



ETL Proof of Concept Written Response

Course: Evaluating ETL Tools and Technologies, afternoon session
ETL Vendors in Action

This section contains responses provided by Talend to the scenario questions. It can be used as reference materials after the course. Additional information can be provided by contacting any of Talend's offices (full contact info: <http://www.talend.com/open-source-provider/contacts.php>, or simply via email: info@talend.com).

About Talend

Talend is the first provider of open source data integration software. After three years of intense research and development investment, and with solid financial backing from leading investment firms, Talend revolutionized the world of data integration when it released the first version of Talend Open Studio in 2006. Talend's solutions are used primarily for integration between operational systems, as well as for ETL (Extract, Transform, Load) for Business Intelligence and Data Warehousing, and for migration. Unlike proprietary, closed solutions, which can only be afforded by the largest and wealthiest organizations, Talend makes data integration solutions available to organizations of all sizes, and for all integration needs.

Table of Contents

Proof of Concept Overview	3
Scenario Overview	3
Demo Scenarios / Topics	5
1. Product Architecture Overview and ETL Perspective.....	1
2. Extract Scenario 1: Customer Dimension Incremental Extract.....	3
3. Extract Scenario 2: Shipments Fact Table Extract	4
4. Extraction Scenario 3: Open Case	4
5. Extraction Scenario 4: Time Dimension	5
6. Maintenance Features	5
7. Operations and Deployment.....	6
8. Pricing	8
9. Performance Features	11

Proof of Concept Overview

The scenarios for the proof of concept are all based on a wholesale business that supplies specialty products to retailers. The scenarios are based on the items that one might consider important when evaluating an ETL solution for a single data warehouse.

The examples are all built around a wholesale shipments schema, with a set of source tables loaded with data and a set of target tables to be populated by the tools. The extract rules for the schema are simple, but should be enough to demonstrate basic and some advanced capabilities in the products.

The afternoon will be a mix of discussion and demo, with the emphasis on showing how the products are used to accomplish specific tasks. While the focus is on extraction, some of the scenarios or presentation topics involve showing other features like metadata management, data profiling or monitoring job execution.

Because there's no way to show the entire set of ETL for three vendors in the time allotted, we'll be using different elements to show different features. For the scenarios listed we expect to see the features used to accomplish the task live. It isn't expected that we can see the entire extract constructed for each scenario in the time given. However, a complete set of extracts is required in order to show how dependencies, scheduling and monitoring work.

Demo time is limited so there are topics/scenarios labeled "time permitted" which we may not be able to show. They are included in case we have extra time at the end of the class.

Scenario Overview

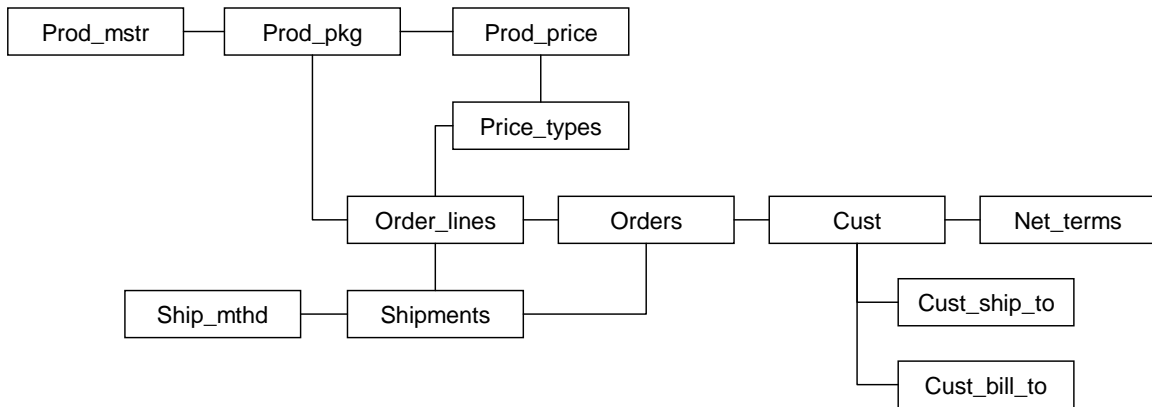
In a proof of concept you provide to vendors all the source and target table definitions, extract rules and source data. Since this is meant to reflect the real ETL you'll be doing, it's a good idea to select both simple extracts and complex extracts or extracts that have problem data. When you provide this information, it should be formally documented so the vendor understands in detail what they are supposed to show.

Part of the reason for selecting source data with quality problems is that this will show how a developer is expected to work within the tool. If all the extracts are based on ideal tables and data, as with standard vendor demos, then you won't see what a developer really has to face when dealing with data exceptions.

As a rule, you should have imperfect data, tables with relationship problems like different types on join or lookup columns, and you should always require the use of relational database in the proof of concept.

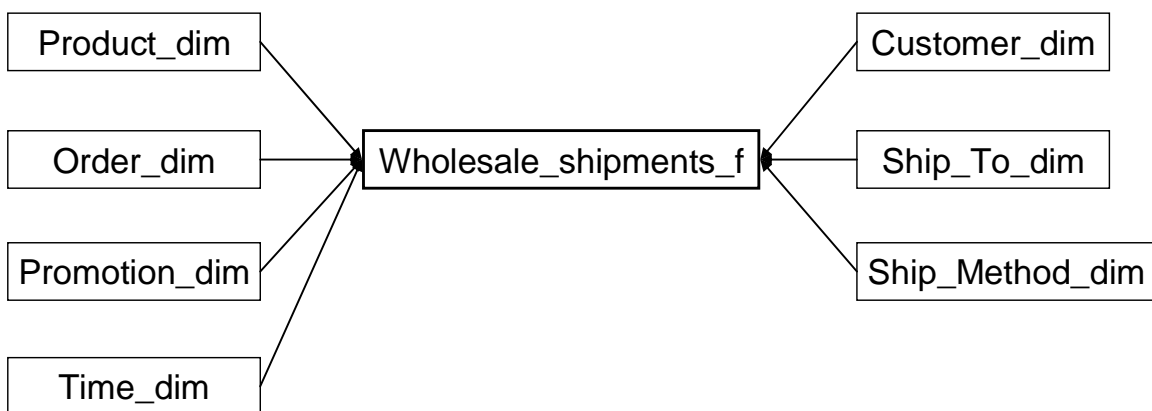
Using a database is important because it will show you how the tool interacts with a database. Many vendor demos you see are based on text files as input and output, which can avoid some of the difficulties when dealing with SQL since all the work is done directly in the ETL engine.

For our scenarios, we will be using the following source and target schemas. The source schema consists of 12 tables with some of the common design problems found in source databases.



Among the problems are a mix of second and third normal form, multi-part keys, invalid foreign key constraints, and multiple join paths to the same data elements. In addition to the above tables there are two change-data-capture tables for the shipment and customer data. These are used for incremental extract examples.

The target schema has seven dimension tables and one fact table. The fact table contains all shipments of order lines by day to customer addresses in the ship_to dimension.



There are several things in this target schema that complicate the extracts. The time dimension is based on a fiscal calendar using 4 and 5 week periods rather than a simple date-based dimension. There is no source for this data, so it must be constructed by the ETL job. A non-data-driven extract is a challenge for some

ETL products. The ship_method dimension has unique rows based on a multi-part key which can cause trouble for some ETL tools' lookup functions. The specific details about extract rules and data are available at the end of this document.

Demo Scenarios / Topics

The following section describes each of the presentation topics or scenarios that will be reviewed during the course. For each scenario, there is a list of directions or questions to be answered and a short view of the schema and data (if applicable).

Included with the descriptions are sample criteria you might use during an evaluation so that you can score the vendors during the class. After this section there are responses from each of the vendors to all of the scenario questions so you have something to refer back to.

1. Product Architecture Overview and ETL Perspective

The product name or components used for this demo, whether all components are included in the base product and / or how it's packaged.

The product used for this demo is **Talend Integration Suite**, Talend's enterprise data integration solution. Talend Integration Suite is an extension of Talend Open Studio, Talend's product offered at no cost under a GPL v2 license. Talend Integration Suite extends Talend Open Studio with teamwork capabilities and enterprise deployment features. It is offered by Talend as a subscription that includes all the features and technical support. All components used in the demo are included in Talend Integration Suite.

**What is the core ETL product offering?
What optional components are provided?
How are components / editions bundled?**

Talend's base ETL product offering is **Talend Open Studio**. It includes the business modeling capabilities (for non-technical business users), the ETL job design capabilities, the local metadata repository, the run-time capabilities, all components and connectors. Talend Open Studio is a full-featured data integration environment, technically and functionally on par (if not superior) with proprietary offerings.

Talend Integration Suite extends Talend Open Studio with:

- Teamwork capabilities
- Enterprise-scale deployment options
- Integration process monitoring features

In addition, Talend also offers **Talend On Demand**, which is a data integration offering provided in a Software-as-a Service mode, enabling data integration teams to share their repository.

Both Talend On Demand and Talend Integration Suite are commercial offerings; their price is however only a fraction of the cost of proprietary ETL solutions.

What requirements are there for server installs, client installs, etc.?

Talend Integration Suite runs on Windows (all versions including Vista), Linux, Mac, all major Unix flavors, etc. (it is actually an Eclipse plug-in so the burden of portability lies with the Eclipse folks, who are doing an excellent job at it!). At run-time, Talend Integration Suite generates Perl, Java and SQL programs – the most portable languages ever.

What external dependencies are there? For example, a license for a database to be used as a metadata repository.

None, open source databases can be used for the metadata repository.

What do you consider your sweet spot in applications or in industries?

Talend Integration Suite caters to all data integration needs, for all types and sizes of organizations. The solution is widely adopted by:

- Former users of proprietary solutions who are becoming increasingly wary of the high maintenance costs of their products and the lack of responsiveness of the vendor to their needs.
- Teams used to doing “manual coding” of their integration jobs, with all the consequences on the quality, error processing and maintainability of these jobs.

What are your product’s strongest points?

Why would a customer choose your product over others?

Talend Integration Suite’s strongest points fall into two categories:

- From the **financial** perspective:
 - **Licensing:** There exists a fully capable GPL v2, no cost version (Talend Open Studio). Users may try, validate the features and test at their own pace without vendor pressure.
 - **Price:** The low-cost license removes a high barrier for adoption. Based on the number of ETL designers, the pricing model removes all licensing limitations based on the technical architecture (deploy on as many servers as you want), data volumes (transform as much data as needed), number of connectors available (all connectors available at no additional fee and it’s easy to create customized connectors) or number of data sources or targets (reach out as broadly as required). This allows customers to drive the most value out of the platform.
 - **Subscription:** The subscription approach allows customers to buy just the right amount of licenses they need and to adjust it as their needs evolve over time – no shelfware, optimum TCO.
- From the **technical** perspective:
 - Talend Integration Suite’s **business modeler** lets business users get involved in the data integration project with a non-technical, top-down approach.
 - The graphical, drag-and-drop job designer offers excellent ease of use and **increases productivity**.

- The code-generation approach with grid execution offers **unbeatable performance** and runs on commodity hardware – no need to deploy a super expensive 24-CPU server to run the ETL engine!
- Talend Integration Suite is the only data integration environment to support **both the ETL and the ELT** approaches.
- Talend Integration Suite is highly versatile and expandable – it features the **broadest connectivity** of all ETL solutions, and it's very easy to create customized connectors.

2. Extract Scenario 1: Customer Dimension Incremental Extract

This scenario demonstrates the extract logic and facilities for a slowly changing dimension where history is preserved in the rows using start and end dates.

How do you perform a simple lookup from another table?

Talend Integration Suite's tMap component supports lookups, simple and multiple (ie. lookups on lookups). This is entirely graphical, and can handle as many inputs and outputs than required.

Are there any features that automate or simplify the maintenance of slowly changing dimensions, or must you do this work yourself?

There is a built-in Slowly Changing Dimension component that covers types 1, 2 and 3. This component is available out of the box for all major databases.

How do you deal with both inserts and updates against a target table without requiring separate logic?

Database output components support the following options:

- insert
- insert or update
- update or insert
- delete

It is not necessary to implement a specific logic to handle inserts and updates.

Do you have any problems with locking or staging when retrieving and updating rows in the target table as part of the extract?

No such problem was reported by Talend Integration Suite users.

How do you preview source data from within the design interface?

The graphical design tool built into Talend Integration Suite (SQLBuilder) includes a preview mode that allows previewing data during job design.

Can you look at both the input and output data from within the developer interface to see how updates were applied?

In Trace Mode, it is possible to view data as it flows through the different steps of the job and to follow transformations.

In the case where you do not preserve history, what steps do you have to take for a destructive load?

Perform a Clear or a Truncate before the load.

Is there any support for managing hierarchies in dimensions?

Not at this point. This is planned for a future version.

3. Extract Scenario 2: Shipments Fact Table Extract

The purpose of this scenario is to show a more complex extract involving conditional logic, calculations, many source tables, multiple lookups, and available facilities for dealing with surrogate key matching and validation for dimensional fact tables.

Answers provided during the demonstration.

4. Extraction Scenario 3: Open Case

This is an open case. The vendors have been asked to demonstrate something that is unique or compelling about their products. The specific features they show aren't always known in advance.

Answers provided during the demonstration.

5. Extraction Scenario 4: Time Dimension

This scenario involves building a derived table where there is no data source. The job must construct a time dimension based on a fiscal (4-4-5) calendar with a number of derived date attributes. The vendors have been asked to explain and demonstrate how one would construct a derived table like this where there is no data source and the data must be created by the ETL program.

This sort of extract also shows how vendors deal with looping constructs and conditional logic, both of which are needed to work out problems like leap years and fiscal years crossing into new calendar years.

What facilities are there to generate data?

Talend Integration Suite include a tRowGenerator component. It generates data that match a description. It generates data that is both technically consistent (type, length, unicity, etc.) and business-like (email, name, address, etc.) It is also possible to add custom code generation functions, they will be added automatically to the tRowGenerator.

Ease of addressing conditional logic and loops.

Conditional logic is handled fully graphically through the tMap component (multiple IF...THEN levels with and/or Boolean capabilities).
Loops are also supported (finite and infinite loops).

Date management and attribute functionality.

Many date functions are available (Talend uses all Perl and Java libraries available), for example add/substract a number of days from a date, convert dates, compute number of days between two dates, etc.

Does the product recognize time as a specific type of dimension?

Not yet (planned for 2008).

6. Maintenance Features

Post-deployment maintenance is an important aspect of the ETL process. There is no specific scenario. Instead, the vendors have been asked to describe and

demonstrate features available to developers that address the types of questions outlined below.

Assume there is a change to the data type of the order_nbr column in the Orders dimension. What is the process of tracing the impact of this change and how would a developer change the affected extracts?

Impact analysis will be provided in a future version.

What source control or version control features are available?

All objects are fully versioned. It is possible to go back to any version of any object.

Check In/Check Out (with locks management) is also supported.

What facilities are available for documenting or annotating extracts?

Notes can be added to any job. Each component can also be documented. Talend Integration Suite includes technical documentation automatic generation features.

Advanced business modeling features are also available.

How does a developer compare and find differences between an extract that has been deployed in production and one in the repository / development environment?

This is done through versioning.

How does a developer address production deployment and rollback of jobs?

During deployment, the administrator chooses the job version he wants to deploy. It is always possible to select an earlier version.

7. Operations and Deployment

To show features available for scheduling and monitoring, we are using the complete set of extracts for the dimension and fact tables. The vendors have been asked to show how dependencies between the various extracts are configured, how a schedule for execution is created, and how the extract jobs are

monitored. The goal is to show product features related to the types of questions outlined below.

What is the executable unit of work in the product?

The executable unit of work is a job, which is made of several components linked to one another. Several links can be used:

- flow
- iterate (from a file list or a loop component for example)
- before (chaining sub-jobs)
- conditional (“if ok”, “if error”, “user defined if”)

A job can call another job and pass parameters.

How do you make a change to the dependencies?

With the metadata manager.

How do schedule-based initiation and event-based initiation of jobs work?

This is done through Talend’s built-in scheduler. Talend Scheduler is:

- Time based (support CRON syntax and more)
- Event based (supports email call, Web Service call, File (dis)appear, MOM call, RDBMS call (trigger)...))

How can execution be monitored by a developer within the development environment, and by an administrator outside the development environment?

Talend provides 2 ways to do this:

- Through the Activity Monitoring Console (rich client, eclipse based)
- Through the Business Dashboard (lightweight application)

Explain the mechanisms available for monitoring execution and sending alerts if there are problems.

Two types of information can be trapped: log information (status, errors) and statistical information (number of records processed, execution time for a job, for a component, etc.)

This information can be sent to files or in a RDBMS. Talend Activity Monitoring Console reads and presents graphically the information (charts, etc.).

8. Pricing

Basic criteria:

Is there a per seat cost for developers?	Yes
Is there a per seat cost for administrators?	No
Is there a price per server by CPU? Per core?	No
Is the price different for different server operating systems?	No
Are there per source or per target server / instance charges?	No
Are there additional source / target connector charges?	No
Is there a charge for development or test environments? If so, is it the same cost?	No
How is maintenance charged? What is the range if it is a percent of some set of costs?	Maintenance is charged per user, at a fixed price . The cost of maintenance does not depend on the complexity of the project, the number of CPUs, the number of sources...
How many different editions or bundles are offered?	<ul style="list-style-type: none"> • Talend Open Studio • Talend On Demand • Talend Integration Suite <ul style="list-style-type: none"> ○ Team Edition ○ Professional Edition ○ Enterprise Edition (see products descriptions below)
Are there additional charges for team-based development, e.g. source control, project-level security, role-based access?	No

Products description:

- **Talend Open Studio** is Talend's full featured ETL environment that has been presented during the demo. It is licensed under a GPLv2 (no-cost) license. It changes the economics of data integration by providing a more scalable, less-

expensive open source alternative to traditional, proprietary data integration suites. The open source approach chosen by Talend extensively leverages the community, expediting product enhancements that directly address user feedback and needs.

- **Talend On Demand** is the first open source data integration solution on the market delivered as a service. It provides a centralized and shared repository, facilitating project-team collaboration and object and code reuse, as well as promoting development best practices – without requiring sensitive enterprise data to be moved outside the corporate firewall. Project data is hosted separately, facilitating optimal performance while ensuring privacy.
- **Talend Integration Suite** is an enterprise-grade, high-performance data integration offering available at a fraction of the cost of proprietary solutions. It enables customers to remain in control through both the open source and subscription models, without being tied in to long-term licensing fees.

Scenario 1: Department / project level ETL

- A single data warehouse, with one ETL project
- 3 ETL developers, 1 administrator / operations role
- 2 different (non-legacy) database source types, and some file-based extracts
- 1 target database server, 4 CPUs
- One production environment
- One test and development environment
- Small volumes of raw data moved through standard star-schema style batch extract, with the total target warehouse size of 180 GB of data (60GB of data loaded per year).
- Standard next-business-day support

Note: please specify the number of servers and CPUs used to support this configuration

Talend Integration Suite Team Edition

Subscription: **\$8,500/year**, including support

Required configuration

Designer: 1 GB RAM, 500 MB HD; OS: Win32 or Linux/Unix
 Admin server: 1 GB RAM, 1 GB HD; OS: Win32 or Linux/Unix
 Execution Server: not applicable (code generation)

Scenario 2: Enterprise ETL

- Multiple data warehouses/marts with several different ETL projects
- 10 ETL developers, 3 administrator / operations roles
- 3 different (non-legacy) database source types, file-based extracts, one SAP source system, requirement to consume XML formatted data
- 3 target database servers for warehouses / marts - 12 4, 4 CPUs respectively
- 3 production environments for the three projects (but infrastructure, repositories, etc. is centrally managed)
- 3 half-scale development environments
- 3 half-scale test environments
- Moderate volumes of raw data moved through standard star-schema style batch extracts for two projects, with the total target sizes of each at 500 GB of data (~160 GB of data loaded per year).
- One larger project environment with 2 TB of data (~650GB of data loaded per year), through more complex rules in batch mode, plus small amounts of data streaming in through your choice of either message queues / ESB / event publishing and then processed through your choice of either on-demand or in mini-batches. Specify the preferred option for streaming data consumption for the purpose of this pricing exercise. Assume an 8 CPU database sever is this is important.
- Note: this environment requires team-based development support and project-level security and roles.
- Enterprise level (same-day response) support including off-hours and weekends

Note: please specify the number of servers and CPUs used to support this configuration

Talend Integration Suite Enterprise Edition

Subscription: **\$73,000/year**, including support

Required configuration

Designer: 1 GB RAM, 500 MB HD; OS: Win32 or Linux/Unix
Admin server: 1 GB RAM, 1 GB HD; OS: Win32 or Linux/Unix
Execution Server: not applicable (code generation)

9. Performance Features

There is no easy way in a class setting to demonstrate performance. Instead, the vendors have been asked to describe features in the product that specifically address performance needs and answer common performance questions like the following.

What features are available to deal with large volumes of source data?

- ETL / ELT combination (can be used in the same job)
- Native connectors to large data warehousing appliances (eg. Teradata)
- Multi CPU optimization
- Multi server deployment with load balancing

How does a developer set up or create an extract that can execute in parallel? Does it require special transforms or is it the same ETL logic regardless of parallel degree? Does it require changes to the database like table partitioning?

What features are available for caching of source data or lookup values, and what are the limitations?

Data are cached automatically based on available memory. This feature is used for example in all lookups and in Slowly Changing Dimension components.

How can the product be configured to run extracts on more than one physical server, if this capability is available? What must the developer do differently to take advantage of these features? Are there grid / concurrent processing capabilities? If so, how do they work?

Job Conductor is a module that can distribute jobs across the runtime servers. This can be done statically: a job is assigned to a physical server, or dynamically: one creates virtual servers and assigns jobs to them. When launching the job, the application deploys and executes it on the most available physical server (based on criteria defined by the administrator: available disk, available CPU, network speed, etc.)

In both these modes it is possible to assign triggers (time based or event based) to start jobs.

Can individual extracts be parallelized to run across servers?

Yes, the same way than explained above.