Streamlining the Data Flow Process with ETL Best Practices and Tools

What is your (business) question?
Do you need to...

Extract, Transform, and Load

Why ETL Tools

How We Use ETL Tool

Start an ETL Project

Conclusion

How You Can Use It

That's how we use it.
How would you use it?

Think about it.
Explore your options.
Contact us.

Higher ETL Data Warehouse Terms
The Data Warehousing Toolkit
Streamline your ETL efforts with...
Streamlining the Data Flow Process with ETL Best Practices and Tools

Presented by:
Carrie Shumaker and Fusen Li
from ITS Data Delivery
What is your (business) problem?

Do you need to...

- Build a robust data warehouse to ensure informed business critical decisions;
- Mix and match data from different sources in different formats without building a comprehensive data warehouse;
- Convert data:
  - identify inconsistencies and incompatibilities of your old data,
  - convert or cleanse
  - load into your new system.

Then, you are using ETL!
Do you need...

Build a robust data warehouse to ensure informed business critical decisions;

Convert data
- identify inconsistencies and incompatibilities of your old data,
- convert or cleanse
- load into your new system.

Mix and match data from different sources in different formats without building a comprehensive data warehouse.
More examples of ETL functions:

**EXTRACT**
- Join data from disparate sources / formats (XML, Excel files, different databases)
- Export data from different databases

**TRANSFORM**
- Regroup data
- Sort and de-dup data
- Create new columns/fields
- Apply unit-specific business rules
- Filter
- Summarize
- Move data from one form to another (XML, text files, etc.)

**LOAD**
- Create outputs in different formats (XML file, Excel file, report, database format)

and load it into another database, data mart or warehouse, or another operational system.
Start an ETL Project

Start

Design Target and ETL

- Deliver to the target! It is a database or file or report? Do we need to redesign a database? If so, should it be a relational or dimensional? (Do we need to break change library or not?)

- Examples: Assume we already have dimensional modeling and cube/warehouse, we will need to add Oracle data and performance data to it.

DO NOT!! assume you need a specific solution (dashboard, cube, etc.) (no matter how badly someone wants it)
Sounds like a ETL project

Start
Definition of business need and action

(What is the question, and how will we respond to the answer?)

Business goal: How many students are taking Chemistry 130 this year, and how does that compare to last year?
Response: If fewer, change times that Chem 130 is offered

How are they performing compared to last year?
Response: If worse, send out emails encouraging them to use help resources
Identification of data sources

(Who has the data, and how do we get it?)
Don't limit your thinking to an existing dataset.

Example: Select enrollment data from Student Data Warehouse. Combine it with Gradebook data to compare student performance this year vs. last year.
Design Target and ETL

(Where is the target? Is it a database or file or report? Do we need to design a database? If so, should it be relational or dimensional? Do we need to track change history or not?)

Example: Assume we already have dimensional data model student enrollment, we only need to add Grade data and performance data to it.
Iterative Development

- Test
- Deploy
- Develop

Detail ETL Design

Effort Estimate and Planning
### Why ETL Tools

**Does this look familiar?**

- Time consuming to learn
- Hard to change
- 400 lines of code to search
- Difficult to read
- Logging written manually
- Hard to roll back or set checkpoints
- Can’t load incrementally (just changes)
- Large volume = slow
- No data lineage and hard to do impact analysis

**How can we make this easier?**

**Hand Coded**

- Time consuming to learn
- Hard to change
- 400 lines of code to search
- Difficult to read
- Logging written manually
- Hard to roll back or set checkpoints
- Can’t load incrementally (just changes)
- Large volume = slow
- No data lineage and hard to do impact analysis

**ETL Tool**

- Easy to learn
- Easy analysis/change when data structures change
- Easy-to-read, visual format
- What you see is what you get: built-in logging and monitoring
- Easily set checkpoints and roll back or restart
- Can load incrementally (just changes)
- In-memory processing and data pipelining
- Data lineage and impact analysis

**What if you could do this instead?**

---

**Tool Evaluation Criteria**

When ETL tools are evaluated, several factors should be considered. Here are some key criteria:

- **Cost:**
  - Development Time
  - Implementation
  - Support
  - Maintenance

- **Easier:**
  - Data lineage tracking
  - Data quality
  - Integration between systems

- **Major Advantages:**
  - Improved data quality
  - Enhanced integration
  - Streamlined data lineage
  - Efficient data transformation
Why ETL Tools

Does this look familiar?

Hand Coded
- time consuming to learn
- costly to change
- 460 lines of code to search
- difficult to read
- logging written manually
- hard to roll back or set checkpoints
- can’t load incrementally (just changes)
- large volume = slow
- No data lineage and hard to do impact analysis

ETL Tool
- easy to learn
- easy analysis/change when data structures change
- easy-to-read, visual format
- what you see is what you get
- built-in logging and monitoring
- easily set checkpoints and roll back or restart
- can load incrementally (just changes)
- in-memory processing and data pipelining
- data lineage and impact analysis

What if you could do this instead?

Tool Evaluation Criteria

When ITS evaluated tools we developed a set of metrics based on the following factors:
- ETL capability
- Metadata capabilities (Metadata is simply information about your data, like table structure, field length and type, etc.)
- Information can import metadata from an existing database or BusinessObjects Universe or from EPMstudio EPM data modeling tool
- Development Environment
- Implementation support
<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
<th>Group By</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNT</td>
<td>ACCOUNT</td>
<td>Yes</td>
</tr>
<tr>
<td>EFFDT</td>
<td>EFFDT</td>
<td>Yes</td>
</tr>
<tr>
<td>DESCR</td>
<td>DESCR</td>
<td>Yes</td>
</tr>
<tr>
<td>ACCOUNT_TYPE</td>
<td>ACCOUNT_TYPE</td>
<td>Yes</td>
</tr>
<tr>
<td>EFF_STATUS</td>
<td>EFF_STATUS</td>
<td>Yes</td>
</tr>
<tr>
<td>DESCRLONG</td>
<td>DESCRLONG</td>
<td>Yes</td>
</tr>
<tr>
<td>ACCOUNT_TYPE_DESCR</td>
<td>ACCOUNT_TYPE_DESCR</td>
<td>Yes</td>
</tr>
<tr>
<td>STATISTICS_ACCOUNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TREE_NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TREE_NODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TREE_NODE_DESCR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TREE_LEVEL_NUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leaf_TREE_NODE_NUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node_TREE_NODE_NUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_TYPE</td>
<td><code>iif</code>(ACCOUNT_TYPE=' ' AND STATISTICS_ACCOUNT=' ')</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' ' OR ACCOUNT_TYPE=' ' AND STATISTICS_ACCOUNT=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_DESCR</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' ' OR ACCOUNT_TYPE=' ' AND STATISTICS_ACCOUNT=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SRC_USE_TYPE</td>
<td><code>decode</code>(trim(trim(<code>max</code>[<code>iif</code>(TREE_NAME='UM SRC USE ACCT',''), ACCOUNT_TYPE)), ' ')</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SRC_USE</td>
<td><code>max</code>[<code>iif</code>(TREE_NAME='UM SRC USE ACCT',''), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SRC_USE_DESCR</td>
<td><code>max</code>[<code>iif</code>(TREE_NAME='UM SRC USE ACCT',''), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_GASB</td>
<td><code>max</code>[<code>iif</code>(TREE_NAME='UM ACCT_GASB' AND ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_GASB_DESCR</td>
<td><code>max</code>[<code>iif</code>(TREE_NAME='UM ACCT_GASB' AND ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SPON_PROJ</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='E' AND TREE_NAME=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SP_VOL_RES</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_SP_VOL_RES_DESCr</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_ISR_BUDGET</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='E' AND TREE_NAME=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_MED_SCH</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_MED_SCH_DESCr</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE='R' OR ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_others</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE&lt;&gt;'R' AND ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
<tr>
<td>out_ACCOUNT_GRP_DESCR_others</td>
<td><code>max</code>[<code>iif</code>(ACCOUNT_TYPE&lt;&gt;'R' AND ACCOUNT_TYPE=' '), ACCOUNT_TYPE]</td>
<td></td>
</tr>
</tbody>
</table>
Tool Evaluation Criteria

When implementing tools, we developed a set of metrics based on the following factors:

- **ETL Capability**
- **Metadata capabilities** (Metadata is simply information about your data, like table structure, field length and type, etc.)
- **Information can import metadata from an existing database or BusinessObjects Universe, or from ER/Studio, ITS data modeling tools**
- **Development Environment**
- **Implementation support**

**Cost**
- Purchase license for ETL tool
- Consultant reports may be expensive
- Staff training

**Easier**
- Data Element Mapping
- Readability

**Major Advantages**
- Faster time to build and maintain data warehouses
- Better data quality
- Improved data accuracy
- Reduced time to load data
- Enhanced data quality
- Improved data availability
- Enhanced data security
Easier

Data Element Mapping

Readability
Major Advantages

- Less time to build and maintain (40-60% less)
- Lower learning curve (days vs. months)
- Faster and less load time (400 million rows in 2 hrs vs. does not finish)
- Use existing scheduling tool
- Type 2 dimensions
In-memory processing and pipelining

Data Lineage
- Purchase license for ETL tool
- Contract experts more expensive
- Staff training
Tool Evaluation Criteria

When ITS evaluated tools, we developed a set of metrics based on the following factors:

- ETL Capability
- Metadata capabilities (Metadata is simply information about your data, like table structure, field length and type, etc. Informatica can import metadata from an existing database or BusinessObjects Universe, or from ER/Studio, ITS' data modeling tool).
- Development Environment
- Implementation support
# How We Use ETL Tool

## Use It to Load Next Generation Data Warehouse

### Next Generation DW Architecture

- Source
- Data Staging
- Data Transformation
- Data Staging
- Data Integration
- Data Consumption
- Data Load

### NextGen DW ETL Flow

1. Integration Layer
2. Data Staging
3. Data Consumption

## Many Other Data Movements

- Integrate data from different sources into our warehouse. **Concur, myLINC**
- Send data to external databases eRAM
- Rewrite existing ETL for enhancement or to resolve performance concerns.

## Do vs. Don't

<table>
<thead>
<tr>
<th>Do</th>
<th>Don't</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Make your data and business analysts data</td>
<td>- Pile ETL tool or another feature</td>
</tr>
<tr>
<td>- Use group or team design sessions for ETL</td>
<td>- Use too many staging tables</td>
</tr>
<tr>
<td>- Create templates for common patterns</td>
<td>- Use multiple ETL services for one file</td>
</tr>
<tr>
<td>- Follow coding standards and best practices</td>
<td>- Use incorrect transformation when data is big</td>
</tr>
<tr>
<td>- Create reusable chunks of code</td>
<td>- Use incorrect name for one table or another</td>
</tr>
<tr>
<td>- Use your knowledge base</td>
<td>- Use incorrect transformation when data is big</td>
</tr>
<tr>
<td>- Share your knowledge base with others</td>
<td>- Use incorrect name for one table or another</td>
</tr>
<tr>
<td>- Develop ETL users from data analysts</td>
<td>- Use staging tables to build new entity ETL</td>
</tr>
<tr>
<td>- Use staging tables to build new entity ETL</td>
<td>- Don't use ETL in different approaches and scenarios</td>
</tr>
<tr>
<td>- Develop ETL users from data analysts</td>
<td>- Don't use any debugging tool</td>
</tr>
<tr>
<td>- Use staging tables to build new entity ETL</td>
<td>- Don't use a debugging tool</td>
</tr>
<tr>
<td>- Use staging tables to build new entity ETL</td>
<td>- Don't use any debugging tool</td>
</tr>
<tr>
<td>- Use staging tables to build new entity ETL</td>
<td>- Don't use a debugging tool</td>
</tr>
</tbody>
</table>
OLTP
Staging between OLTP and Integration layer
Quick extract
Few/no transformation

Staging
Incremental load only

End of day load

Keep full history of changes

MD5 & CDC capture changes

Use record begin date, end date, and current record indicator

Integration Layer
Staging between Integration layer and Star Schema
Use as needed
Use it to break down complex transformation

Staging
Use control table to determine last load date from IL

End of day load

Maintain referential integrities in data consumption layer

Dimensions and Fact tables in this layer
Some dimensions keep changes for "As-Was" reporting

Data Consumption Layer
<table>
<thead>
<tr>
<th>CL_TABLE_NAME</th>
<th>LAST_LOAD_START_DATE_TIME</th>
<th>LAST_LOAD_END_DATE_TIME</th>
<th>LOAD_START_TIME</th>
<th>LOAD_END_TIME</th>
<th>LOAD_DURATION</th>
<th>LAST_SUCCESSFUL</th>
<th>SUCCESS_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Academic Sub Plan</td>
<td>20-JUN-2011 00:27:01</td>
<td>20-JUN-2011 00:27:07</td>
<td>00:27:22</td>
<td>00:27:07</td>
<td>7</td>
<td>20-JUN-2011 00:27:07</td>
<td>Succeeded</td>
</tr>
<tr>
<td>Dimension Date</td>
<td>20-JUN-2011 00:27:01</td>
<td>20-JUN-2011 00:27:06</td>
<td>00:27:22</td>
<td>00:27:06</td>
<td>9</td>
<td>20-JUN-2011 00:27:06</td>
<td>Succeeded</td>
</tr>
<tr>
<td>Dimension Term</td>
<td>20-JUN-2011 00:27:01</td>
<td>20-JUN-2011 00:27:08</td>
<td>00:27:22</td>
<td>00:27:30</td>
<td>8</td>
<td>20-JUN-2011 00:27:30</td>
<td>Succeeded</td>
</tr>
<tr>
<td>Fact Application Tracking Management</td>
<td>20-JUN-2011 00:28:53</td>
<td>20-JUN-2011 00:30:52</td>
<td>00:28:53</td>
<td>00:29:57</td>
<td>70</td>
<td>20-JUN-2011 00:30:52</td>
<td>Succeeded</td>
</tr>
</tbody>
</table>
Many Other Data Movements

Integrate data from different sources into our warehouse. Concur, myLINC

Send data to external databases eRAM

Rewrite existing ETL for enhancement or to resolve performance concerns.
<table>
<thead>
<tr>
<th>Do</th>
<th>Don't</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Work with data architect and business analyst closely</td>
<td>- Think ETL tool is another language</td>
</tr>
<tr>
<td>- Do group or pair design your first few ETL</td>
<td>- Use too many staging tables</td>
</tr>
<tr>
<td>- Create templates for common patterns</td>
<td>- Use cached lookup when the lookup table is big</td>
</tr>
<tr>
<td>- Follow naming standards and best practices</td>
<td>- Use joiner transformation when tables are from the same database</td>
</tr>
<tr>
<td>- Create reusable objects like maplets</td>
<td>- Use custom code when it can be done using a transformation</td>
</tr>
<tr>
<td>- Use vendor's knowledge base</td>
<td>- Assume working is good enough</td>
</tr>
<tr>
<td>- Share templates/patterns with others</td>
<td></td>
</tr>
<tr>
<td>- Do tuning in ETL server first</td>
<td></td>
</tr>
<tr>
<td>- Use staging tables to breakdown complex ETL</td>
<td></td>
</tr>
<tr>
<td>- Evaluate trade-offs of using different approaches/transformations</td>
<td></td>
</tr>
<tr>
<td>- Use existing job scheduling tool</td>
<td></td>
</tr>
<tr>
<td>- Use version control</td>
<td></td>
</tr>
<tr>
<td>- Use the error handling strategy</td>
<td></td>
</tr>
<tr>
<td>- Use debugging tool</td>
<td></td>
</tr>
</tbody>
</table>
How You Can Use It

Thats how we use it.

How would you use it?

Think about it.

Explore your options.

Contact us.

Service under construction

Data moved by ITS from one system to another outside the data warehouse

Data pulled by ITS into data warehouse from other sources, managed by ITS

Data pushed into ITS data warehouse from other sources, managed by the sourcing unit

Interested in this service? Let us know if you would like to pilot this service.

bi.inform@umich.edu

Service under construction

Data moved by ITS from one system to another outside the data warehouse

Data pulled by ITS into data warehouse from other sources, managed by ITS

Data pushed into ITS data warehouse from other sources, managed by the sourcing unit

Interested in this service? Let us know if you would like to pilot this service.

bi.inform@umich.edu
That's how we use it.

How would you use it?
...and we like to share
Let us know if you are interested in being a pilot at bi.inform@umich.edu

Service under construction

Data moved by ITS from one system to another outside the data warehouse

Data pulled by ITS into data warehouse from other sources, managed by ITS

Data pushed into ITS data warehouse from other sources, managed by the sourcing unit

Or you can use Informatica installed on ITS server

Or you can leverage our contract to save on your own Informatica products

Interested in this service?
Let us know if you would like to pilot this service
bi.inform@umich.edu
I'm going to stop listening now...because I can't pay for it

Wait! UM already owns all of this:
- Informatica Power Center (ETL)
- Informatica Metadata Manager
- Informatica Metadata Exchange for BO and ER/Studio
- Informatica Data Analyzer (Reporting Services)
- Informatica Reference Table Manager

...and we like to share
Let us know if you are interested in being a pilot at bi.inform@umich.edu
Think about it.

Explore your options.

Contact us.
Conclusion

You may not call it ETL, but you are probably writing it, or need to start writing it.

It may seem "easier" to use PL/SQL or another language that is familiar...to you

...but is it less costly in the long run? For those who didn't write it?
...For you when you maintain it?
...can it perform at huge volumes?
...can it support business-critical demands like error handling, incremental loading, or restart/recovery from checkpoints?

For more information:

The Kimball Group
http://www.kimballgroup.com/

Higher Ed Data Warehouse Forum
http://hedw.org/

The Data Warehousing Institute
http://tdwi.org/