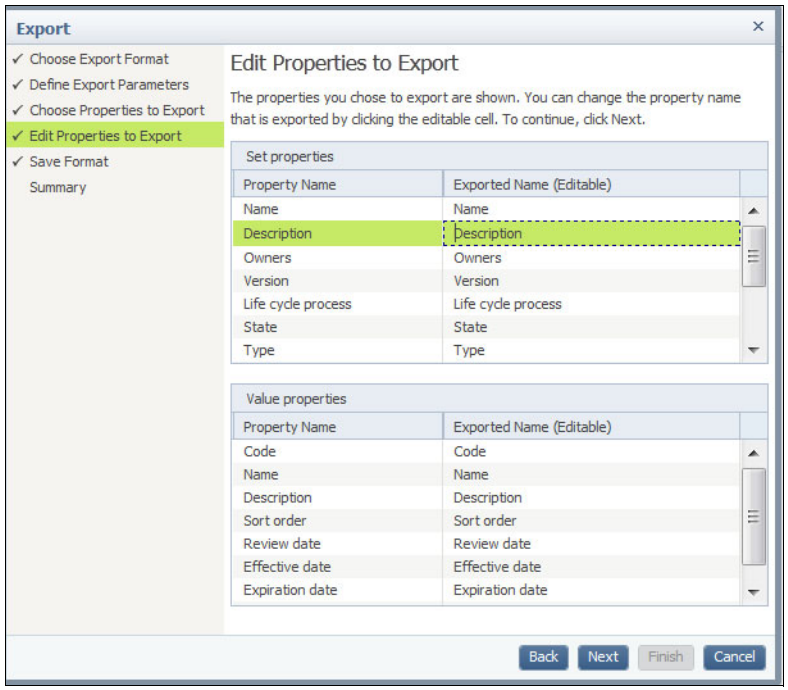


Figure 7-36 Edit properties to export

7. In Save Format, click **Finish**.
8. In Summary (Figure 7-37), click **Download File** to download the exported XML file to your file system. Then, click **Done** to complete the export task.

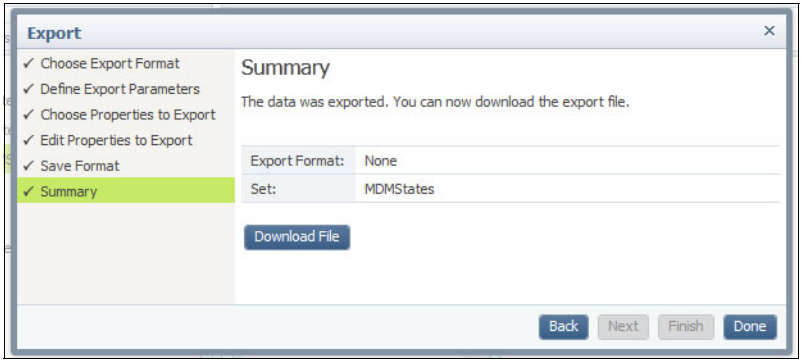


Figure 7-37 Summary

Now you can open the downloaded file to examine the exported contents. The file should have one SetLevelProperties and one list of RefDataValue.

Example 7-13 shows the content snippet of `MDMStates_Draft_1.xml`, our exported XML file.

Example 7-13 Exported XML file content

```
<SetLevelProperties>
<Name>MDMStates</Name>
<Description/>
<Owners name="Owners">
<Owner>crm</Owner>
<Owner>enterprise</Owner>
<Owner>mdm</Owner>
</Owners>
<Version>1</Version>
<LifeCycleProcess>Simple Approval Process</LifeCycleProcess>
<State>Draft</State>
<Type>CountryType</Type>
<ReviewDate/>
<EffectiveDate>2013-02-19T17:48:07Z</EffectiveDate>
<ExpiryDate/>
<LastUser>tabs</LastUser>
<LastUpdate>2013-02-18T07:58:40Z</LastUpdate>
</SetLevelProperties>
<RefDataValue>
<Code>AL</Code>
<Name>Alabama</Name>
<Description/>
<SortOrder>1</SortOrder>
<Properties>
<Property name="ISOCountry" type="Reference DataSet">
<Property name="Code">US  </Property>
</Property>
</Properties>
<ReviewDate/>
<EffectiveDate>2013-02-18T07:58:40Z</EffectiveDate>
<ExpiryDate/>
</RefDataValue>
...
```

Exporting reference data sets with the web service APIs

To export a reference data set, you must first retrieve a reference data set, the set values, and the translations of each value, and then write the reference data set into an XML or CSV file.

Example 7-14 shows the getRDValueSet API. The `rdValueSetID` is the set ID; the `inquiryLevel` is to control whether the value translations are returned.

Example 7-14 The getRDValueSet API

```
public RDValueSetResponse getRDValueSet(Control control, long rdValueSetID,
long inquiryLevel) throws java.rmi.RemoteException,
com.ibm.wcc.service.intf.ProcessingException;
```

For a description of how to create an instance of RDMService and create a control object, see 7.3.1, “Creating reference data sets” on page 172.

Example 7-15 shows how to get the RDValueSet object to be exported.

Example 7-15 An example of retrieving a reference data set

```
try {
// inquiry level is 2 for retrieving
RDValueSet response = RDMService.getRDValueSet(control, setID,2).getResponse();
} catch (RemoteException e) {
e.printStackTrace();
} catch (ProcessingException e) {...}
```

For details about the RDValueSet object, see the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.trxref.doc/data/data_RDValueSet.html

Exporting reference data sets with batch export commands

Batch export is a stand-alone Java application that allows you to export set, set data type, and mappings.

The batch export is located in `%RDM_PATH%/BatchExport`, where `%RDM_PATH%` is your InfoSphere MDM Ref DM Hub installation directory. In our example, it is `C:\Tools\IBM\InfoSphere MDM Reference Data Management`.

Figure 7-38 shows the eight files under the BatchExport directory.

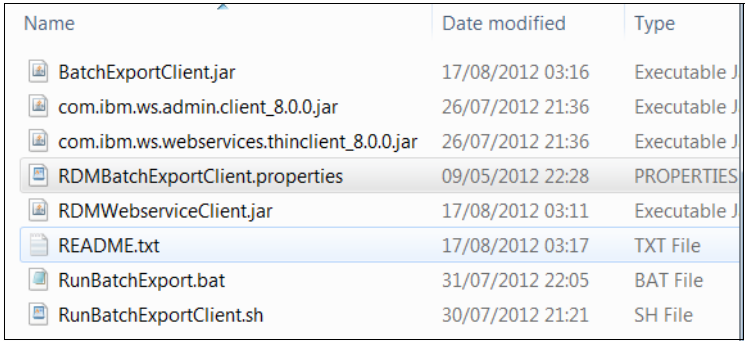


Figure 7-38 Batch export files

You must configure the `RDMBatchExportClient.properties` file to communicate with the InfoSphere MDM Ref DM Hub server. Figure 7-39 shows an example of the `RDMBatchExportClient.properties` file.

```
#Provider URL of the server instance Host:Port (The port is
usually 9080)
provider_url = rdmdemo:9080

#User Credentials
user = tabs
password = tabs
```

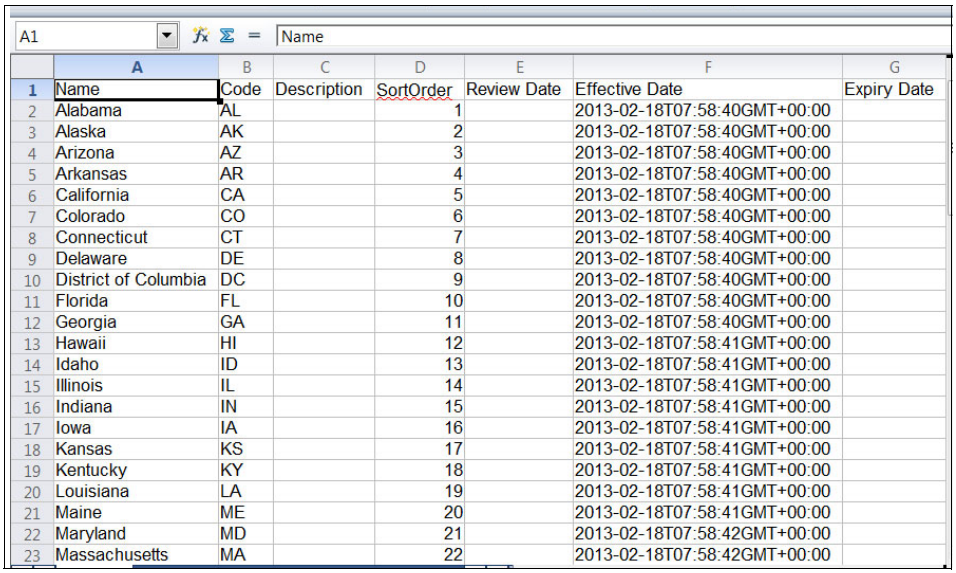
Figure 7-39 Example of batch export configuration

The following example exports a data set and type in a Windows operating system. Run the following command from a command prompt:

```
C:\Tools\IBM\InfoSphere MDM Reference Data
Management\BatchExport>RunBatchExport.bat rds=MDMStates, state=Draft, version=1
```

This command exports the MDMStates version 1 set in draft state to a CSV file. The CSV file output folder is defined in the `RDMBatchExportClient.properties` file. In our example, it is `C:\Output`.

Figure 7-40 shows the exported `MDMStates_Draft_1.csv` file.



	A	B	C	D	E	F	G
1	Name	Code	Description	SortOrder	Review Date	Effective Date	Expiry Date
2	Alabama	AL		1		2013-02-18T07:58:40GMT+00:00	
3	Alaska	AK		2		2013-02-18T07:58:40GMT+00:00	
4	Arizona	AZ		3		2013-02-18T07:58:40GMT+00:00	
5	Arkansas	AR		4		2013-02-18T07:58:40GMT+00:00	
6	California	CA		5		2013-02-18T07:58:40GMT+00:00	
7	Colorado	CO		6		2013-02-18T07:58:40GMT+00:00	
8	Connecticut	CT		7		2013-02-18T07:58:40GMT+00:00	
9	Delaware	DE		8		2013-02-18T07:58:40GMT+00:00	
10	District of Columbia	DC		9		2013-02-18T07:58:40GMT+00:00	
11	Florida	FL		10		2013-02-18T07:58:40GMT+00:00	
12	Georgia	GA		11		2013-02-18T07:58:40GMT+00:00	
13	Hawaii	HI		12		2013-02-18T07:58:41GMT+00:00	
14	Idaho	ID		13		2013-02-18T07:58:41GMT+00:00	
15	Illinois	IL		14		2013-02-18T07:58:41GMT+00:00	
16	Indiana	IN		15		2013-02-18T07:58:41GMT+00:00	
17	Iowa	IA		16		2013-02-18T07:58:41GMT+00:00	
18	Kansas	KS		17		2013-02-18T07:58:41GMT+00:00	
19	Kentucky	KY		18		2013-02-18T07:58:41GMT+00:00	
20	Louisiana	LA		19		2013-02-18T07:58:41GMT+00:00	
21	Maine	ME		20		2013-02-18T07:58:41GMT+00:00	
22	Maryland	MD		21		2013-02-18T07:58:42GMT+00:00	
23	Massachusetts	MA		22		2013-02-18T07:58:42GMT+00:00	

Figure 7-40 MDMStates.CSV file exported with batch export command

Figure 7-41 shows other batch export command arguments.

```
1 - rds=[...], state=[...], version=[...]
2 - map=[...], state=[...], version=[...]
3 - hier=[...], rds=[...], state=[...], version=[...]
4 - allrdt=[...]
5 - rdt=[...]
6 - ms=[...]
7 - sl=[...]
8 - folders=[...]
```

Figure 7-41 A full list of batch command argument list

The UNIX command syntax is similar. For details, see the information center:

http://pic.dhe.ibm.com/infocenter/mih/v10r1/topic/com.ibm.swg.im.mdmhs.rdm.usage.doc/topics/ExportingDataFromTheCommandLine.html

7.3.4 Distributing mappings

Distributing mappings is the process of exporting mappings to either a CSV file or an XML file for consumption by external applications or systems. This section shows you how to export mappings through the InfoSphere MDM Ref DM Hub web UI, API, and Batch export command.

Exporting mappings in InfoSphere MDM Ref DM Hub web UI

This example exports the *MDM To CRM* mapping, which is created in 7.3.3, "Distributing reference data sets" on page 200, to an XML file y using the InfoSphere MDM Ref DM Hub web UI.

The steps are as follows:

1. Log in to the InfoSphere MDM Ref DM Hub web UI with a user name and password that has proper authorization.
2. Click the **Mappings** tab and select **MDM To CRM States**.
3. Right-click the **MDM To CRM States** mapping and then select **Export** → **to XML** as shown in Figure 7-42.

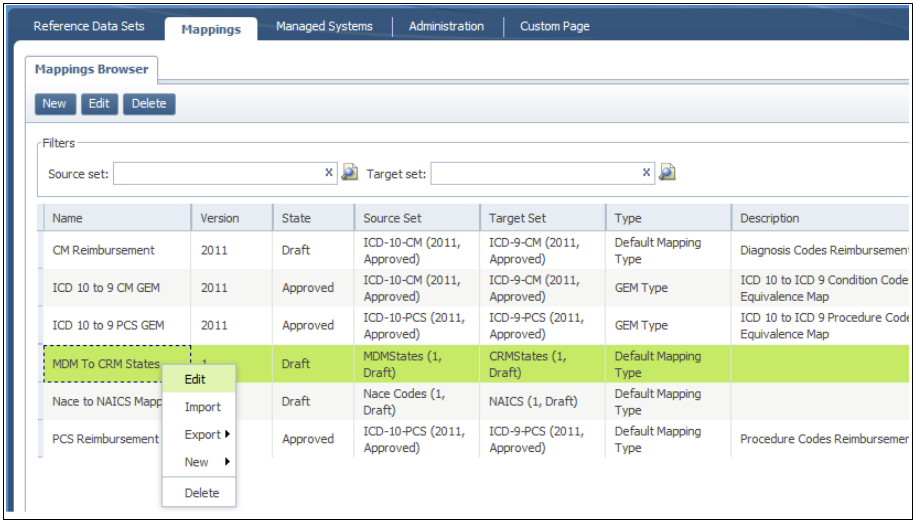


Figure 7-42 Export MDM To CRM States

4. The `MDM To CRM States_Draft_1.xml` is ready for opening or saving to a folder (Figure 7-43).

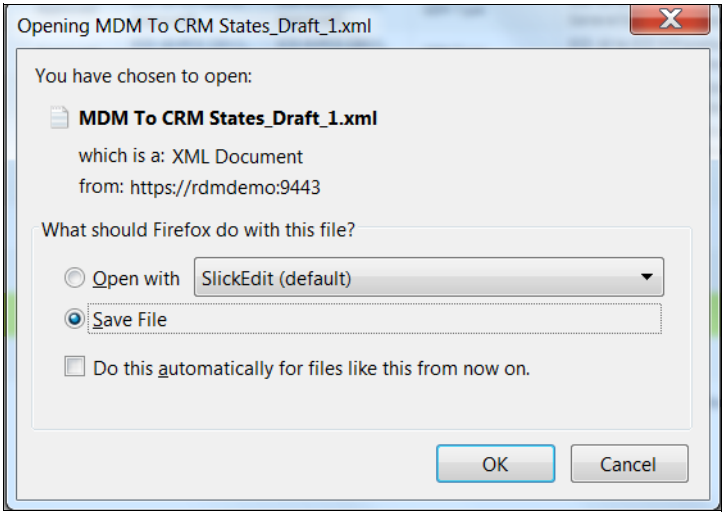


Figure 7-43 Exported MDM To CRM States XML

5. Click **Save** to save the file to your local file system for review or reuse later.

Figure 7-44 shows a snippet of the exported mapping in XML format.

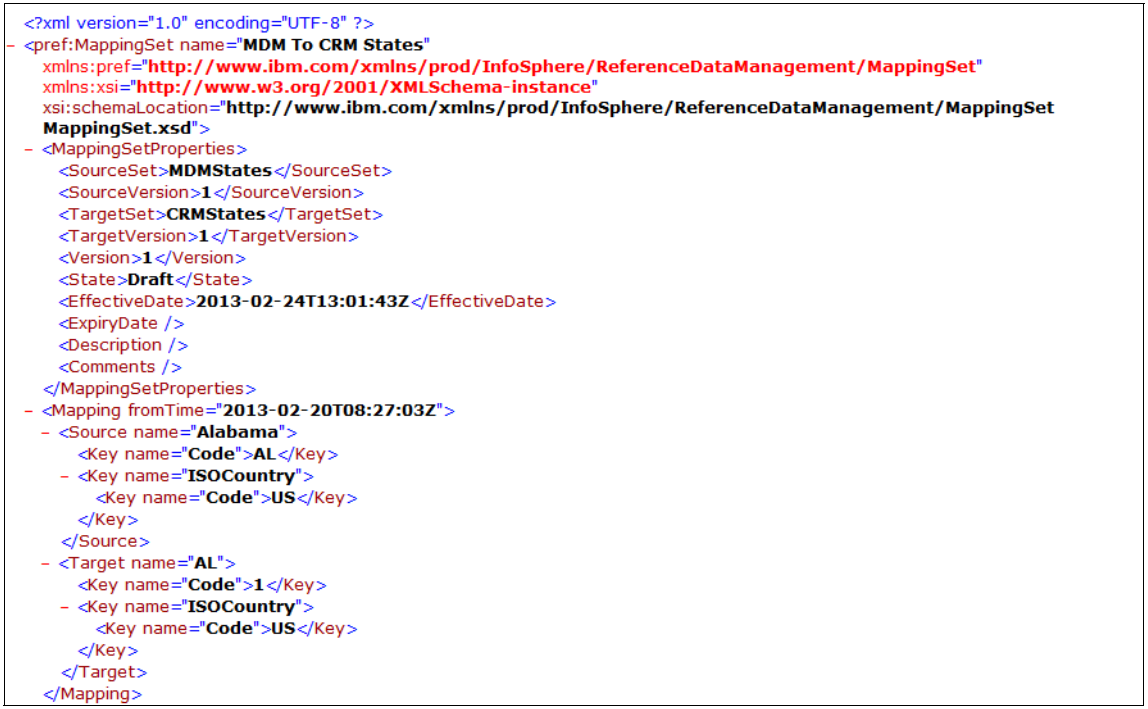


Figure 7-44 A code snippet of the exported mapping file

Exporting mappings with the web service API

You can export mappings with the web service APIs. Example 7-16 shows the API for exporting mappings.

Example 7-16 The related mapping API

```
public RDValueSetRelationshipResponse getRDValueSetRelationship(Control
control, long rdValueSetRelationshipID, long inquiryLevel) throws
java.rmi.RemoteException, com.ibm.wcc.service.intf.ProcessingException;
```

Complete the following steps before exporting the mapping set:

1. Retrieve the mapping set with a mapping ID.
2. Get information about the source and target sets.
3. Set the custom properties for this mapping set.
4. Set up the writer to write the mapping properties and the entire mapping set.

Example 7-17 shows a code snippet for exporting mappings with the web service API.

Example 7-17 Exporting mappings web service API code

```
RDValueSetRelationship map = RDMService.getRDValueSetRelationship(control,
mapId, 0).getResponse();

RDValueSet sourceSet = RDMService.getRDValueSet(control,
map.getSourceSetVersionIdPk(), 0).getResponse();
RDValueSet targetSet = RDMService.getRDValueSet(control,
map.getTargetSetVersionIdPk(), 0).getResponse();

RDValueType sourceSetType = RDMService.getRDValueType(control,
sourceSet.getDefaultRDValueTypeID(), 1).getResponse();
RDValueType targetSetType = RDMService.getRDValueType(control,
targetSet.getDefaultRDValueTypeID(), 1).getResponse();
...
```

Exporting mappings with batch export commands

Exporting mappings with the InfoSphere MDM Ref DM Hub batch export command is similar to the export reference data sets process. In this example, you export the MDM To CRM States mapping that was created in “Creating a reference data set value manually” on page 175.

Example 7-18 shows the command to export a particular version of a mapping.

Example 7-18 An example of batch command to export a mapping

```
RunBatchExport.bat map="MDM To CRM States", state=Draft, version=1
```

White space: You must use a double quotation marks around a parameter value that contains a white space character, such as the map name value in Example 7-18.

The `MDM To CRM States_Draft_1.csv` file is exported to the path you configured in the `RDMBatchExportClient.properties` file.

Figure 7-45 shows the contents of the exported mapping in CSV format. All of the following items are exported: both the source code and the target code, both source and target custom property code, their names, the effective date, and the expiration date are exported.

	A	B	C	D	E	F	G	H
1	Source Code	Source Country Code	Source Country Name	Target Code	Target Country Code	Target Country Name	Effective Date	Expiration Date
2	AL	US	United States	1	US	United States	2013-02-20T08:27:03GMT+00:00	
3	AK	US	United States	2	US	United States	2013-02-20T08:27:03GMT+00:00	
4	AZ	US	United States	3	US	United States	2013-02-20T08:27:04GMT+00:00	
5	AR	US	United States	4	US	United States	2013-02-20T08:27:04GMT+00:00	
6	CA	US	United States	5	US	United States	2013-02-20T08:27:04GMT+00:00	
7	CO	US	United States	6	US	United States	2013-02-20T08:27:04GMT+00:00	
8	CT	US	United States	7	US	United States	2013-02-20T08:27:04GMT+00:00	
9	DE	US	United States	8	US	United States	2013-02-20T08:27:04GMT+00:00	
10	DC	US	United States	9	US	United States	2013-02-20T08:27:04GMT+00:00	
11	FL	US	United States	10	US	United States	2013-02-20T08:27:04GMT+00:00	
12	GA	US	United States	11	US	United States	2013-02-20T08:27:04GMT+00:00	
13	HI	US	United States	12	US	United States	2013-02-20T08:27:04GMT+00:00	
14	ID	US	United States	13	US	United States	2013-02-20T08:27:04GMT+00:00	
15	IL	US	United States	14	US	United States	2013-02-20T08:27:04GMT+00:00	
16	IN	US	United States	15	US	United States	2013-02-20T08:27:05GMT+00:00	
17	IA	US	United States	16	US	United States	2013-02-20T08:27:05GMT+00:00	
18	KS	US	United States	17	US	United States	2013-02-20T08:27:05GMT+00:00	
19	KY	US	United States	18	US	United States	2013-02-20T08:27:05GMT+00:00	
20	LA	US	United States	19	US	United States	2013-02-20T08:27:05GMT+00:00	
21	ME	US	United States	20	US	United States	2013-02-20T08:27:05GMT+00:00	
22	MD	US	United States	21	US	United States	2013-02-20T08:27:05GMT+00:00	
23	MA	US	United States	22	US	United States	2013-02-20T08:27:05GMT+00:00	
24	MI	US	United States	23	US	United States	2013-02-20T08:27:05GMT+00:00	

Figure 7-45 The exported MDM To CRM States mapping CSV file

7.4 Customization

With InfoSphere MDM Ref DM Hub, you can customize both the server component and the web UI component. This section shows you how to customize the InfoSphere MDM Ref DM Hub for your own specific requirements. The following scenarios are typical customization scenarios:

- Notification for reference data consumers
- External business rule validation
- The web user interface

7.4.1 Notification

The InfoSphere MDM Ref DM Hub can centralize and manage an organization's critical reference data entities such as country, language, business units, and more. Some InfoSphere RDM reference architectures that are described in Chapter 2, "Solution reference architecture" on page 35 require the notification of external applications to synchronize to the reference data that is managed in the InfoSphere MDM Ref DM Hub.

In a tightly coupled system, the applications can connected to the InfoSphere MDM Ref DM Hub directly. However, as illustrated in Figure 7-46, changes applied to the data in the InfoSphere MDM Ref DM Hub are usually not directly sent to an application. Instead, an enterprise application integration (EAI) component is used to receive the change notification and distribute the event to all connected applications that are interested in the change.

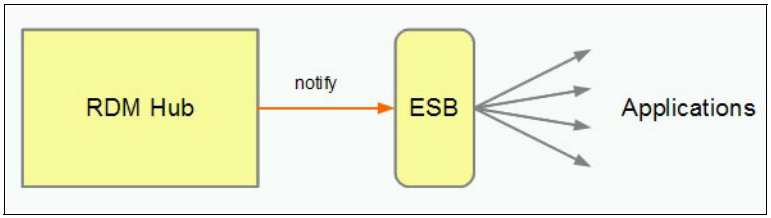


Figure 7-46 Notification scenario

The necessary steps to send the change notification to WebSphere Enterprise Service Bus (ESB) are as follows:

1. Enable notifications in the InfoSphere MDM Ref DM Hub database.
2. Create a connected JMS topic in the ESB.

   The JMS topic that is used to distribute the customer data is already defined in InfoSphere MDM Ref DM Hub. However, to receive this data in the ESB, a corresponding topic is required in WebSphere Enterprise Service Bus. The JMS topics must be connected. Using the connected topics, messages sent to the InfoSphere MDM Ref DM Hub topic can be received with the ESB topic.
3. Extend the InfoSphere MDM Ref DM Hub model with behavior extensions by using the Custom Domain Hub Workbench.

   With a behavior extension, business logic can be executed before or after an InfoSphere MDM Ref DM Hub transaction. This extension can send a simple text message indicating the update of a InfoSphere MDM Ref DM Hub reference data set to the InfoSphere MDM Ref DM Hub topic, which forwards the notification event to the ESB.

Enabling notifications in the InfoSphere MDM Ref DM Hub database

To work with InfoSphere MDM Ref DM Hub notifications, you must enable the Custom Domain Hub notification mechanism by using the following steps:

1. Open a DB2 command window and connect to your Custom Domain Hub database (`cdhdb`) by using the following command:

   ```
   db2 connect to cdhdb
   ```

2. Enable notifications at the application level by using the following command:

   ```
   db2 update configelement set value='true', last_update_dt=current timestamp
   where name='/IBM/DWLCommonServices/Notifications/enabled'
   ```

3. Disconnect from the CDH database by using the following command:

   ```
   db2 disconnect cdhdb
   ```

Creating a connected JMS topic in the ESB

To create a connection topic, set up the JMS on your ESB:

1. Open the Administration console of the Process Server and expand the Resources branch and the JMS branch.
2. Click **Topics**.
3. Select the scope:
   - For Node, enter the name of your WebSphere Application Server node.
   - For Server, enter `server1`.
4. Click **New** to add a new topic.
5. Ensure that the **Default messaging provider** is selected and click **OK**.

   The Topic configuration panel opens (Figure 7-47 on page 215).
6. Enter `MDMTopic` for the name and `com/dwl/base/notification/MDMTopic` for the JNDI name of the topic.
7. Set the Topic name to `notification/ElementChange`, as defined by InfoSphere MDM Ref DM Hub.
8. Set the Bus name to **other, please specify**. For the Bus name, specify `MDM.SIB.server1`, as defined in InfoSphere MDM Ref DM Hub.
9. Set the Topic space to **other, please specify**. For the name of the origin Topic Bus destination, specify `notification.ElementChange`.

10. Click **OK** to confirm the changes and save the configuration.

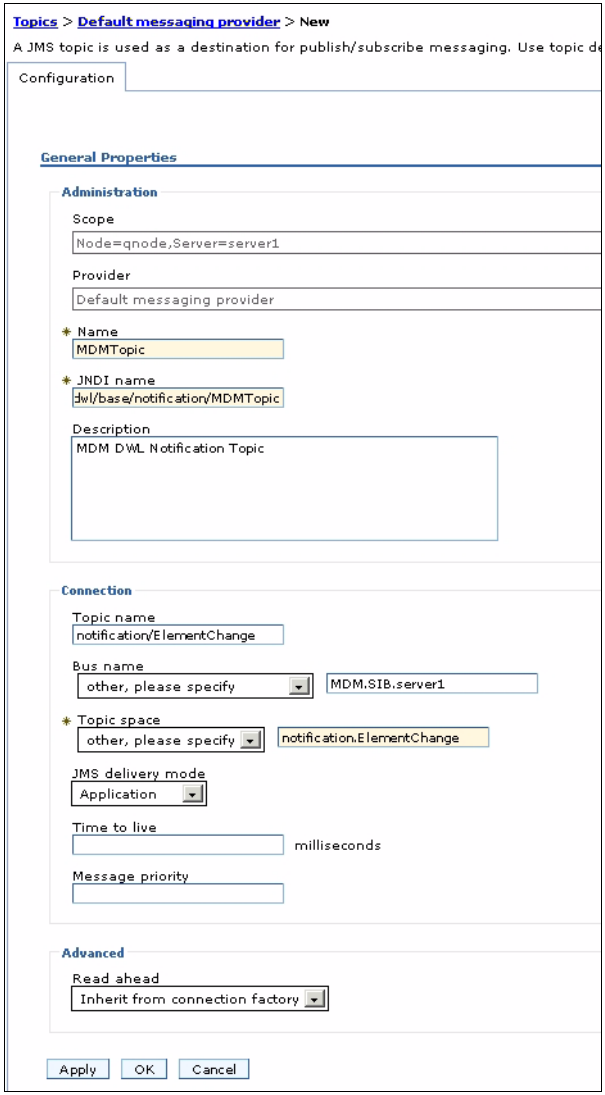


Figure 7-47 ESB Configuration: Create JMS Topic in ESB

Creating and connecting the activation specification

Configure the activation specification for the JMS with the following steps:

1. Click **Activation specifications**.
2. Select the scope:
   - For Node, enter the name of your WebSphere Application Server node.
   - For Server, enter `server1`.
3. Click **New** to add a new activation specification (Figure 7-48).

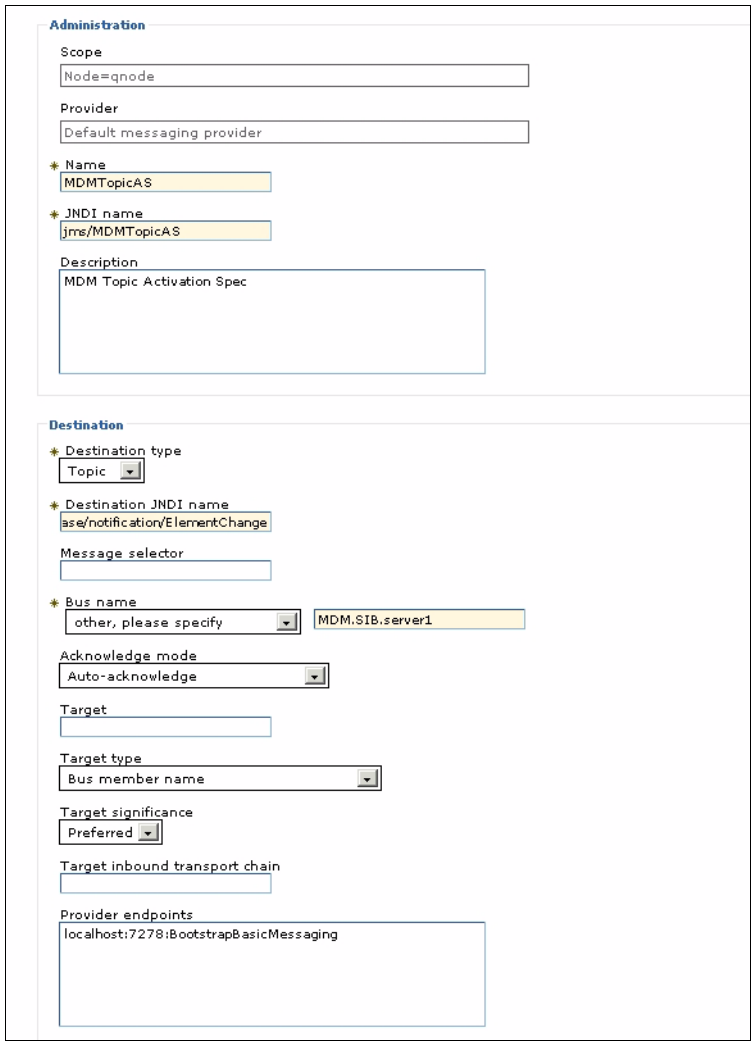


Figure 7-48 Creating the activation specification

4. Ensure that **Default messaging provider** is selected and click **OK**.

   The Activation Spec configuration panel opens.

5. Enter `MDMTopicAS` for the Name and `jms/MDMTopicAS` for the JNDI name of the activation specification.
6. For the Destination type, select **Topic**.
7. Set the Destination JNDI name to the JNDI name of the appropriate MDM Topic JNDI name as `com/dwl/base/notification/ElementChange`.
8. Set the Bus name to **other, please specify**. Specify `MDM.SIB.server1` for the name of the origin MDM bus name.
9. In the Multi-entry Provider endpoints field, enter the MDM server destination in the following format, where `port` is the SIB_ENDPOINT_ADDRESS port of your InfoSphere MDM Ref DM Hub server:

   ```
   host:port:chainname
   ```

   Examples are as follows; the example in line two is for secured hosts:

   ```
   mdmhost:7277:BootstrapBasicMessaging
   mdmhost:7287:BootstrapSecureMessaging
   ```

10. Click **OK** to confirm the changes and save the configuration.

Extending InfoSphere MDM Ref DM Hub model with behavior extensions using the Custom Domain Hub Workbench

In this example, the behavior of the updateRDValueSet transaction is extended. In general, you can extend any transaction that is listed on the transaction reference page in the InfoSphere MDM Ref DM Hub information center:

`http://publib.boulder.ibm.com/infocenter/mdm/v10r0m0/topic/com.ibm.swg.im.mdmhs.rdm.ref.doc/reference_overview.html`

To receive notification of data updates, you must create a Behavior Extension in MDM Server:

1. In the MDM Workbench, create a Hub Module project by selecting **File** → **New** → **other** and scrolling to the InfoSphere Master Information Hub folder. Select **Hub Module** project and click **Next**.
2. Fill the fields of the wizard page:
   - Project name: `RDValueSetBehaviorExtensions`
   - Base Java package name: `com.example.mdm`
   - Service namespace URI: `http://example.com/mdm`
   - EAR project name: `MDM`

- Hub base name: `MDM`
- Database schema name: `db2admin`

  The schema name must be identical to the schema of the database user name you entered during DB2 installation with the Development and Test Environment wizard.

3. Click **Finish** to create the project.
4. Add the required behavior extensions:
   a. Select the **Model** tab to switch to Hub Module model.
   b. Click the **PartyExtensions** folder, then select **New** and click **Behavior Extension**.
   c. Enter `RDValueSetBehaviorExtensions` in the Name field; in the Implementation field, be sure **Java** is selected.
   d. Right-click **RDValueSetBehaviorExtensions** extension and select **New** → **Action Event**.
   e. Enter `UpdateRDValueSetEvent` in the Name field and click **Edit** for the transaction.
   f. Expand the model and select the **updateRDValueSet** event and click **OK**.
5. Click **Validate model** on the bottom. The following message is displayed:

   ```
   Model validation complete: no problems found
   ```

   If it does not appear, fix the error.
6. When all issues are resolved, click **Generate Implementation**. The `Generating Code` message is displayed and the extension code generation is started.
7. When the code generation is completed, open the RDValueSetBehaviorExtensions project and switch to **ejbModule** → **com.example.mdmsap.behaviour**. This is where you implement the extension by implementing the generated method stubs.

   Implementing an extension to send notifications is simplified with the notification framework. implement the following class and return the message string to be sent in the getMessage() method:

   ```
   com.dwl.base.notification.CommonNotification
   ```

   Depending on your scenario, the returned message can be a computed string based on the currently processed model object or just represent a trigger for an application refresh.

Example 7-19 shows the code sample of sending the message.

Example 7-19 Sending message call

```
ServiceLocator serviceLocator = ServiceLocator.getInstance();
NotificationLocalHome home = (NotificationLocalHome)
serviceLocator.getLocalHome(<TOPIC_JNDI_NAME>);
NotificationLocal theNotificationBean = home.create();
theNotificationBean.notify(theCommonNotification);
```

8. Make sure that the **Project** → **Build automatically** option is enabled. Refresh the project.
9. Update the class path by right-clicking the **RDValueSetBehaviorExtensions** project and selecting **Properties**. Now switch to Java EE Module Dependencies and look for the `RDM.jar` entry in the list. Select the check box next to `RDM.jar` file. Click **OK**.
10. Update the database with the generated behavior extensions:
    a. Open a DB2 command window.
    b. Change to the `PartyExtensions\resources\sql\db2` directory in your RSA workspace.
    c. Run the **`RDValueSetBehaviorExtensions _MetaData_DB2.sql`** SQL script:

```
db2 connect to cdhdb
db2 -tvf RDValueSetBehaviorExtensions_MetaData_DB2.sql
db2 disconnect cdhdb
```

7.4.2 Services

You can customize InfoSphere MDM Ref DM Hub to inject validation rules or event notification for both before and after a transaction. You need the following development tools:

- IBM Rational® Software Architect (RSA) V8.0.4
- Master Information Hub Workbench Extension Development Tools V10.1
- IBM WebSphere Application Server V8.0.*

This section describes the following tasks:

- Set up a development environment for the InfoSphere RDM server projects in RSA.
- Create a new InfoSphere Hub module project.

Setting up development environment for service customization

Complete the following steps:

1. Open RSA 8003 with a new workspace named `C:\IM\RDMExtension`.
2. Switch to J2EE perspective in RSA and make sure J2EE capability is enabled first.
3. Download the `MDM-App.ear` file from the RDM V10.1.* release (`MIH-App.ear` is for RDM V10 release).
4. Import `MDM-App.ear` into RSA by right-clicking to select import an EAR file and provide the project name `MDM`. Omit the following text because the code generation does not use the dash (-) character:

   ```
   -App
   ```

5. Select only **RDM.jar** and **RDMWS.jar** in the utility wizard.
6. Select all EAR files on the second panel.
7. Create a general project named `CustomerResources` and click **Finish.**
8. Create an `xsd` folder under the CustomerResources project.
9. Extract the `properties.jar` file from the `MDM-App.ear` file to a properties folder.
10. Copy the properties folder to the CustomerResources project.
11. Extract the `xsd` and `wsdl` files from `DWLSchemas.jar` file in the `MDM-App.ear` to the `xsd` folder.
12. Two build errors are displayed, you can double-click each one to view:
    - Error one:

      ```
      cvc-complex-type.2.4.a: Invalid content was found starting with element
      'webApp'. One of '{"http://websphere.ibm.com/xml/ns/javaee":servlet,
      "http://websphere.ibm.com/xml/ns/javaee":default-error-page,
      ....
      "http://websphere.ibm.com/xml/ns/javaee":enable-serving-servlets-by-clas
      s-name}' is expected.ibm-web-ext.xml/MDMWSProvider/WebContent/WEB-INF
      line 4 XML Problem
      ```

      Double-click the error and you see that the error is caused by the following line; you can ignore this error message:

      ```
      <webApp href="WEB-INF/web.xml#WebApp_ID"/>
      ```

– Error two:

```
cvc-complex-type.4: Attribute 'userid' must appear on element 'run-as'.
ibm-application-bnd.xml/MDM/META-INF in MDMline 7XML Problem
```

You can correct this error by double-click the error message and changing `<run-as/>` to the following line:

```
<run-as userid="" password="{xor}"/> and save it .<run-as userid=""
password="{xor}"/>
```

13..Save the changes in the file.

Creating a new hub module project

Use the following steps to create a new InfoSphere MDM Ref DM Hub module in RSA:

1. Create a new InfoSphere MDM Ref DM Hub module by clicking **File** → **New** → **Other** → **Hub Module Project**.
2. Type a project name, for example, `RDMCustomHub` under the InfoSphere Master Data Management Server.

There are normally two errors in your workspace after you create a new InfoSphere MDM Ref DM Hub module.

- The first error results because the newly created InfoSphere MDM Ref DM Hub module does not have a targeted runtime that is associated with it. You must manually add WebSphere Application Server V8.0 at run time.
- The second error is as follows:

```
Errors occurred during execution
java.lang.NullPointerException
```

You must change the `application.mdmxmi` file under the MDM project to make sure that the name is **MDM**, which is same as the EAR file name. In our example, the assumption is that the EAR name was changed from `MDM-App.ear` to `MDM` when importing the `MDM-App.ear` file.

Note: If the EAR name differs from the name that is defined in `application.mdmxmi`, you receive the NULL pointer exception.

7.4.3 Callout and user exits

You can customize a callout or user exits for external business rule extension. When multiple business users are working on the same data from different geographies, data entered in InfoSphere RDM must be managed with a lifecycle management. InfoSphere MDM Ref DM Hub provides basic validation rules for simple string type data checks. For advanced data consistency checks, use an external business rule extension to accomplish this task.

For example, a custom reference data set property has a set of values that must be restricted with certain types of values, such as all values must have a prefix of 90, 92, or 94. Any other values should not be used by the container reference data set. You can implement this rule through the process of sending a request to append a reference data set for approval**.**

The basic steps for creating a user exit are as follows:

1. Create a behavior extension.
2. Generate implementation.
3. Run database scripts.
4. Deploy the MDM ear.

Creating a behavior extension

Before creating a behavior extension, you must add the RDM project to the RDMCustomHub dependency so that you can see all the InfoSphere MDM Ref DM Hub transactions.

Complete the following step to create a behavior extension:

1. Open the `module.mdmxmi` file from the newly created hub module, `RDMCustomHub`.
2. Open the model tab to create a new behavior extension by right-clicking **RDMCustomHub** → **New** → **Behavior Extension**, as in Figure 7-49.

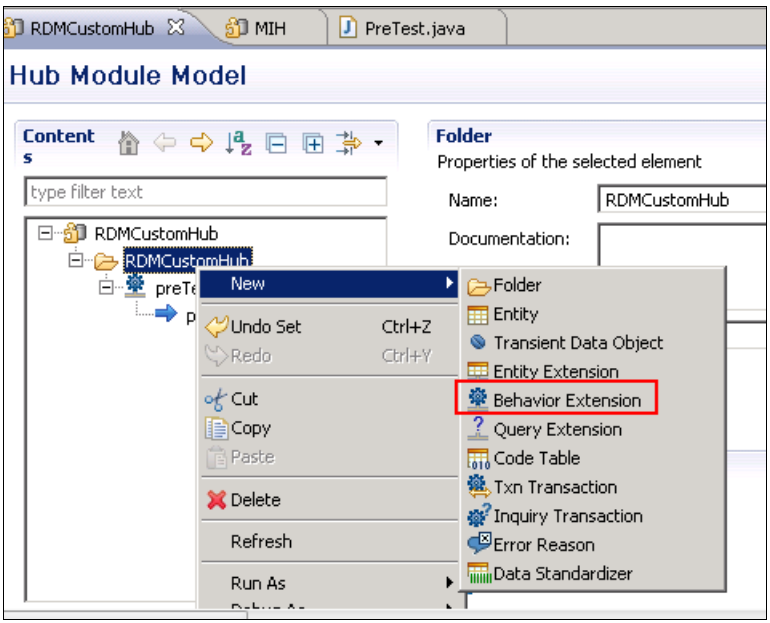


Figure 7-49 Create a behavior extension

3. Enter a name for the behavior extension, such as `preTest`.

4. Create a new transaction event named `preTestCheckup` (Figure 7-50).

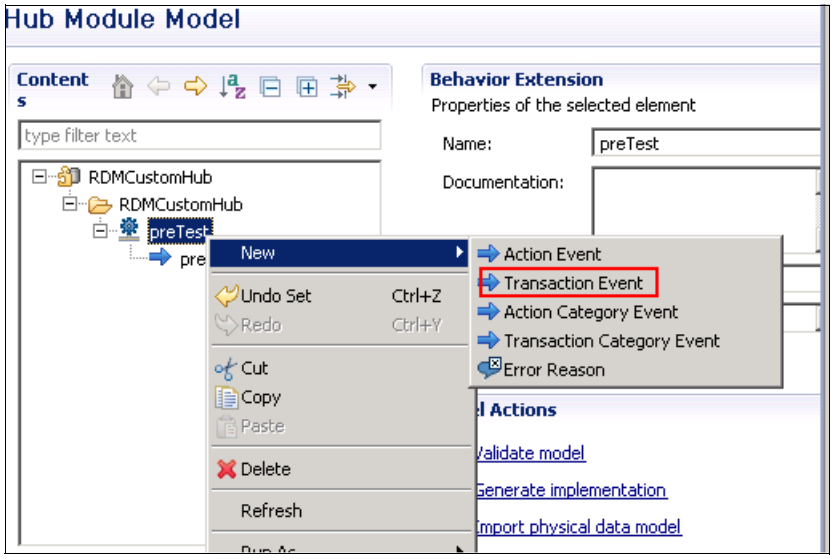


Figure 7-50 Create a transaction event

5. Set the transaction and the event type by selecting the **Pre** check box, as shown in Figure 7-51, and clicking **Edit** to enter the transaction name, `updateRDValue`. Click **OK** to complete the behavior extension.

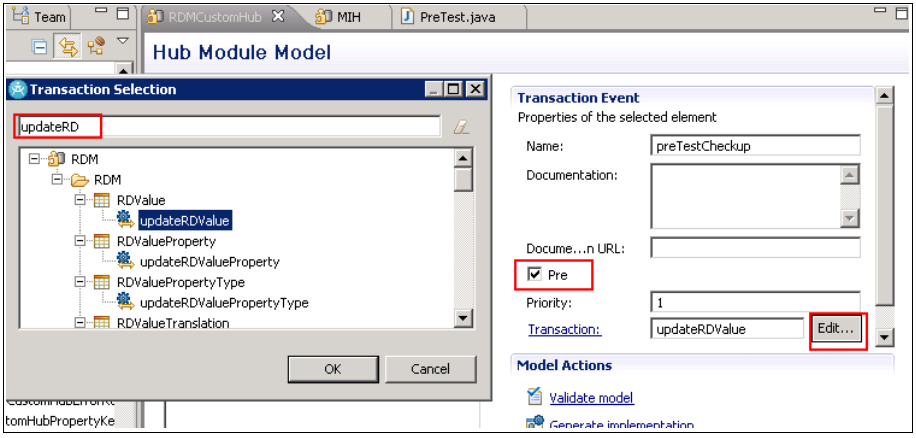


Figure 7-51 Select a transaction to customise

6. Press Ctrl+S to save the changes.

Generating the implementation

Use the following steps:

1. Before generating the implementation, ensure no model error occurs.
2. Click **Generate Implementation**, which generates the code for you. Ensure no errors occur.
3. The `PreTest.java` class is generated for you. Open this Java file in your workspace to add the business logic for the validation rules.

In your workspace, you might have one XML problem in MDMWSProvider, but you can ignore it.

Running database scripts

Normally, two SQL scripts are created when an InfoSphere MDM Ref DM Hub module is created. The following scripts are examples:

```
RDMCustomHub_MetaData_DB2.sql
Rollback_RDMCustomHub_MetaData_DB2.sql
```

You must run `RDMCustomHub_MetaData_DB2.sql` before you update the `MDM.ear` file.

If you make any changes to the InfoSphere MDM Ref DM Hub module, you must run the rollback script before running the new MetaData script.

Deploying the new MDM EAR file

You can export the MDM or MIH EAR file and update the EAR through the WebSphere Application Server administration console. This method is quicker than doing it manually.

To do this method manually, complete the following steps:

1. Export RDMCustomHub to a JAR file by right-clicking on the project and then selecting **Export** → **Java** → **Jar** file.
2. Update the `DWLCommonServicesEJB.jar` file in the `MDM-App.ear` file with `RDMCustomHub.jar` and `MANIFEST.MF` files:
   a. Extract the `MANIFEST.MF` file from the `DWLCommonServicesEJB.jar` file.
   b. Add the `RDMCustomHub.jar` name in the `MANIFEST.MF` file.
3. Drop the `DWLCommonServicesEJB.jar` in the MDM-App folder, where MDM-App was initially installed.
4. Restart the MDM application from the WebSphere Application Server administration console.

7.4.4 User interface

InfoSphere MDM Ref DM Hub web UI has a custom page where you can customize your own UI. which can be implemented using HTML, JavaScript, or Java code. If you log in to the InfoSphere MDM Ref DM Hub web UI, you see the Custom Page tab on the navigation tab, as show in Figure 7-52.

Figure 7-52 Custom page for your customization

You can import the InfoSphere MDM Ref DM Hub client component EAR file, `RDMClientEAR.ear`, into your development environment (Rational Software Architect 8.0.4) and you can find the `custom.inc.jsp` file in the RDMClientWeb project under the `WebContent\pages\custom\` location.

For customizing the web UI, modify `custom.inc.jsp` by using InfoSphere MDM Ref DM Hub REST services.

8

Operation and administration

This chapter contains thoughts about how to plan for the ongoing administration of an IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) solution. Establishing overall governance to the management of reference data is key to the ongoing quality. This chapter includes the following topics:

- Managing InfoSphere MDM Ref DM Hub with multiple user groups
- New user group setup process
- Reporting

8.1 Managing InfoSphere MDM Ref DM Hub with multiple user groups

An InfoSphere MDM Ref DM Hub installation can be used for many various owners and consumers. Each user group (owner and consumer, also called an adopter) can manage, use, or share different reference data sets and mappings. Each reference data set and mapping can have different type definitions. All of this complexity leads to a best practice to document and update the initial information design and ongoing changes.

8.1.1 Initial model design for a new user group

The initial design should capture all the information necessary to set up the objects in InfoSphere MDM Ref DM Hub. Figure 8-1 illustrates a design option that establishes a set of standard Set and Value properties that are in all reference data sets (RDS) to ensure content quality. Then, individual taxonomies and mappings have custom properties.

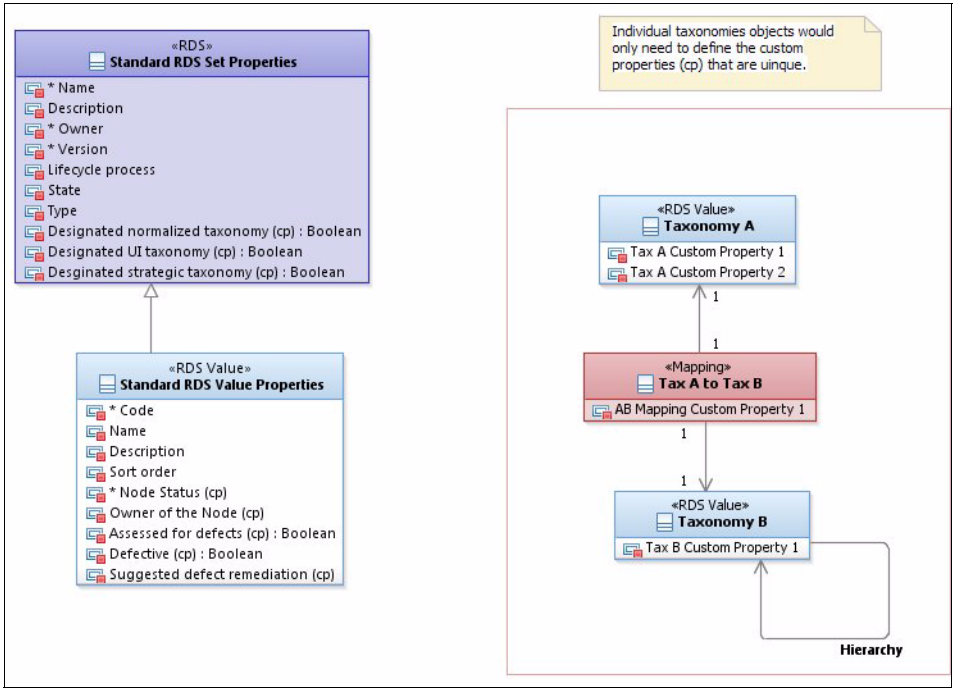


Figure 8-1 Example of the use of common set properties and documenting custom only

This approach helps to ensure consistency where appropriate, while allowing for the addition of information for each user group when necessary.

In addition to the information model, the following items should be in the adoption documentation:

- A matrix of the roles that are required and the LDAP groups that will be used to provide access

  The matrix shows the groups to which users must be added:

  - InfoSphere MDM Ref DM Hub Role access

    These groups allow a user to perform a specific role such as administrator, approver, reviewer, and stakeholder (reader). In InfoSphere MDM Ref DM Hub, these groups control which tabs and functions the user can perform.

  - Individual InfoSphere MDM Ref DM Hub Object access

    These groups provide control over the reference data sets and mappings that a user can see and modify.

- A matrix of the lifecycle processes to be used for each object

8.1.2 Managing ongoing changes

InfoSphere MDM Ref DM Hub provides a feature that allows properties to be added to a type after data is collected. Consider the following information when adding properties:

- Adding new property

  A “new” property can be added at any time without affecting the remainder of the data.

  You can initialize a new property manually. Or, you can export the reference data set or mapping, update the value of the set attribute to a default or specific setting, and import back into a new version of the object.

- Changing existing property

  Changing an existing property or deleting a property is more difficult. That property must have a null value in all values in a set or mapping.

  One option is to export the set, null all the values for the property you want to change, and then import the set back in. The reference data set type can then be changed with the modification to that property.

  The set can then be re-imported with the appropriate value in that property.

8.1.3 Engaging new user groups

A new user group (owners, consumers, or both) fits into one of three categories:

- Systems that want to consume only existing reference data

  In this category, the engagement process focuses mostly on these items:

  - Creating the appropriate documents of understanding with the set owners to ensure appropriate communication of changes to sets in the format that the system requires.
  - Documenting the system integration required and the format of any outputs.

- Owners of reference data who want to make it available for others to consume

  In this category, the business process and information modeling, described earlier, apply.

- User groups who want to consume both existing reference data and add more sets

  In this category, both of the previous categories must be considered.

InfoSphere MDM Ref DM Hub is a tool that helps to provide governance of reference data. The reference data needs of any new user group must be compared against the existing data available. The user group setup process must ensure that no duplicate instances are introduced.

8.2 New user group setup process

When a new user group of InfoSphere MDM Ref DM Hub is being set up, you must take appropriate steps.

Enabling new personnel to work with InfoSphere MDM Ref DM Hub involves several steps in planning, configuring, and customizing InfoSphere MDM Ref DM Hub to support the new users. Training is important. As much as possible, training should focus on the tasks that the individuals will perform and provide extra information to help them understand the broader capabilities of InfoSphere MDM Ref DM Hub and how their team is using it. The other major activity in setting up new personnel involves understanding the work they will be doing and planning their access control. In some InfoSphere MDM Ref DM Hub applications, the process of setting up a new user group is called a *boarding process*.

8.2.1 Access control groups and roles

InfoSphere MDM Ref DM Hub uses the WebSphere Application Server access control functions. Users must be identified to WebSphere Application Server with user ID and passwords. Users must be placed in groups known to WebSphere Application Server. Those groups are then granted two forms of access or capability in InfoSphere MDM Ref DM Hub.

LDAP is an appropriate complementary technology on which to build the group and individual access control capabilities. InfoSphere MDM Ref DM Hub can pull in LDAP-based information about individuals and groups so that the organization can more easily manage the access of those individuals and share that access control infrastructure with other systems.

In the remainder of this section, groups are used to provide access, without distinguishing the considerations for WebSphere Application Server managed groups versus externally managed groups that are accessed through LDAP protocols.

InfoSphere MDM Ref DM Hub employs two classes or concepts of capability. The two user capability concepts in the following list are controlled by groups:

- The ability to make changes on specific reference data sets and mappings. These groups are named in the *owner* property. The owner property is a standard reference data set level property. If a user is not in one of these groups, the user cannot modify any reference data set, however, the user can browse the reference data sets.
- The ability to perform certain functions or operations within InfoSphere MDM Ref DM Hub, for example, approve a reference data set, modify the properties (reference data type) for an reference data set.

Roles users might fulfill

The roles are based on the operations that the roles play in InfoSphere MDM Ref DM Hub. InfoSphere MDM Ref DM Hub includes the following roles:

- Reference Data Steward: This basic role is for someone who creates and maintains reference data values.
- Reference Data Set Approver: This role carries more responsibility than the steward because an understanding of the implications of approving new versions of reference data sets is required.
- Reference Data Expert and Reference Data Subject Matter Expert: The person or persons in this role have the most expertise in the information architecture of the reference data and knowledgeable of the master data in the systems that use the reference data.

The roles of users determine the function control groups in which they are included.

Reference data set access

If multiple teams use InfoSphere MDM Ref DM Hub to manage various sets of reference data, place the users in groups that align with their team or teams. Those groups should then be used in the “owner” property of the reference data sets with which the teams will be collaborating to manage so that users can perform operations or functions on the correct reference data.

Special access control considerations

Multiple InfoSphere MDM Ref DM Hub systems should be established to support an organization that needs orderly control over changes to reference and master data. In such situations, three or more InfoSphere MDM Ref DM Hub systems might be configured with related but distinct roles an in the following examples:

- RDM production system
- RDM (new) function test system
- RDM development system
- RDM training system

The access control groups on these systems vary. New users are likely enabled on the training system first. Users who are given more responsibility are likely getting that authority *also* on the training system first. The development system might be accessible only to the development team and selected users, including super users (the Reference Data Subject Matter Experts). Most users will have access to both the production system and the test system. Some access control groups will be shared across the training, test, and production systems. Some groups will provide super user access uniquely on the production and development systems. Many developers will not have access to the production system. Likewise, most users will not have access to the development system.

8.2.2 Customized training materials for each role defined

Providing proper training about the routine tasks that a user performs after deploying InfoSphere MDM Ref DM Hub is essential for smooth operation. The training topics described next are suggestions.

Basis training topics

Include the following topics in the basic training:

- Working with reference data sets:
  - Locating a reference data set in the list view of reference data sets
  - Locating a reference data set in the folder view of reference sets
  - Locating a value in an reference data set
  - Locating a hierarchy
  - Adding a value to an existing reference set
  - Editing an existing value or other properties in an reference set
  - Changing a hierarchy
  - Workflow steps for a reference set
- Working with mappings:
  - Adding a new value to value mapping

Intermediate training topics

Include the following topics in the intermediate training:

- Creating folders to organize reference data sets
- Moving reference data sets into folders

Advanced training topics

Include the following topics in the advanced training:

- Working with reference data sets:
  - Creating new reference data sets
  - Creating new versions of reference data sets and hierarchies
  - Managing translations
- Working with mappings:
  - Creating new mappings
  - Creating new versions of mappings
- Importing and exporting:
  - Importing reference data sets and hierarchies
  - Importing mappings
  - Exporting reference data sets and hierarchies
  - Exporting mappings

8.3 Reporting

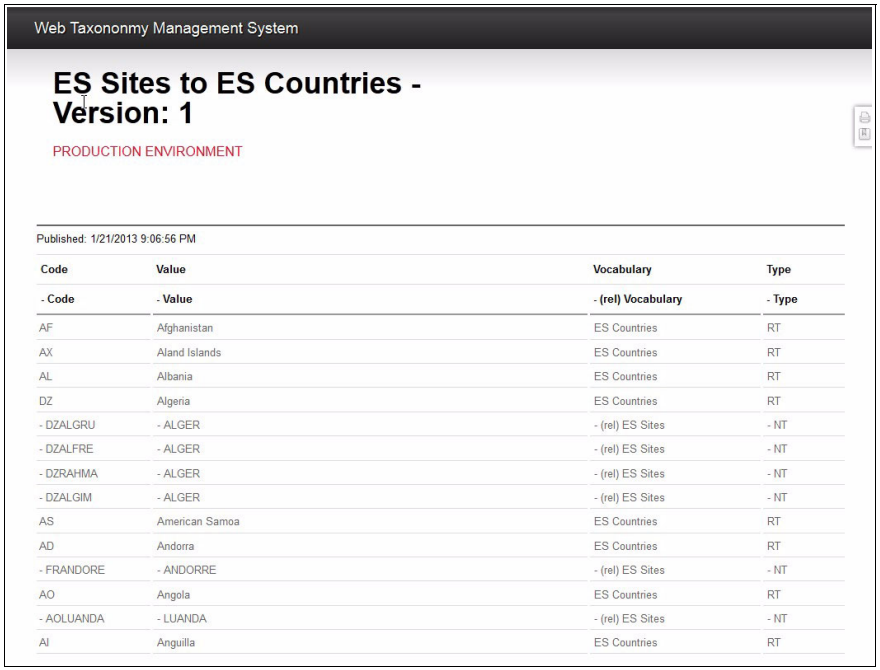
The InfoSphere MDM Ref DM Hub user interface (UI) provides access to individual objects such as reference data sets, hierarchies, and mappings. You can filter long lists to find and view the values and metadata managed. However, there are times when exporting and transforming the reference data into HTML reports is helpful, to be viewed in a browser or in comma-delimited files that can be opened in a spreadsheet.

Several reasons for providing reports are as follows:

- To provide access to the reference data information for users who will never have access to the InfoSphere MDM Ref DM Hub user interface.
- To assemble multiple objects for viewing, for example, showing the mapping between two or more reference data sets.
- For quality assurance; links to reports can be distributed to various users for review and feedback.
- To compare versions of complex object relationships.

Examples of typical reports are as follows:

- A single reference data set: Variations of this report might include the following information:
  - Node status information from a custom property (Figure 8-2)


Web Taxononmy Management System

ES Sites to ES Countries - Version: 1

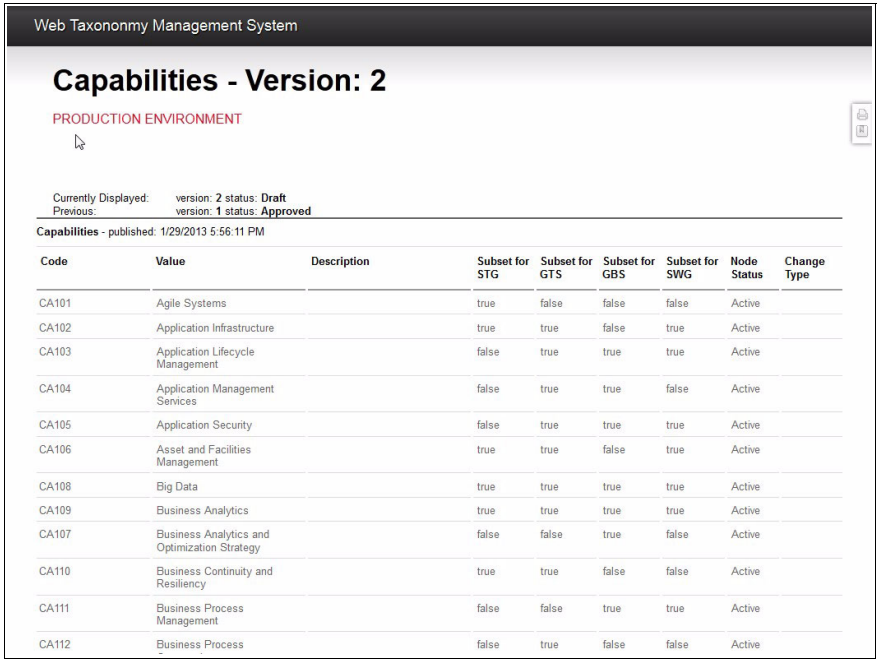
PRODUCTION ENVIRONMENT

Published: 1/21/2013 9:06:56 PM

Code	Value	Vocabulary	Type
- Code	- Value	- (rel) Vocabulary	- Type
AF	Afghanistan	ES Countries	RT
AX	Aland Islands	ES Countries	RT
AL	Albania	ES Countries	RT
DZ	Algeria	ES Countries	RT
- DZALGRU	- ALGER	- (rel) ES Sites	- NT
- DZALFRE	- ALGER	- (rel) ES Sites	- NT
- DZRAHMA	- ALGER	- (rel) ES Sites	- NT
- DZALGIM	- ALGER	- (rel) ES Sites	- NT
AS	American Samoa	ES Countries	RT
AD	Andorra	ES Countries	RT
- FRANDORE	- ANDORRE	- (rel) ES Sites	- NT
AO	Angola	ES Countries	RT
- AOLUANDA	- LUANDA	- (rel) ES Sites	- NT
AI	Anguilla	ES Countries	RT

Figure 8-2 Single reference data set with node status report

– Additional columns for custom properties (Figure 8-3)



Web Taxonomy Management System

Capabilities - Version: 2

PRODUCTION ENVIRONMENT

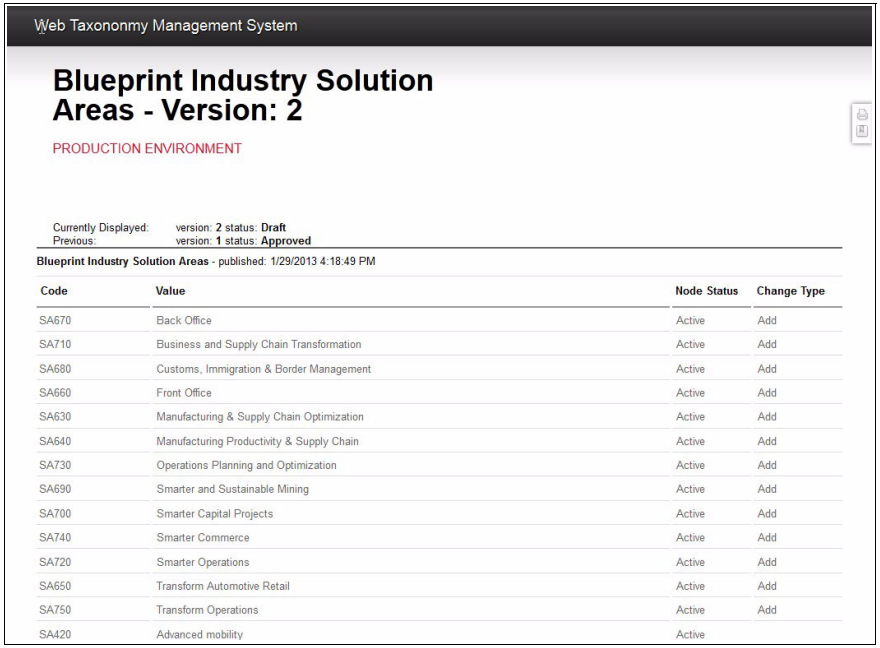
Currently Displayed: version: **2** status: **Draft**
Previous: version: **1** status: **Approved**

Capabilities - published: 1/29/2013 5:56:11 PM

Code	Value	Description	Subset for STG	Subset for GTS	Subset for GBS	Subset for SWG	Node Status	Change Type
CA101	Agile Systems		true	false	false	false	Active	
CA102	Application Infrastructure		true	true	false	true	Active	
CA103	Application Lifecycle Management		false	true	true	true	Active	
CA104	Application Management Services		false	true	true	false	Active	
CA105	Application Security		false	true	true	true	Active	
CA106	Asset and Facilities Management		true	true	false	true	Active	
CA108	Big Data		true	true	true	true	Active	
CA109	Business Analytics		true	true	true	true	Active	
CA107	Business Analytics and Optimization Strategy		false	false	true	false	Active	
CA110	Business Continuity and Resiliency		true	true	false	false	Active	
CA111	Business Process Management		false	false	true	true	Active	
CA112	Business Process		false	true	false	false	Active	

Figure 8-3 Report with custom attributes

– Version comparisons, with node change information (Figure 8-4)


Web Taxononmy Management System

Blueprint Industry Solution Areas - Version: 2

PRODUCTION ENVIRONMENT

Currently Displayed: version: **2** status: **Draft**
Previous: version: **1** status: **Approved**

Blueprint Industry Solution Areas - published: 1/29/2013 4:18:49 PM

Code	Value	Node Status	Change Type
SA670	Back Office	Active	Add
SA710	Business and Supply Chain Transformation	Active	Add
SA680	Customs, Immigration & Border Management	Active	Add
SA660	Front Office	Active	Add
SA630	Manufacturing & Supply Chain Optimization	Active	Add
SA640	Manufacturing Productivity & Supply Chain	Active	Add
SA730	Operations Planning and Optimization	Active	Add
SA690	Smarter and Sustainable Mining	Active	Add
SA700	Smarter Capital Projects	Active	Add
SA740	Smarter Commerce	Active	Add
SA720	Smarter Operations	Active	Add
SA650	Transform Automotive Retail	Active	Add
SA750	Transform Operations	Active	Add
SA420	Advanced mobility	Active	

Figure 8-4 Report with node change information, comparing versions

- A single reference data set with a hierarchy (Figure 8-5)

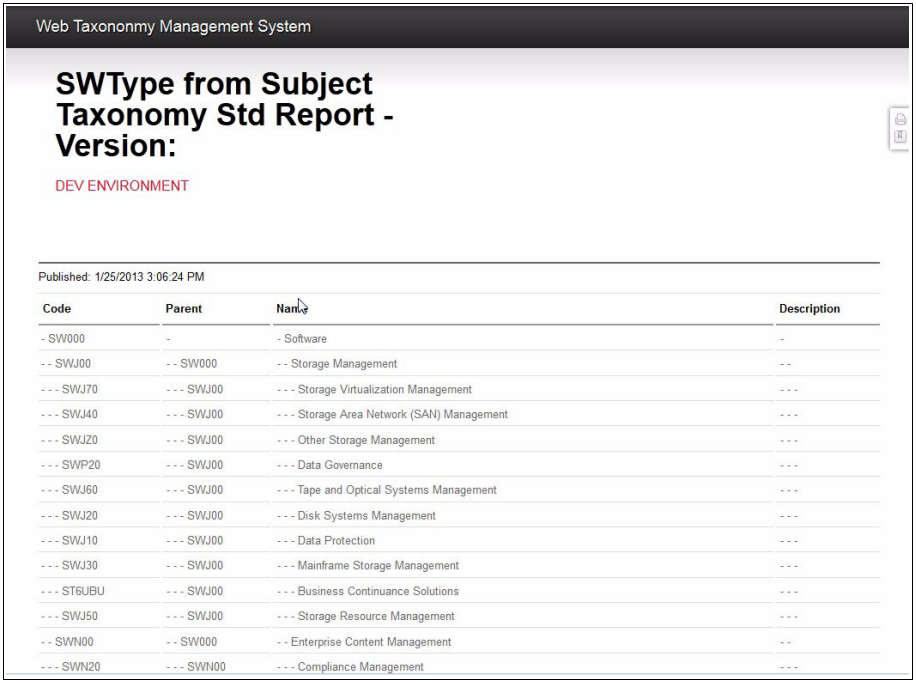
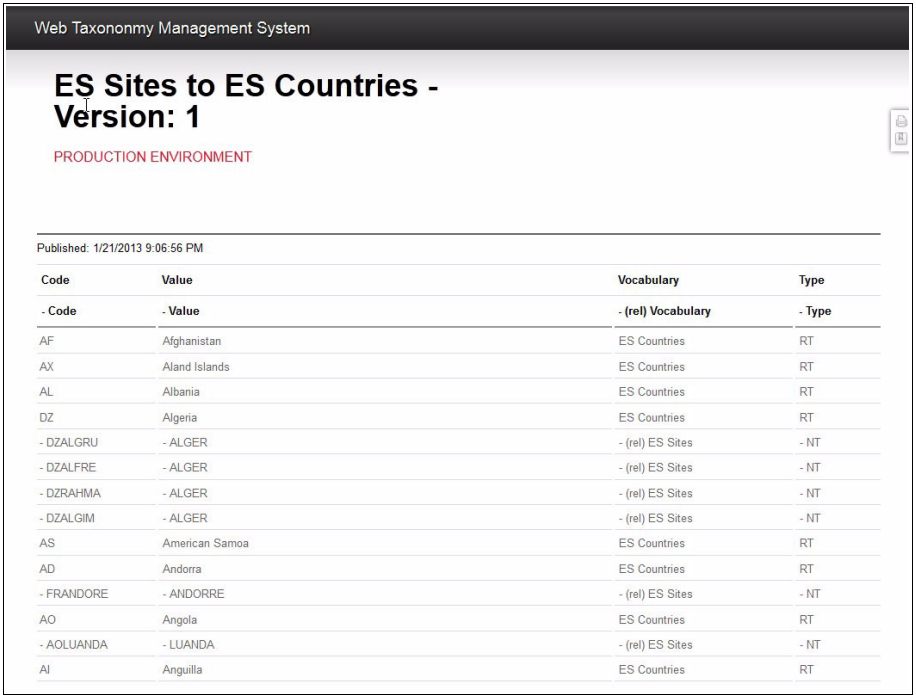


Figure 8-5 Reference data set with a hierarchy

- A mapping of two reference data sets, A to B (Figure 8-6)



Code	Value	Vocabulary	Type
- Code	- Value	- (rel) Vocabulary	- Type
AF	Afghanistan	ES Countries	RT
AX	Aland Islands	ES Countries	RT
AL	Albania	ES Countries	RT
DZ	Algeria	ES Countries	RT
- DZALGRU	- ALGER	- (rel) ES Sites	- NT
- DZALFRE	- ALGER	- (rel) ES Sites	- NT
- DZRAHMA	- ALGER	- (rel) ES Sites	- NT
- DZALGIM	- ALGER	- (rel) ES Sites	- NT
AS	American Samoa	ES Countries	RT
AD	Andorra	ES Countries	RT
- FRANDORE	- ANDORRE	- (rel) ES Sites	- NT
AO	Angola	ES Countries	RT
- AOLUANDA	- LUANDA	- (rel) ES Sites	- NT
AI	Anguilla	ES Countries	RT

Figure 8-6 Mapping report of two reference data sets (A to B)

- A mapping of three reference data sets, A to B to C, or A to B and C (Figure 8-7)

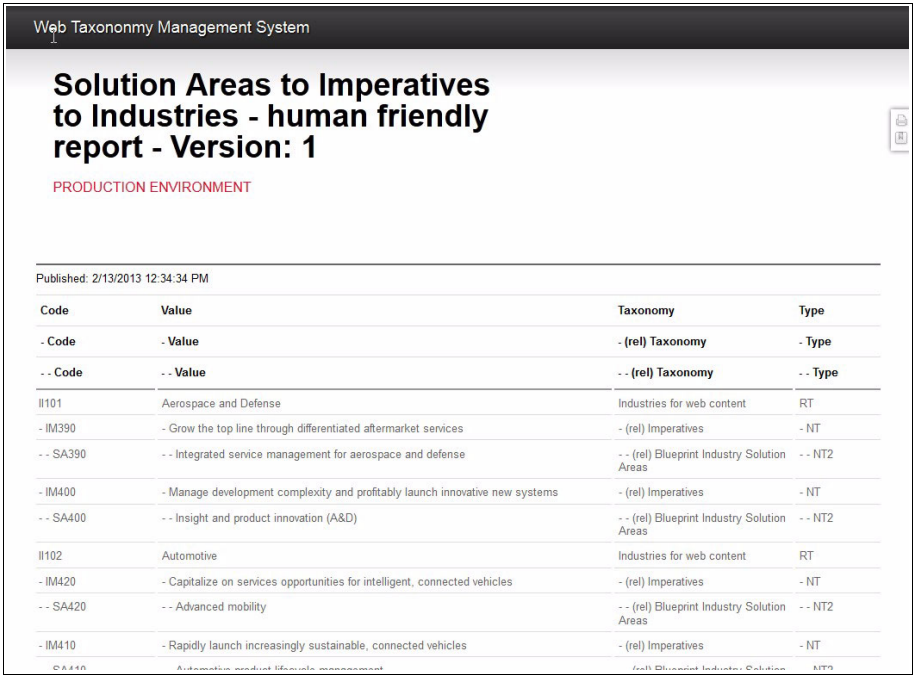


Figure 8-7 Mapping report of three reference data sets (A to B to C)

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publication provides additional information about the topic in this document. Note that this book might be available in softcopy only.

- *Smarter Modeling of IBM InfoSphere Master Data Management Solutions*, SG24-7956

You can search for, view, download or order this document and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- IBM InfoSphere Master Data Management Reference Data Management Hub information center:

  http://pic.dhe.ibm.com/infocenter/mdm/v10r0m0/index.jsp?topic=%2Fcom.ibm.swg.im.mdmhs.rdm.nav.doc%2FRDM_Introduction.html

- Metadata standard code lists:

  http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM&StrGroupCode=SCL&StrLanguageCode=EN

- Index of correspondence tables:

  http://ec.europa.eu/eurostat/ramon/relations/index.cfm?TargetUrl=LST_REL&StrLanguageCode=EN&IntCurrentPage=1

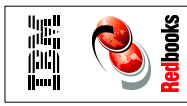
Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



A Practical Guide to Managing Reference Data with InfoSphere MDM Ref DM Hub

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

A Practical Guide to Managing Reference Data with IBM InfoSphere Master Data Management Reference Data Management Hub

Gain insight into governance, process, and security of managing reference data

Understand the RDM reference solution architectures

Learn RDM solution implementation

IBM InfoSphere Master Data Management Reference Data Management Hub (InfoSphere MDM Ref DM Hub) is designed as a ready-to-run application that provides the governance, process, security, and audit control for managing reference data as an enterprise standard, resulting in fewer errors, reduced business risk and cost savings.

This IBM Redbooks publication describes where InfoSphere MDM Ref DM Hub fits into information management reference architecture. It explains the end-to-end process of an InfoSphere MDM Ref DM Hub implementation including the considerations of planning a reference data management project, requirements gathering and analysis, model design in detail, and integration considerations and scenarios. It then shows implementation examples and the ongoing administration tasks.

This publication can help IT professionals who are interested or have a need to manage reference data efficiently and implement an InfoSphere MDM Ref DM Hub solution with ease.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8084-00 ISBN 0738438022