Using Rapid Prototyping to Develop a Data Mart
Abstract

Traditional methods of data mart development tend to slow the development velocity of projects as developers try to iterate and test changes to data models. The rapid prototyping method of development simplifies the collaboration between business analysts and business users on one side and developers on the other, speeding development. This article describes how to use a combination of techniques and Informatica tools to reduce time and expense in the development of data models.

Supported Versions

- Informatica PowerCenter 9.6.0, 9.6.1
- Informatica Data Services 9.6.0, 9.6.1

Table of Contents

Overview ............................................................... 2
Traditional Versus Rapid Prototyping Methods of Data Mart Development. ........................................... 2
Rapid Prototyping Example. ............................................... 3
Developing with the Rapid Prototyping Method ................................................................. 3
  Step 1. Create and Validate Data Source Views. ............................................................ 4
  Step 2. Create and Validate Mapping Specifications. ....................................................... 5
  Step 3. Add Transformations to the Logical Data Object Model. ........................................ 5
  Step 4. Iterate Changes to the Virtual Model. ............................................................ 5
  Step 5. Convert the Virtual Data Model to a Physical Schema for Production ....................... 5
How to Get Started. ............................................................................................................ 6
Use Cases. ........................................................................................................................ 6
Informatica Documentation .................................................................................................. 6

Overview

This article contrasts the traditional method of developing a data mart with the rapid prototyping method, which uses a combination of Informatica tools to quickly iterate data mart development. The article explains how to perform the rapid prototyping method using several Informatica tools. The article also gives additional resources so you can learn more about rapid prototyping and how to perform the individual tasks that make up this method.

Traditional Versus Rapid Prototyping Methods of Data Mart Development

You can reduce time and expense in the development of data marts by using the rapid prototyping method of development over the traditional method.

The traditional method of developing a data mart involves frequent interaction among business analysts and business users, developers, and database administrators (DBAs). The iterative nature of the prototyping process is time-consuming. This is partly because the efforts and communication of multiple staff must be coordinated. Worse, the process of viewing sample data takes time because it involves loading data into a physical schema to validate the data, with the DBA revising the physical schema prior to repeating the data validation process.
The rapid prototyping method significantly reduces the time to develop data marts by using Informatica Analyst and Informatica Developer to prototype data structures virtually rather than physically. Removing the need to manifest the data in physical data structures makes the necessarily iterative prototyping process faster. You can validate, change and review virtual views of the data in minutes or hours, not days or weeks.

The rapid prototyping method consists of the following steps:

1. In the Analyst tool, create and validate data source views.
2. In the Analyst tool, create and validate mapping specifications.
3. In the Developer tool, open mapping specification as a mapping, and add transformations.
4. In the Developer tool, developer and business analyst iterate changes to requirements and model.
5. In the PowerCenter client, convert the virtual data model to a physical schema.

Note: Perform this step only if requirements call for the model to move into production.

Rapid Prototyping Example

A health care company wants to make its data integration and business intelligence initiatives more agile so it can respond to the frequent changes in the health care industry.

A line-of-business finance director asks a business analyst to compare costs of various surgical procedures against recovery times. The analyst decides to create a data mart with all the relevant data, and creates requirements.

The analyst uses the Analyst tool to create a logical model that combines data from billing records from some of the organization's member hospitals. To create the logical model, the analyst creates mapping specifications that get data from each hospital's billing databases. Then the analyst and business user validate the output of the mapping specifications against requirements.

After the output is validated, the analyst notifies a developer that the project is ready for further development. Using the Developer tool, the developer opens the mapping specification as a logical data object. The developer adds transformations to enhance the model. Running the model produces the results specified in the requirements.

The finance director reviews the results and asks for data from all of the organization's member hospitals to be added to the model. The director also asks for changes to the range of dates from which data is used. The analyst revises the requirements and institutes some of the changes. The developer on the team creates additional transformations to provide the business user with the requested views. These changes can be made quickly, in minutes or hours.

After final validation, the developer exports the model to the PowerCenter client, where final changes are made to prepare the model for export to a physical data structure. The model is then exported, and a database administrator imports the model into a physical schema.

Following final loading of the data sources, the data mart is ready to go into production. The health care company uses the data mart to make faster and better decisions, updating the data mart not only with fresh data but with changes to business logic as requirements evolve.

Developing with the Rapid Prototyping Method

You can use the rapid prototyping method to quickly develop or revise data warehouse structures.
Step 1. Create and Validate Data Source Views

In the Analyst tool, a business analyst can quickly convert requirements into a preliminary view of data.

1. A business user stakeholder requests a new or revised data mart.
2. A business analyst creates a requirements document for the data mart.
   The requirements document might be in any form required by the business, such as a presentation or a MS Word document.
3. In the Analyst tool, the analyst registers data sources based on the requirements.
   The analyst can preview and profile data from data sources immediately. Doing this enables the analyst to discover source data anomalies that require cleansing. For example, some hospitals use "0" to signify "none," while others might use an older database schema that permits users to enter "no" instead of the number "0."
4. The analyst and business user validate initial views against requirements.
Step 2. Create and Validate Mapping Specifications

In the Analyst tool, a business analyst creates a mapping specification. A mapping specification is an asset that describes the movement and transformation of data from a source to a target. The mapping specification populates a target table with data that you can leverage to report on the target table.

1. The analyst uses the Analyst tool to create a mapping specification based on validated data sources.
   
   For example, the mapping specification can join data from the records of several hospitals. The mapping specification can also perform some initial data cleansing, such as normalizing "no" entries to "0."

2. The analyst runs the mapping specification.
   
   The results show the joined, normalized data from several hospitals.

3. The analyst validates the results of the mapping specification with the business user.

4. If necessary, the analyst revises the requirements for the data mart.

5. The analyst informs a developer on the project team that the mapping specification is ready for additional development.

Step 3. Add Transformations to the Logical Data Object Model

In the Developer tool, a developer adds transformations to the logical data object model. A transformation is a repository object that generates, modifies, or passes data. Transformations in a mapping represent the operations that the Integration Service performs on the data.

1. A developer opens the mapping specification in the Developer tool.
   
   The mapping specification resides in a Model repository that both the analyst and developer can access. The mapping specification appears in the Developer tool as a mapping in a logical data object model.

2. The developer uses the Developer tool to add transformations to the mapping.
   
   For example, the developer creates an Expression transformation that uses admission and discharge dates to calculate the length of a hospital stay.

Step 4. Iterate Changes to the Virtual Model

In the Developer tool, the developer iterates changes with the business user.

1. The developer runs the logical data object mapping and provides previews and profiles of the data to the business user who owns the original requirements.

2. Based on preview and profiles provided by the developer, the analyst or business user refines requirements.

3. As requirements are revised, the developer refines the model in the Developer tool.

4. Repeat until the business user is satisfied with the results and approves the data model.

Step 5. Convert the Virtual Data Model to a Physical Schema for Production

In the PowerCenter client, the developer prepares and exports the model. In the physical database, the database administrator (DBA) takes steps to move the database to production.

1. In the target database, the database administrator (DBA) makes changes to the physical schema.

2. In the PowerCenter Target Designer, the developer creates target definitions.
   
   The developer creates relational, flat file, or XML target definitions to match the physical schema administered by the DBA.

3. In the Developer tool, the developer exports the Logical Data Object Model to PowerCenter.
4. In the Power Center client, the developer adds any additional transformations that are necessary to prepare the model for loading to a physical target. For example, the developer might add a sequence generator transformation to generate primary keys so the structure of the model conforms to the structure of the data mart.

5. If the target definition is relational, the developer runs SQL code to create a target in a relational database. **Note:** For more information about this part of the process, read the How-To Library (H2L) article "How to Query an SQL Data Service in PowerCenter" ([https://mysupport.informatica.com/docs/DOC-12498](https://mysupport.informatica.com/docs/DOC-12498)).

6. In the PowerCenter client, the developer creates a session and a workflow, then runs the workflow.

7. The PowerCenter Integration Service loads data to the target.

8. The DBA moves the database into production.

**How to Get Started**

Consult additional resources on the [Informatica Global Customer Support](https://informatica.com) site for more information about rapid prototyping.

**Use Cases**

The following resources are available from the Informatica website:


**Informatica Documentation**

**Analyst tool**

To learn the basics of data sources, mapping specifications, and how to use them in the Analyst tool, see the [Mapping Specification Getting Started Guide](https://informatica.com).

To learn the full range of things you can do with mapping specifications, including applying rules, filters, aggregators, lookups, and joins, see the [Informatica Mapping Specification Guide](https://informatica.com).

To learn how to use general Analyst tool functionality, see the [Informatica Analyst Tool Guide](https://informatica.com).

**Developer tool**

To learn how to develop transformations for data models, see the [Informatica Developer Transformation Guide](https://informatica.com).

To learn how to create and use a SQL data service, see the [Informatica SQL Data Service Guide](https://informatica.com).

To learn how to use general Developer tool functionality, see the [Informatica Developer Tool Guide](https://informatica.com).

**PowerCenter**

To learn how to configure transformations in PowerCenter, see the [PowerCenter Transformation Guide](https://informatica.com).
To learn how to use general PowerCenter functionality, see the *PowerCenter Designer Guide* and the *PowerCenter Workflow Basics Guide*.

You can download a .zip archive, `0681_INFA_Userdocs_re_Rapid_Prototyping.zip`, containing all these manuals at the following URL: [https://mysupport.informatica.com/docs/DOC-12732](https://mysupport.informatica.com/docs/DOC-12732).

**Author**

Mark Pritchard  
Principal Technical Writer