A discussion of information integration solutions
November 2005

WebSphere® software

Profiling: Take the first step toward assuring data quality.

Contents

Why profile your data?

Analyst studies have shown that over 75% percent of data integration projects either overrun or fail. They either fail to deliver the required features, exceed their budget, or end due to cancellation before they are completed. Why the high failure rate?

The traditional approach to data integration follows these basic steps:

Step 1. Analyze the user's needs and build a target database specification. After interviews of the users, a grand scheme is devised for a database model that will answer all of the questions the user would like answered by the target application.

Step 2. Analyze the data sources available. A set of data sources from legacy systems, operational systems, and other sources is compiled and analyzed to determine their relevance to the target database. Documentation for the data sources may or may not be available, or it may be inaccurate. Samples of the source data are analyzed to detect the properties of the data.

Step 3. Build a set of source data to target database mappings. A plan for transforming the various data sources to the target is devised. Typically, this step is performed with an ETL tool or by hand-coded programs.

Step 4. Stage the data. The source data is loaded into a staging area where it can be massaged, cleansed, and manipulated into the form needed for the target data store. Data quality software may be deployed during this stage to standardize and link records.

Step 5. Load the data. The data is moved from the staging area into the target application. This step includes formatting the data for reporting While this approach appears logical, it contains flaws that contribute to the high failure rate for data integration projects: It is highly dependent on manual effort, and it makes a fatal assumption.

Don't assume "we know our data!"

The main weakness in the traditional data integration approach is that it makes the assumption that data required for an application is actually available from the data sources. Major corporations have spent millions of dollars on a data integration project, only to find out that the source data will not support the target model – whether they build the model themselves or it was defined by an enterprise application vendor. Because the process is made up of a series of disjointed steps usually executed manually by independent teams of programmers, the discontinuity between the steps often leads to disaster.

Organizations typically spend 80% of their project budget on Steps 3 and 4, staging and loading the data. Unfortunately the actual mechanics of specifying a set of source data-to-target mappings is a small part of the overall task of integrating multiple data sources. The real work lies in the necessary exercise of determining:

- *What exactly is in the source data?*
- *How is it organized?*
- *How can this data best be expressed in a target database schema?*
- *How can we map these sources and targets together? Usually, a lack of knowledge about the source data limits the possibilities for success in step 2, and eliminates any possibility of success in the subsequent steps.*

Most data integration projects that overrun their budgets or fail entirely do so because of a lack of understanding of the metadata. Without the use of automated metadata reverse engineering tools, developers are left to investigate the source data by hand. Documentation for the metadata of legacy systems is usually incomplete at best, or at worst, non-existent. The personnel needed for interpretation of the data often have left the company. Haphazard guesses are used instead of a complete analysis of content. This leads to a process where the integration of source data into the target data store is debugged far downstream in the development cycle. Rather than at design time, problems in the metadata are reflected too late in the process—in production systems.

A defect that isn't detected upstream (during requirements or design) will cost from 10 to 100 times as much to fix downstream. In the case of data integration, this translates into a significant financial loss for the enterprise that attempts to work with data without truly understanding the properties of the source data and to manually build target databases. The lack of tools that can detect problems in the extract/transform/load (ETL) process upstream is costing businesses a significant portion of their data warehousing budgets.

For success in data integration, start profiling

Poor data quality is the root cause of failure across a wide range of corporate initiatives. Profiling your source data up front generates significant benefits:

- *Reduces project risk*
- *Enhances ROI on a variety of enterprise projects, including business intelligence, enterprise application implementation, instance consolidation, single view of customer, master data management, and regulatory and compliance initiatives*
- *Validates business requirements as achievable or unachievable*
- *Ensures that disparate source data supports target requirements before the investment of time and resources in the data-integration development effort*
- *Pinpoints data problems early during the project cycle, substantially reducing costly testing and correction efforts*
- *Enables more accurate project planning of resources (people, skill sets, and time)*

Meet IBM WebSphere ProfileStage

WebSphere® ProfileStage™ brings automation to the critical and fundamental task of data source analysis, expediting comprehensive data analysis, reducing the time to value, and minimizing overall costs and resources for critical data integration projects. WebSphere ProfileStage profiles source data—analyzing column values and structures—and provides target database recommendations, such as primary keys, foreign keys, and table normalizations. Armed with this information, WebSphere ProfileStage builds a model of the data to facilitate the source-to-target mapping and automatically generates data integration jobs.

WebSphere ProfileStage allows users to integrate multiple disparate systems by providing a complete understanding of the metadata, and by discovering dependencies within and across tables and databases. Because the metadata is based upon the actual source data, its accuracy is typically 100%, reducing the project risk by uncovering integration issues before development begins. By leveraging this advanced data profiling capability, you can achieve a robust and reliable implementation that avoids critical and costly data integration problems. WebSphere ProfileStage can take the typical six to eight month project and deliver the same results in thirty to sixty days, a 70% average time savings.

Understand your source data

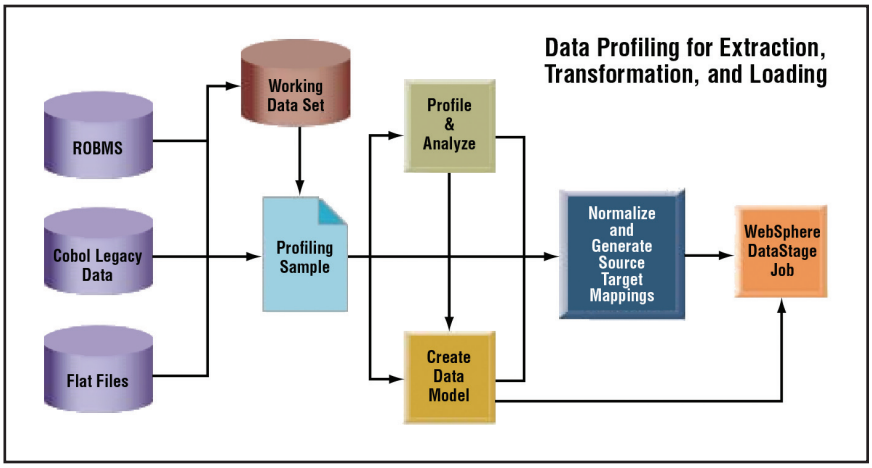
WebSphere ProfileStage makes no assumptions about the content of your data. You need only supply a description of the record layouts. WebSphere ProfileStage reads any source data, and automatically analyzes and completely profiles the data, so that the properties of the data (and hence the metadata) are generated without error. The properties include the tables, columns, probable keys and interrelationships among the data. Once these properties are known and verified, WebSphere ProfileStage automatically generates a normalized target database schema. The business intelligence reports and source-data-to-target-database transformations are all automatically specified as part of the construction of this target database.

After the source data is understood, the data integration project team is still faced with the daunting challenge of transforming that data into a relational database using a schema that makes sense. When using the traditional multi-step process, mistakes made in the design process are often manually debugged in the production systems. WebSphere ProfileStage automates this process, providing a proposal for the target database that can be easily edited by the user to gain the best possible end result.

Seven key processes lead to profiling success

The major processes involved in data profiling include:

- *Column Analysis*
- *Table Analysis*
- *Primary Key Analysis*
- *Cross-table Analysis*
- *Normalization*
- *Reporting & Data Definition Language (DDL) generation*
- *Integration with extract, transform & load (ETL) tools*



Column Analysis

Column Analysis examines all values for the same column to infer the column's definition and other properties such as domain values, statistical measures and minimum/maximum values. During Column Analysis, each available column of each table of source data is individually examined in depth. Many properties of the data are observed and recorded; some examples are:

- *Minimum, maximum, and average length*
- *Precision and scale for numeric values*
- *Basic data types encountered, including different date/time formats*
- *Minimum, maximum and average numeric values*

- *Count of empty values, NULL values, and non-NULL/empty values*
- *Count of distinct values or cardinality*

Additionally, Column Analysis makes certain inferences about the column's data—for example:

- *What data type, precision, and scale apply to the column*
- *Whether or not NULLs are permitted*
- *Whether or not the column contains a constant value*
- *Whether or not the column values are unique*

During Column Analysis, users create transformation notes and rules that will be used in the ETL process. This notation process contributes significantly to the project ROI.

Table analysis

Table Analysis is the process of examining a random data sample selected from the data values for all columns of a table in order to compute the functional dependencies for this table. Table Analysis seeks to find associations between different columns in the same table.

A functional dependency exists in a table if one set of columns is dependent on another set of columns. Each functional dependency has two components:

- *Determinant – A column or set of columns in a same table whose values can be used to predict the values in other columns in the table*
- *Dependent column – A column whose values are dependent on the values of the determinant column or columns in the same table. A column is said to be dependent if, for a given value of the determinant, the value of the dependent column is always the same.*

WebSphere ProfileStage not only displays functional dependencies supported 100% by your data, but it also displays functional dependencies poorly supported by your data. For example, during the dependency profiling step, WebSphere ProfileStage calculates the percentage of rows supporting a given functional dependency. It then pinpoints any problematic structures and displays the magnitude of the problem. Understanding if a functional dependency is broken is merely a first step; you need to understand just how broken it is to effectively scope and correct the problem, and WebSphere ProfileStage facilitates this important analysis.

Primary key analysis

Primary Key Analysis is the process of identifying all candidate keys for one or more tables. The goal is to detect a column or set of columns that might be appropriate as the primary key for each table. This analysis step must be completed before subsequent steps, such as Cross-table Analysis, can be performed.

Normally, Primary Key Analysis uses results from Table Analysis. Table Analysis identifies the dependencies among the columns of a table, and records each as an "aggregate dependency." Each row in the Aggregate Dependency table represents a single dependency for a given table. Each dependency has two components: a single column or set of columns (in the same table) that make up the determinant, and a set of columns (also in the same table) that are dependent upon the determinant. A set of columns is dependent on a determinant if, for a given value of the determinant, the value of the dependent columns is always the same. As you would then expect, a primary key determines all the values for the rest of the columns in the table. During Primary Key Analysis, one or more of the aggregate dependencies will become candidate keys. Subsequently, one candidate key must be confirmed by the user as the primary key.

Cross-table analysis

Cross-table Analysis is the process of comparing all columns in each selected table against all columns in the other selected tables. The goal is to detect columns that share a common domain. If a pair of columns is found to share a common domain, then this might indicate the existence of a foreign key relationship between the two tables, or it might simply indicate redundant data. These possibilities are examined during the subsequent Relationship Analysis step. Each row in the DomainsCompared table represents a pair of columns whose domains have been compared during Cross-table Analysis. A domain comparison is a bidirectional process, which might conclude that one column's domain is contained within that of the other column, or vice versa. Each row in the CommonDomains table represents the fact that one column (the "base" column) shares a common domain with another column (the "paired" column) in a different table. The common domain is noted only from the perspective of the base column; it makes no representation of whether or not a common domain also exists in the other direction.

A user may leverage the Cross-table Analysis within WebSphere ProfileStage to identify foreign keys across multiple tables. WebSphere ProfileStage first identifies the primary key for each table and then identifies identical or overlapping data across all tables or files.

Where identical or overlapping is identified, the user has the option to designated the primary key and corresponding columns as a foreign key relationship.

Normalization

Normalization involves computing a third normal form relational model for the target database. The user interface provides a "Normalization Wizard" that guides the user through the process of normalizing the target database model. The information gained through the analysis phases is used to aid the user in making intelligent decisions in the construction of the target data model. When WebSphere ProfileStage spots a candidate for normalization, the user is presented with a proposed normalization. The user can accept the proposed normalization, reject the normalization, or modify the model, as he or she desires.

The information gained from the three profiling phases is stored in the WebSphere ProfileStage Metadata Repository, a relational database of your choice containing information about all of the metadata in the project. This repository provides the basis for generating profiling reports, normalization, a data model of the target database, and source- to-target mappings.

Reporting & data definition language (DDL) generation

The profiling reports describe in detail the information gained from the profiling steps. These reports can be used as the basis for estimating the scope of the project, for obtaining signoff from end users and stakeholders, and investigating the true composition of the source data. The reports can be output to a variety of formats, including the user's screen, a printer, a file, email, Word, HTML and others.

The data model constructed can be exported to popular data modeling tools and in a variety of formats. The user can then examine the data model in a variety of combinations. If after examining the data model the user determines that changes in the target schema are necessary, he or she can adjust values in the Normalization Wizard or in the analysis phases.

New or revised models can be loaded into the WebSphere ProfileStage Metadata Repository and integrated into the project.

WebSphere ProfileStage supports generation of SQL in a variety of dialects, including SQL Server, ANSI SQL and Oracle. The DDL can also be generated in XML format.

Support for extract, transform & load (ETL) tools

Once the mappings have been confirmed, creating the ETL jobs for performing the creation of the target database requires merely the push of a button. This approach also supports mapping from sources to pre-defined targets with a drag-and-drop interface.

WebSphere ProfileStage automatically generates the code for WebSphere DataStage® job transforms. The steps outlined above convert a non-normalized source database into a fully normalized target database. No programmer time is necessary to build the WebSphere DataStage jobs for these basic transformations. Since the WebSphere ProfileStage approach derives the data model for the target database from the information stored in the WebSphere ProfileStage Metadata Repository, the source-to-target mappings are automatically computed.

WebSphere ProfileStage provides an intuitive and efficient interface for modifying the source-to-target mappings, including the addition of columns, transformation, and summarization. Through tight integration with other tools within the IBM WebSphere Information Integration portfolio, WebSphere ProfileStage provides the environment for both specification building and ETL generation. After the user confirms the validity of the source-to-target mappings, WebSphere ProfileStage automatically generates the set of jobs to jumpstart the WebSphere DataStage processing. The data can start from a variety of sources, including all ODBC-compliant relational databases, COBOL legacy data, and ASCII flat files. The end of the process is the generation of a WebSphere DataStage job to migrate the data.

Avoid the pitfalls of the traditional manual process

You can avoid the pitfalls of the traditional manual process by merging the traditional steps into an integrated process, with the addition of enlightened inferences from the metadata and a stable delivery environment. Some of the advantages of using WebSphere ProfileStage include:

- *The correct metadata is generated from what actually exists in your data, rather than from the wishful thinking of the developers. WebSphere ProfileStage ensures a specification that is correct by definition.*
- *Invalid data is spotted and rectified early in the project through the specification of the source-data-to-target-warehouse processes.*
- *Accurate documentation for the source data is automatically created from reports in the system and verified by the user. The documentation is automatically generated and reflects the actual data present in the source system.*
- *There is no dependence upon the programmers who developed the applications that produced the source data. The only resource that is needed is access to the data.*
- *Keys are inferred from what is actually present in the data.*
- *Field types are inferred from what is actually present in the data.*
- *The true range of domain values for coded fields is generated and mapped as part of the specification.*
- *A normalized target database is generated automatically, eliminating costly mistakes in data modeling.*
- *Dependency relations are inferred from what is actually present in the source data.*
- *The target database definition, including foreign and primary keys, normalizations, and correct data types, is automatically generated.*
- *Manual data conversion tasks are reduced through the generation of the WebSphere DataStage job.*

The productivity achieved by utilizing WebSphere ProfileStage reduces the staffing requirements for a data integration project. This is not to say that using WebSphere ProfileStage eliminates any possible problems in the process. Analysts and developers still have to make informed decisions and apply their talents to the problems. But by eliminating the vast array of pitfalls that traditional multi-step data integration projects encounter, you can dramatically reduce the time and effort needed for the project. Customer implementations across multiple industries have shown that WebSphere ProfileStage can take the typical six to eight month project and deliver the same results in thirty to sixty days. WebSphere ProfileStage contributes to a favorable result, identifying serious problems with the source data early in the process when correction is far less costly in time and budget.

IBM WebSphere Information Integration

Organizations face an information challenge. Where is it? How do I get it when I need it in the form I need? What does it mean? What insight can I gain from it? Can I trust it? How do I control it? The list goes on, and the challenges grow unceasingly if businesses cannot ensure that they have access to authoritative, consistent, timely and complete information.

The IBM WebSphere Information Integration platform integrates and transforms any data and content to deliver information you can trust for your critical business initiatives. It provides breakthrough productivity, flexibility and performance, so you and your customers and partners have the right information for running and growing your businesses. It helps you understand, cleanse and enhance information, while governing its quality to ultimately provide authoritative information. Integrated across the extended enterprise and delivered when you need it, this consistent, timely and complete information can enrich business processes, enable key contextual insights and inspire confident business decision-making.

Addressing key questions about the source data at the beginning of any data integration project, WebSphere ProfileStage is an integral part of the WebSphere Information Integration portfolio.

For more information

For more information about WebSphere ProfileStage or the WebSphere Information Integration portfolio, contact your IBM marketing representative or IBM Business Partner, or visit: ibm.com/software/data/integration

IBM Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
12-05

GC-18-9728-00