Exploring data lineage: Get a complete picture of your data flows

Analyze enterprise data movement through extended lineage in IBM InfoSphere Metadata Workbench

Hayden Marchant

January 07, 2010

Explore the new capabilities of the lineage functions of the IBM InfoSphere™ Metadata Workbench 8.1.2, including how to use extension mappings and data sources in the Metadata Workbench to extend lineage and how to manage them through the command-line interface. This article addresses both current and potential users of Metadata Workbench, as well as IT architects evaluating solutions to support compliance requirements, such as Sarbanes-Oxley, Basel II, or internal corporate governance programs.

Introduction

Recently, increasing numbers of government regulations and internal corporate data governance initiatives have prompted IT organizations to implement automated processes to provide accurate data lineage on key enterprise information assets, such as business intelligence reports. IBM InfoSphere Information Server is a data integration software platform that helps organizations derive value and understanding from complex, heterogeneous information in their organization. The platform is composed of various components to understand, cleanse, transform, and deliver information. Following is a list of the components in Information Server 8.1:

- Metadata Workbench
- Business Glossary
- FastTrack
- Information Analyzer
- QualityStage
- DataStage
- Information Services Director

To achieve a high degree of active sharing and reusability of information, all components of Information Server access and persist their metadata through a common metadata layer consisting of a metadata server and a shared metadata repository.
InfoSphere Metadata Workbench plays an integral part in Information Server by providing data lineage and impact analysis reports on various assets stored in its metadata repository, including business intelligence reports, databases, legacy data assets, and so on. One of the core features in Metadata Workbench is its built-in support to automatically deduce linkages among metadata that Information Server components create or import.

Using these functions, business users can get answers to questions such as "What are the sources for this report that I'm looking at?" and "I don't trust the data in this report. Which transformations were applied to the data on which this report is based?" The answers to such questions are critical to the compliance laws currently in place in many industries, including finance, banking, and manufacturing. Similarly, IT analysts can get answers to questions regarding impact of change, including "Which assets and processes would be affected if I change the constraints on this column?" and "Can I decommission this database?"

**Lineage reports**

Out of the box, Metadata Workbench provides various lineage reports about assets, including:

- Imported databases (tables, columns) and table-structured files and their fields
- DataStage and QualityStage ETL jobs and contained stages
- Imported BI reports and report fields

Figure 1 shows an example data transformation and movement process in which data from two sources (RUT_UT and RUT_EMEA) are being moved by two separate DataStage ETL jobs into a staging database (CRT_STAGING). From there, another DataStage ETL job moves the data into a data warehouse called ENT_DWH. Finally, a fourth DataStage ETL job (TO_MART) builds out a data mart (MART) that three Cognos BI reports are then using.

**Figure 1. Sample data flow topology in enterprise**

In this particular data flow, Metadata Workbench supports all participating components (jobs, database tables, and BI reports). Therefore a data lineage report started on one of the BI reports, such as the SalesPerQtr report, would successfully report the entire flow all the way back to the two data sources. Listing 1 shows how the results of a data lineage report would look for the SalesPerQtr report.

**Listing 1. SalesPerQtr report**

```
MART
TO_MART Jobs
ENT_DWH
STG_TO_DWH Jobs
CRT_STAGING
US_TO_STAGING Jobs
EMEA_TO_STAGING Jobs
RUT_USRUT_EMEA
```
For the example, all metadata describing how the data moves from one place to the next is stored in the metadata repository, and thus enables Metadata Workbench to build out the complete lineage report by deducing the chain of source-to-target relationships.

In an ideal world, a customer would only need one tool, Information Server, to do data transformation and movement. However, in almost all cases, a customer uses more than one method or application to perform data transformation and movement. Examples include stored procedures, other vendors' ETL systems, ERP systems, and homegrown applications. The fact that metadata for these processes and systems is not natively stored in Information Server's repository impacts a customer's ability to see the complete data flow that Metadata Workbench data lineage reports provide.

Incomplete data lineage reports

To showcase this impact, the next example uses a different data transformation and movement process. Instead of a DataStage job, the data warehouse is now populated by stored procedures STG_TO_DWH, as shown in Figure 2.

Figure 2. Sample data flow topology in enterprise with non-Information Server processes

If you run a data lineage analysis on the BI report SalesPerQtr, you get the following result:

Listing 2. SalesPerQtr BI report

```
MART
TO_MART Jobs
ENT_DWH
```

The lineage stops at ENT_DWH because the Metadata Workbench does not have enough information available to know that the stored procedures are moving data from CRT_STAGING to ENT_DWH. An IT analyst familiar with the Metadata Workbench might realize this and consequently run another lineage from CRT_STAGING. Listing 3 shows the results:

Listing 3. Running lineage from CRT_STAGING

```
US_TO_STAGING Jobs
EMEA_TO_STAGING Jobs
RUT_USRUT_EMEA
```

So when information from the staging database is moved to the data warehouse through stored procedures, there is a gap in the lineage. The stored procedures or other external applications that manipulate data prevent the user from seeing complete data lineage reports.
With Metadata Workbench 8.1.2, these gaps in the data flow analysis are closed by describing data-movement processes that occur outside the Information Server through a method called extended data lineage. The rest of this article describes how to obtain accurate extended data lineage reports based on the example scenario in Figure 2.

**Using extended data lineage**

With extended data lineage, you can track the flow of data across your enterprise, no matter which process or application is moving or transforming your data.

For the Metadata Workbench, an external process is a black box with known endpoints (sources and targets). Extension mappings are essentially source-to-target mappings that describe data movement from one or more sources to one or more targets. The extension mappings are linkage points to fill the gap between Information Server assets and external processes (see Lineage reports). By using extension mappings, data lineage analysis can be performed on data flows external to Information Server.

**Extension mapping documents**

Extension mappings are organized into extension mapping documents. Each row in a mapping document contains a source-to-target mapping usually created for a particular process or routine. Mapping documents can be created and managed in a logical way that maps to how the data transformation logic is partitioned in the enterprise.

Figure 3 shows the corrected data lineage report after creating an extension mapping document to track the lineage through the example stored procedure STG_TO_DWH.

**Figure 3. Data lineage report of ETL process with stored procedure corrected with extension mappings**

**Custom attributes**

In addition to the predefined fields, you can add extra custom fields to the extension mappings by defining custom attributes on the mappings. See Related topics to find details about these capabilities.

**Creating extension mapping documents**

Creating, importing, and deleting extension mapping documents are administrative tasks that require an Information Server user with Metadata Workbench administrator access.

Create extension mapping documents in the Metadata Workbench by following these steps:
1. Click the **Advanced** tab in Metadata Workbench on the left control bar. This tab is visible only for a Metadata Workbench administrator.

2. Select **Create Extension Mapping Document** in the Extension Mapping Documents section. Figure 4 shows the window that holds details of this new extension mapping document. The window shows 2 tabs: Properties and Mappings.

**Figure 4. Properties tab of Create Extension Mapping window**

![Properties tab of Create Extension Mapping window](image)

3. On the Properties tab, enter a name for the mapping document. This should be descriptive enough to be able to identify it later.

4. Optionally enter a type. This is a free-text field that can be used to provide categorization of extension mapping documents. For example, if documents are created to describe stored procedures, mainframe applications, or Web services, it would be useful for the Metadata Workbench user to search for extension mapping documents by type or by name. Alternatively, you could also categorize by business domain, so types such as finance, human resources, logistics, and product could also be useful.

5. Optionally enter a description of the extension mapping document. This might contain information such as the source systems that this mapping document is describing, any relevant notes that would be useful in the future, and so on.

**Creating mappings**

On the Mappings tab, enter extension mappings that describe the actual data movement. Each extension mapping describes the data movement of sources to targets.

The Mappings tab is arranged in a table format in which each row in the table is an extension mapping. The following table details each field of an extension mapping and how to complete them.

**Table 1. Extension mapping fields**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Purpose</th>
<th>Completion instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Informative name of the mapping to uniquely identify this mapping within the mapping document.</td>
<td><em>(Optional)</em> Click the cell and enter text.</td>
</tr>
<tr>
<td>Sources</td>
<td>One or more sources of this mapping. See <strong>Defining sources and targets</strong></td>
<td><em>(Required)</em> Figure 5 shows how the asset selector window appears when you click the source field, in which you can locate the required assets to use as the source for the mapping.</td>
</tr>
</tbody>
</table>
Figure 5. Selecting a source or target

Editing mappings

After entering a mapping, additional rows can be added to the table to add new mappings. Once all mappings have been entered, the new extension mapping document can be saved by clicking Save, as shown in Figure 6.
Defining sources and targets

Sources and targets can be of many different asset types. The assets are selected from a list of the available assets in the Information Server metadata repository. The available asset types are:

- Database assets (database, schema, table, column)
- Data Files fields
- DataStage assets (jobs, stages, stage columns)

It is also possible to select extended data sources, which represent assets such as applications, methods, and parameters. Extended data source enable things like Web services or API calls to be sources or targets of a mapping.

It is important to ensure that all the required assets are already imported and are up to date. Although a typical mapping has one source and one target, there are scenarios where a mapping can have multiple sources or targets. For example, a mapping with multiple source tables and a single target table might describe a process of selecting rows from multiple tables (using SQL joins or unions) and of inserting the result into a single table.

Similarly, a mapping that describes the splitting of an address field into Line1, Line2, City, ZipCode would have one source column and four target columns.

These examples show that mappings can be described at different levels of granularity. For example, a stored procedure can be described on a coarse-grained level where sources or targets point to database tables. Alternatively that stored procedure can be described on a fine-grained level through numerous mappings, where sources and targets point to table columns. The decision of what level of granularity to map depends on the resources available to document the relationships and on the level of lineage required. If an enterprise has 10,000 stored procedures on which lineage reporting is required, it might be more practical to map at the table level, as opposed to creating hundreds of thousands of column mappings. However, if the level of required lineage reporting is only to the level of which database a report is sourced from, then there might not be any need to document to a finer level of granularity.

Finding and viewing mapping documents

You can view extension mapping documents in any of the following ways:

- On the Discover tab, click the **Find** link, and select **Extension Mapping Document** as the Asset Type.
- On the Discover tab, click the **Query** link, and create a query to find the extension mapping document.
- Search for an individual extension mapping by doing a find or query for *extension mapping* in the Asset Type. For this type of search, giving meaningful names to individual mappings makes searching for mappings easier.
- On the Discover tab, select **Extension Mapping Document** from the Additional Types dropdown, and click Display to view all the extension mapping documents.
- On the Advanced tab, click **Manage Extension Mapping Documents** to view all the extension mapping documents.

An example of the results of a search is shown in Figure 7.

**Figure 7. Extension mapping document search results**

Once an extended mapping document appears on the displayed list, you can view it by clicking on the related mapping document. Figure 8 shows how the extended mapping document opens in view mode.
Managing extended mapping documents

This section describes how to edit extended mapping documents and how to assign terms and stewards.

Editing extended mapping documents

To manually modify mappings in an extended mapping document, a Metadata Workbench administrator can access the edit window for an extended mapping document in one the following ways:

- In the Find/Query results window, right-click the extended mapping document and select **Edit**.
- In the view mode window of an extended mapping document, click **Edit** in the right-hand panel.

Both of these methods open the edit window for the extended mapping document. This is the same window you see when you create an extended mapping document. You can change all properties on the Properties tab, and you can make any changes to the mappings on the Mapping tab.

Assigning terms and stewards

You can assign terms and stewards to extension mapping documents. Understanding a mapping document is a lot easier when it has a shared business meaning and when there is someone in the organization who is accountable for it.
Figure 9 shows that you can select multiple extension mapping documents and click the **Assign Term** link. The Business Term select window opens. After you select the term, it is assigned to all of the selected extended mapping documents.

Similarly, you can assign a steward to extension mapping documents. You can select one or more extension mapping documents and then click the **Assign Steward** link. The Steward select window opens. After you select the steward, it is assigned to all of the selected extended mapping documents.

**Figure 9. Assigning terms and stewards**

Importing extension mapping documents

Metadata Workbench provides an automated way to create extension mapping documents by enabling import of mapping documents from csv files. Instead of creating and maintaining many extension mapping documents in the edit window (Figure 5), you can import them from csv files into the Metadata Workbench. Many businesses require hundreds or thousands of extension mapping documents, and importing these files greatly accelerates the deployment of the lineage capabilities. Because csv format is well-documented and simple to comply with, IT developers can easily generate Workbench-compliant mapping document files from existing spreadsheets, databases, and other sources.

A Metadata Workbench administrator can import extension mapping documents by doing the following:

1. In the Advanced tab, click **Import Extension Mapping Documents**.
2. Select one or more csv files or directories by clicking the **Add** button.
3. If the sources or targets in the csv files contain relative paths to the assets to be mapped, add the source prefix in the Source field and add the target prefix in the Target field. For example,
if the extension mapping document contains mappings between columns in a server \textit{SVR},
database \textit{DB1}, schema \textit{MY\_SCH}, and table \textit{MYTBL}; the csv file could relate to the columns by full
name, such as \textit{SVR\_DB1\_MY\_SCH\_MYTBL\_COL1}. Or, it can reference them just as \textit{MYTBL\_COL1}, in
which case the Source field in the import dialog should contain the prefix \textit{SVR\_DB1\_MY\_SCH}.

Extension mapping documents that are imported can be re-imported by simply importing the same
file with any changes. This allows for the extension mappings to be maintained externally, and
whenever desired, the new version of the mappings can be imported into the metadata repository.

\textbf{Extension mapping document file format}

Extension mapping document files are in csv format with each extension mapping being
represented by one row in the csv. On each row, the values of the mapping fields are entered as
described in \textbf{Table 1}. The file must comply with the following rules:

\begin{itemize}
  \item The file must be in UTF-8 or ANSI encoding.
  \item The first line must be a header with the following columns (case and order irrelevant):
    \begin{itemize}
      \item Name
      \item Source columns
      \item Target columns
      \item Rule
      \item Function
      \item Specification description
    \end{itemize}
    \begin{itemize}
      \item (Any other columns are ignored)
    \end{itemize}
  \item The separator must be a comma.
  \item Dot notation identifies a source or target asset. The identity should use the same names as
    they appear in the metadata repository. Using the Metadata Workbench to see the names of
    the assets helps to accurately name the assets. Each asset type requires a different notation
    as follows:
    \begin{itemize}
      \item Schema is identified as \{server\}.\{database\}.\{schema\}
      \item Table is identified as \{server\}.\{database\}.\{schema\}.\{table\}
      \item Column is identified as \{server\}.\{database\}.\{schema\}.\{table\}.\{column\}
      \item File is identified as \{server\}.\{data\_file\}
      \item File field is identified as \{server\}.\{data\_file\}.\{data\_file\_field\}
    \end{itemize}
  \item Multiple sources and targets should be surrounded by double-quotes ("), with each entry
    separated by a comma.
  \item Rule, function, and specification description values should be surrounded by double-quotes
    ("") if the values contain a comma.
  \item Content can be in any language, as long as the encoding is in UTF-8.
  \item If any of the sources or targets cannot be located in the metadata repository, the mapping in
    which the source is referenced will be ignored.
\end{itemize}

Listing 4 shows an example csv file.
## Listing 4. Sample extension mapping document csv file

<table>
<thead>
<tr>
<th>Line</th>
<th>Source Columns</th>
<th>Target Columns</th>
<th>Rule</th>
<th>Function</th>
<th>Specification Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;reg map&quot;, DB2.CRT_STG.HAYDEN.SALES.CITY, DB2.CRT_STG.HAYDEN.SALES.SALES_COUNTRY</td>
<td>DB2.ENT_DWH.HAYDEN.SALES_AGGR1A.REGN</td>
<td>conc city and country, based on IS...</td>
<td>REGN = CITY+&quot;[&quot; + SALES_COUNTRY+&quot;&quot;&quot;]&quot;</td>
<td>&quot;speak to Bob for more info&quot;</td>
</tr>
<tr>
<td>2</td>
<td>DB2.CRT_STG.HAYDEN.SALES.RETURNS_SALES, DB2.CRT_STG.HAYDEN.SALES.TOTAL_SALES</td>
<td>DB2.ENT_DWH.HAYDEN.SALES_AGGR1A.TOTAL_SALES_DOLLAR</td>
<td>just subtract the RETURNS from the TOTAL_SALES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Importing from FastTrack mapping specifications

You can also import extension mapping documents from FastTrack (see Related topics) by exporting FastTrack mapping specifications to csv files and then importing those files into Metadata Workbench. You can do this because FastTrack mapping specifications and extension mapping documents share the same file format. Content can be edited in a spreadsheet-type application from which the file can be saved as a csv file. These applications generally deal elegantly with adding double-quotes when needed, as well as providing clear, tabular view of mappings.

An imported extension mapping document is automatically given a name based on the file from which it was imported. When importing a large number of mappings, a progress window notifies you of progress and any errors during import.

## Monitoring status

Details of all administrative operations on extension mapping documents are recorded in the Information Server. You can view these details by creating a logging view in the Web console for Information Server by following these steps:

1. Log in to the Web console for Information Server.
2. Navigate to the Administration tab, and open the Log Management section.
4. From the Categories list, select WORKBENCH-EXTENDED-LINEAGE as shown in Figure 10, and select Save and Close.
5. Use this log view to see extension mapping document activity, as shown in Figure 11.

Using the command-line interface for extension mapping documents

You can import and delete extension mapping documents through the Metadata Workbench Web interface or from the command-line interface. With the command-line interface, you can schedule frequent imports of extension mapping documents and deletions of mappings that are no longer relevant without need for human intervention.

In order to access the command-line scripts, install Information Server Manager before installing Metadata Workbench 8.1.2. The command-line scripts are available on both Windows® and
UNIX®-style operating systems on a client installation of Information Server. The command line interface is located at `<InformationServer_Root>/Client/istools/cli`.

Import and delete of extension mapping documents is performed through the `istool` application. In this way, scripts can be written that import or delete extension mapping documents with whatever logic the administrator requires, and, of course, these scripts can be executed by any third-party scheduler.

Listing 5 shows how a whole directory of extension mapping document files can be imported into the Metadata Workbench with a single command.

**Listing 5. Sample command-line calls to import**

```bash
cli>istool workbench extension mapping import -domain is1 -username isadmin -password ****
-filename /mappings/entanys/eg2 -output /mappings/reports/output.log
```

Full details of the syntax and examples of the command-line are available in the Information Server Information Center (see Related topics).

**Conclusion**

The extended lineage feature in the Metadata Workbench 8.1.2 provides IT analysts with the tools to get accurate lineage and impact-analysis reports over their whole enterprise and to manage the extension mappings in a clear and coherent manner. These new functions can help alleviate the high cost that IT organizations have been paying in recent years for not having the tools to understand their data fully.

**Acknowledgments**

The author would like to thank Beate Porst, Brian Byrne, Ernie Ostic, Ilan Prager, and Laurie Marr for their valuable feedback and reviewing of this article.
Related topics

- Read Integrating heterogeneous metadata (developerWorks, 2009 Jul) for more about data governance, data lineage, and impact analysis with the InfoSphere Metadata Workbench.
- Access user documentation about extended data lineage in the IBM Information Server Information Center.

© Copyright IBM Corporation 2010
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)