

ETL Proof of Concept Written Response: Microsoft

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

Table of Contents

Proof of Concept Overview	2
Scenario Overview	2
Demo Scenarios / Topics	4
1. View of ETL / Data Integration and Product Architecture Overview	4
2. Extract Scenario 1: Customer Dimension Incremental Extract	6
3. Extract Scenario 2: Shipments Fact Table Extract	8
4. Extraction Scenario 3: Open Case	9
5. Extraction Scenario 4: Time Dimension.....	9
6. Maintenance Features	10
7. Operations and Deployment	11
8. Pricing	13
9. Performance Features.....	17

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.

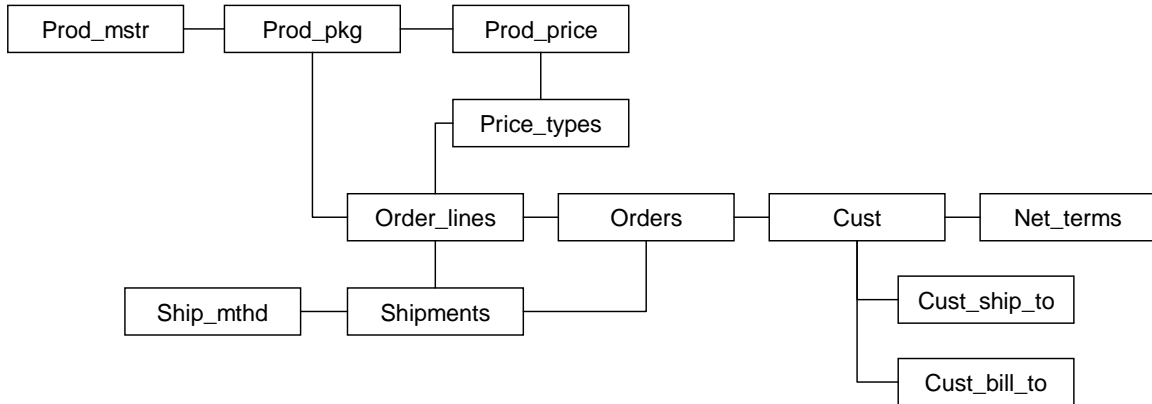
Part of the reason for selecting source data with quality problems is that this will show how a developer is expected to work within the tool. If all the extracts are based on ideal tables and data, as with standard vendor demos, then you won't see what a developer really has to face when dealing with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

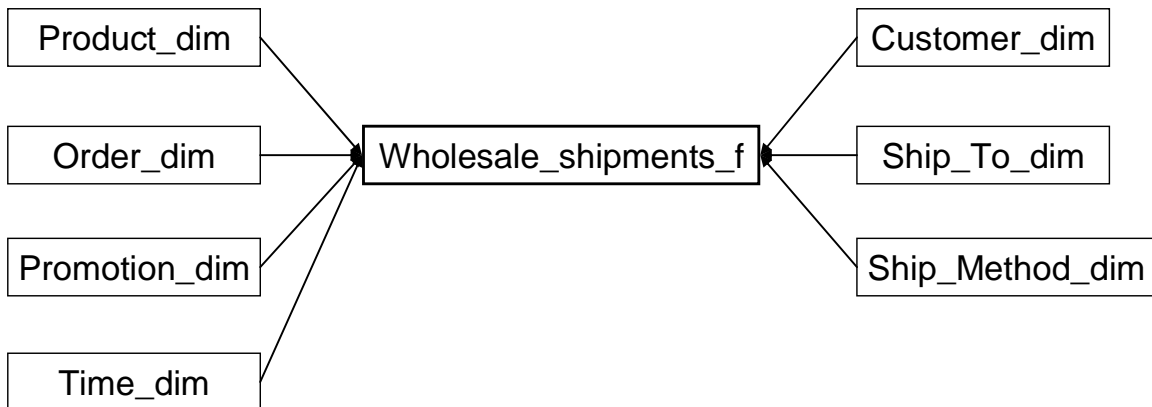
can avoid some of the difficulties when dealing with SQL since all the work is done directly in the ETL engine.

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

This is a short (4 minutes or less) presentation about the company, its products and its view of the ETL and data integration market. You should expect to hear answers to the following types of questions:

- The product name or components used for this demo, whether all components are included in the base product and / or how it's packaged.
Microsoft SQL Server 2008. Most components are available in the Standard Edition of the product except for some advanced functionality in the Enterprise Edition such as Fuzzy Lookup and grouping. SSIS Import and Export Wizard package save function is disabled in Web and Workgroup Edition.
- What is the core ETL product offering?
SQL Server Integration Services
- What optional components are provided?
No optional components
- How are components / editions bundled?

Edition	Components
Web Edition	<ol style="list-style-type: none"> 1. SSIS Import/Export Wizard (with save wizard package option disable) 2. SSIS runtime 3. Basic Source and Destination Adapters
Workgroup Edition	<ol style="list-style-type: none"> 1. SSIS Import/Export Wizard (with save wizard package option disable) 2. SSIS runtime 3. Basic Source and Destination Adapters
Standard Edition	<ol style="list-style-type: none"> 1. SSIS Import/Export Wizard 2. SSIS runtime 3. SSIS Package Designer and VSTA

	4. Design Basic Transforms 5. Basic Data Profiling Tools 6. Basic Resource and Destination Adapter
Enterprise Edition	Everything in Standard Edition, plus 1. Data Mining Model Training Destination Adapter. 2. Dimension Processing Destination Adapter. 3. Partition Processing Destination Adapter. 4. Persistent (high performance) Lookups (needs further break up of sub- features) 5. Data Mining Query Transformation. 6. Fuzzy grouping & Lookup transformation. 7. Term extraction & Lookup transformation.

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

Feature	Minimum Processor	Recommended Processor	Minimum Memory (MB)	Recommended Memory (MB)
Web (x86,x64,ia64)	1 Ghz (x86) or 1.4 Ghz (x64)	> 2 GHz Rec X86/X64, > 1 GHz for IA64	512	2048
Workgroup	1 Ghz (x86) or 1.4 Ghz (x64)	> 2 GHz Rec X86/X64, > 1 GHz for IA64	512	2048
Standard	1 Ghz (x86) or 1.4 Ghz (x64)	> 2 GHz Rec X86/X64, > 1 GHz for IA64	512	2048
Enterprise	1 Ghz (x86) or 1.4 Ghz (x64)	> 2 GHz Rec X86/X64, > 1 GHz for IA64	512	2048

- What external dependencies are there? For example, a license for a database to be used as a metadata repository.
No
- What do you consider your sweet spot in applications or in industries?
Integration Services (like other SQL Server BI servers) is really a platform application – SQL Server provides a Business Intelligence platform, with one stop development for Analysis, Reporting and Integration. It provides an integrated development environment with BIDS for building your ETL

packages, reports and cubes. We also have an integrated management environment in SSMS to manage all your BI solutions. SQL Server BI can provide scalable solution for largest customers all the way to Small and medium business. Our sweet spot as a pure-play product is that we scale linearly for the largest solutions while providing an easy to use interface to manage SMB solutions.

- What are your product's strongest points?
Ease-of-use, extensibility, a huge user and partner community, an excellent performance/price ratio, and the full range of features that come in the SQL Server box.
- Why would a customer choose your product over others?
Typically ease-of-use, the power/price ratio and the ability to license the entire SQL Server stack at no additional cost.

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.

Scenario

Change rows are stored in a change data capture table. This is a table that mirrors the source table, but has additional columns that indicate whether a row is an insert, update or delete, and a timestamp for the activity. This is what you might see as the result of change replication using database replication facilities. The expectation is that the change table provides the activity, while the source table shows the current state of the data.

To properly test the extract, there are four rows for three different conditions. The test cases are as follows:

- Insert of a new customer
- Update of an existing customer, changing address line 2.
- Same day insert and later update of a row (within the same batch).

The goal of this scenario is to show how one would address the issues associated with slowly-changing dimension tables where a changed-data capture table is used as input. The intent is to also show some of the features available for looking at data values before and after update from within the developer's interface. The demonstration should answer the types of questions outlined below.

- How do you perform a simple lookup from another table?
 - **First create a Package with Source and Destination Connection Managers, and an OLEDB Connection Manager for the database containing the lookup table, if not in the source or destination.**
 - **Add a Data Flow Task.**
 - **Add and configure a Source Adapter to perform the source extract as in the previous example.**

- **Add a Lookup component and connect the output of the Source Adapter to it.**
- **Edit the Lookup component, selecting the appropriate Connection Manager, and the reference table or view – in this case net_terms. Alternatively, and in many cases more efficiently, a SQL query can be visually built and tested in the Lookup component editor, to extract only a subset of the full reference table.**
- **Having selected the reference table, the editor will automatically map source columns to lookup columns based on name and type.**
- **You can manually define more columns for the lookup join, or remove unneeded mappings.**
- **Finally, select the columns from which values should be returned by the lookup operation.**
- Are there any features that automate or simplify the maintenance of slowly changing dimensions, or must you do this work yourself?
- **The Slowly Changing Dimension component includes a wizard that helps even novice users to build Type I and II slowly changing dimensions, with additional support for Inferred Members (hybrid I+II) and what we call Type 0 dimensions. Type 0 dimensions require either an error to be raised, or special custom processing, if a change is detected. The wizard creates a fully customizable data flow for handling the dimension load.**
- How do you deal with both inserts and updates against a target table without requiring separate logic?
The SCD wizard creates multiple outputs – including separate outputs and destination adapters for updates and inserts as needed.
- Do you have any problems with locking or staging when retrieving and updating rows in the target table as part of the extract?
No. All the inserts and updates within the data flow enroll in a single transaction. If it is wished to isolate inserts from updates, updates can be staged rather than updated in parallel and applied in a subsequent, serialized, step.
- How do you preview source data from within the design interface?
The source adapters include a preview option – it typically shows the first 200 rows of data in the source table.
Using a more advanced feature – the Data Source View – it is possible to preview larger samples, or to profile the data in the design interface. Using a Data Source View involves the use of some additional objects and is not covered in this scenario here.
- Can you look at both the input and output data from within the developer interface to see how updates were applied?
Destination data can also be viewed in the design interface, once the updates have been committed.
- In the case where you do not preserve history, what steps do you have to take for a destructive load?
The Slowly Changing Dimension wizard allows you to configure Type I dimensions which keep no history during updates. Complete destruction of

the dimension table is easily achieved by using the Execute SQL task, to issue a TRUNCATE on the table.

- Is there any support for managing hierarchies in dimensions?
SQL Server Analysis Services (included with SQL Server) includes a full complement of tools to manage hierarchies in dimensions.

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.

Scenario

The goal of this scenario is to show how one would address the more complex issues associated with building a fact table. Added complications in the data include common problems faced by developers. This includes dealing with missing data (forcing outer joins or conditional lookups), varchar and char data type conversions and comparisons, and missing values. The demonstration should provide answers to the following types of questions.

- How do you do a lookup based on a multi-column input key? (ship_method_key uses both ship_cd and carrier)
This is done in exactly the same way as for a single column join, simply by dragging additional source-to-reference mappings.
- How would you accomplish the equivalent of an outer join between two tables? (the shipments table has some bad product values that won't join to the product tables, there is a non-existent customer number, and the ship_to address is missing for a shipment)
 1. **There are two available for this. One uses the MergeJoin component with an Outer Join selected.**
 2. **The easiest and most common technique is to configure the Error Output of a Lookup component to Redirect Rows on Failure. This is done by selecting the appropriate setting on the Error Handling tab of the Lookup component editor. The Lookup regards the case where a join is not found as an Error. You can select to have errors fail the entire load, or to be ignored (with lookup values passed downstream as NULLs) or for error rows to be redirected to a special output where they can subsequently handled by further components and destinations.**
- Assume the developer forgot to include a table in the metadata so it isn't present but is needed at design time. How does the developer import the metadata so the table can be used in this extract?
 1. **All SSIS metadata is collected at runtime. With no intermediate repository all metadata is "live" so this issue does not typically arise.**
 2. **In cases where Data Source Views have been defined as metadata objects – a more advanced technique than we cover here – it is possible to add more tables at design time if needed, by selecting Add Tables and selecting from a list refreshed at design-time.**

- How do you do calculations involving values from multiple tables, or aggregation? (unit calculations are an example, and missing rows in one table can cause difficulties with these calculations)
 1. **In the first case, the tables must be joined using a source extract query, a MergeJoin, or a Lookup. All these operations bring columns from multiple tables into a single path on the data flow, where a Derived Column component can perform calculations on them together.**
 2. **In the second case, an Aggregation component can perform the summing operation.**
- What features does the tool have to automate common tasks like keying fact tables?
There are no facilities for automatically keying fact tables.

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.

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?
Most ETL developers tend to use our Script Component to write C# or Visual Basic code that generates data. SSIS does not include in-box data generation components, although example data generation components can be found online at msdn.microsoft.com.
- Ease of addressing conditional logic and loops.
Looping is available and easily utilized in a number of contexts within SSIS. In the SSIS control flow, you can graphically create loops of tasks that should be run over a set of data, data sources, or other things. For example, the control flow lets you create a graphical "Foreach loop" that enumerates things like databases, files, events, or rows in a table, and performs arbitrary tasks for each item. The "For loop" in the SSIS Control Flow lets you execute tasks and is controlled by a configurable termination expression.

When generating or processing data, or implementing custom transformations, customers can use any of the syntax provided in the C# or Visual Basic languages in their Script Components for any desired purpose.

For example, a script component could be used as a source in a data flow, be implemented in C#, and contain a standard “for” loop to control the generation of data.

- Date management and attribute functionality.
For generating data involving dates, this example and many real-world examples use date and calendar functionality from the .NET framework. .NET provides a full complement of classes to make date processing and computation easy and accurate.
- Does the product recognize time as a specific type of dimension?
The product’s type system includes specific support for times, but the “Slowly Changing Dimension” wizard treats time dimensions no differently than other types. The creation and population of time dimensions using many different calendar formats is possible with the Calendar component, an example SSIS Data Flow component provided at msdn.microsoft.com. For customers using SQL Server Analysis Services as their OLAP engine, generation of time dimensions with many calendar options can be easily automated using Analysis Services’ *Dimension Wizard*.

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?
 1. **There are two techniques that can be used:**
 2. **Firstly, the Business Intelligence Developer Environment features validation and build steps for Integration Services, just as for code development.**
 3. **On opening an extract Package affected by a change to a column, the Package will be validated against the source metadata and the user warned in the design environment of any breaking changes. Most breaking changes can be simply “fixed-up” by opening the editors for affected components and accepting default the metadata synchronization behavior. Other fixes can of course be applied in the designer at this time too.**
 4. **Of course the question remains – how to find all broken Packages? The Build option in Visual Studio will validate and build all Packages in a Solution – a Solution can consist of multiple Projects, each containing many Packages, so an enterprise Solution (just as for a complex code project) can perform accurate impact analysis and validation on many Packages in a single process.**
 5. **A second option is to download the Metadata Reporting Samples Pack from the Microsoft Developer Network. This pack includes lineage and impact analysis viewers and reports that can visually trace**

lineage from the relational system to affected SSIS packages, and Analysis Services cubes. The Metadata pack is shared source and fully customizable.

- What source control or version control features are available?
Any source code technology that can be used with Visual Studio – such as Microsoft Source Safe, Microsoft Visual Team System – including non Microsoft tools such as Team Coherence.
- What facilities are available for documenting or annotating extracts?
Annotations can be captured on the design surface, and all objects can have free text descriptive fields added. A shared source, customizable, documenter is available for download from the Microsoft Developer Network –it can document all objects together with their descriptions in HTML.
- How does a developer compare and find differences between an extract that has been deployed in production and one in the repository / development environment?
 1. **If a source control system is in use, then differences can be viewed in the source controls differencing viewer. As all Packages are XML format, other XML difference viewers can also be shown.**
 2. **Many users like to document their Packages using the HTML documenter, selecting to document only a subset of all the possible fields and attributes of the objects, and then difference the simplified documents.**
- How does a developer address production deployment and rollback of jobs?
SQL Server Management Studio can be used to move new versions of packages from test servers to production interactively or the same operations may be scripted with the command-line DTUTIL tool. Agent jobs and execution schedules are also managed using management studio, and the movement of jobs from test to production can be performed interactively or in an automated fashion using SQL or the SQL Management Objects API.

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 Package
- Demonstrate how execution dependencies between jobs are set up (e.g. fact table doesn't load until dimension tables have loaded).
 1. **Packages are fully composable containers, so a master package can execute the dimension packages using ExecutePackage Tasks and subsequently execute the fact load package in the same way.**

2. A less popular and less flexible alternative is to schedule the package as ordered steps in the SQL Server Agent scheduling tool.
- How do you make a change to the dependencies?
Because Packages are composable, the same design environment that can be used to build Packages, can also be used to build master control and child Packages. Dependencies between such Packages are therefore implemented in the same way as dependencies between individual tasks within a Package.
 - How do schedule-based initiation and event-based initiation of jobs work?
 1. **Scheduled initiation is typically performed using SQL Server Agent. Agent Jobs can be set up using the New Job wizard in SQL Server Management Studio and selecting a schedule for the job.**
 2. **Event based initiation can also be performed using SQL Server Agent in the same way, but instead of selecting a schedule, select a Windows Management Instrumentation event.**
 - How can execution be monitored by a developer within the development environment, and by an administrator outside the development environment?
 1. **Execution can be monitored by the developer in the design environment in four ways:**
 - a. **Visual progress reporting using color-coded progress indicators, and active row counts, enable a full visual debugging experience for the developer.**
 - b. **The developer can set up logging for any package to raise logging events. These events can be viewed in the dockable Log Events window in the designer.**
 - c. **The developer can view the Progress pane in the designer. This pane shows all events and messages raised by the executing package.**
 - d. **While any or all of the above are in use, the dockable Output and Error windows of Visual Studio are continuously updated with messages from the executing package.**
 2. **Execution can be monitored by the administrator without the design environment in 5 ways:**
 - a. **The Integration Services Object Explorer in SQL Server Management Studio shows all currently executing Packages.**
 - b. **If logging has been set up for a Package, it can post logging events at variable granularities of detail to multiple “providers” such as text or XML files, SQL Server tables, SQL Server Profiler and the Windows NT Event Log. Custom log providers for third party monitoring tools are easily written by developers moderately familiar with .Net programming.**
 - c. **Even if a Package does not have explicit logging, Package start and stop events are always posted to the NT Event log.**
 - d. **The Integration Services engine includes a number of useful performance counters for use in real-time with Windows Performance Monitor.**

- e. **An Integration Services Logging Reporting Pack is downloadable from the Microsoft Developer Network. Like the other packs, this is shared source and therefore fully customizable and extensible. The pack includes summary and detailed reports. Summaries include graphical views of Package execution for viewing or monitoring historical execution time of a Package, along with successful or failed executions. Detailed reports include all error or diagnostic messages raised by Packages.**
 - Explain the mechanisms available for monitoring execution and sending alerts if there are problems. **Covered above**

8. Pricing

Since pricing is so variable we asked for information about the licensing policies and how costs are calculated.

For ETL, there is the added complication of sizing. Depending on the tools, you may need a large, small, or no ETL server. You may also need to expand the warehouse or source servers to accommodate the workload. How you configure your environment can affect the pricing of the software.

Basic criteria:

Is there a per seat cost for developers?

Development Edition

Processor Pricing	Server Plus CAL Pricing
\$49	Not Applicable

Is there a per seat cost for administrators?

It depends on case. User can choose either Enterprise Edition or Standard Edition.

Is there a price per server by CPU? Per core?

Standard Edition	Enterprise Edition
\$6K/Proc	\$25K/Proc
\$1.9K/Svr + 5CALs; \$160/CAL	\$14K/srv + 25CALs; \$160/CAL

Is the price different for different server operating systems?

No

Are there per source or per target server / instance charges?

No. If you buy per proc, there is no CAL. If you buy per SRV, it has CALs associated.

Are there additional source / target connector charges?

No, if customers want it, they need to buy additional CALs.

Is there a charge for development or test environments? If so, is it the same cost?

The customer will need to buy a SQL license for each environment. The customer can reduce the overall cost for development by purchasing the Developer Edition.

How is maintenance charged? What is the range if it is a percent of some set of costs?

<http://support.microsoft.com/?LN=en-us&scid=gp%3Ben-us%3Bofferprophone&x=9&y=17>

Business-hours Support-Call Now for \$259 U.S.	Business-critical After-hours Support** for \$515 U.S.	OR Submit an Online Request for \$99 U.S.	OR Call to Order a 5-Pack Phone Support Contract for \$1,289 U.S.

How many different editions or bundles are offered?

Editions See Feature Comparison Matrix	Typical Workloads	Processor Pricing Learn more	Server Plus CAL Pricing Learn more	Where to Get It
Enterprise Edition Meets the high demands of enterprise online transaction processing and data warehousing applications	OLTP Data Warehousing Data Mining	Retail* \$24,999 Example** \$23,911	Retail* \$13, 969 with 25 CALs Example \$8,487 \$162 per additional CAL	Volume Licenses Find a Microsoft Licensing Reseller (U.S. only) Single Licenses Buy online from the Microsoft Product Information Center
Standard Edition Data management and analysis platform for small and medium-sized organizations	E-commerce Data Warehousing Line-of-Business Solutions	Retail* \$5,999 Example** \$5,737	Retail* \$1,849 with 5 CALs Example** \$885 \$162 per additional CAL	MSDN Subscriptions More Questions? In the US, call (800) 426-9400 to speak to a Microsoft representative.
Workgroup Edition Designed for small organizations that need a database with no limits on size or number of users	Front-end Web server Departmental or branch office operations	Retail* \$3,899 Example** \$3,700	Retail* \$739 with 5 CALs Example** \$730 per server \$146 per additional	In Canada, call (877) 568-2495.

			CAL	Worldwide Visit your local Microsoft Web site or contact your local office
--	--	--	-----	--

Editions See Feature Comparison Matrix	Typical Workloads	Processor Pricing Learn more	Server Plus CAL Pricing Learn more	Where to Get It
Developer Edition May be installed and used by one user to design, develop, test and demonstrate your programs on as many systems as needed.	Includes all the functionality of Enterprise Edition	\$49	Not Applicable	Buy online from the Microsoft Product Information Center MSDN Subscriptions
Express Edition An easy-to-use, lightweight, and embeddable version of SQL Server 2005. Free to download, redistribute, and embed.	Free Download	Not Applicable	Not Applicable	Download the Express Edition for free MSDN Subscriptions
Compact Edition A free, compact embedded database for single-user client applications for all Windows platforms	For all Windows platforms, including Tablet PCs, Pocket PCs, smart phones, and desktops.	Not Applicable	Not Applicable	Download the Compact Edition for free MSDN Subscriptions

Are there additional charges for team-based development, e.g. source control, project-level security, role-based access?

No, SQL Server has no additional charge for these items

Please provide an estimated list price and support cost for these two scenarios:

Please be aware that below pricing are made by assumption. Please contact your local marketing for related questions.

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

4 Dev Edition: \$196

4 Proc Standard: \$23,996

25% of total for support: \$6,048

Total: \$30,240

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 server 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

We charge by per processor not per core.

Production Environment:

8 core, 30GB Memory, 1 TB disk size

The cost based on assumption:

8 proc machine and each proc has 4 core, support is around 25% of total

So the cost is $(8 * \$24,999 + 8 * \$5,999) * 125\% = \$309,980$

Non-production Environment:

The cost based on assumption:

8 proc machine, 10 Dev and 1 standard edition:

$(8 * \$5,999 + 10 * \$49) * 125\% = \$60,602.5$

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?
Volume of source data has rarely been an issue for Integration Services projects, but where source volumes have been in the terabyte range, parallel extracts have been useful.
- 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?
Multiple Source Adapters can be used in a single Data Flow, each using a differently ranged source extract query. Typically the downstream processes will also be duplicated. Design of these parallel processes is easily accomplished by copying and pasting a single completed flow, then editing the source query on the copy. This can be done multiple times. The degree of parallelization used in executing this data flow is automatically determined by the capabilities of the host system at runtime, and is controllable at design or runtime by setting properties on the Data Flow.
- What features are available for caching of source data or lookup values, and what are the limitations?
Source data is not cached by SQL Server Integration Services. Lookup data is cached per execution using either a full cache (all reference data is loaded into memory during a pre-processing step) or a partial cache (as lookup “hits” are found, matched values are cached for subsequent use. The size of the partial cache can be set by the designer. In SQL Server 2008, it is also possible to share lookup caches across multiple components, if desired.
- 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?
Integration Services can run an extract Package on multiple servers by invoking SQL Server Agent Jobs on those servers, each job pointing to the same metadata store to retrieve the same Package. Agent Jobs can be

- invoked on a schedule, or on demand. To invoke an Agent Job on demand, you can use the SQL Server stored procedure sp_start_job, or within an Integration Services Package you can use an Execute Agent Job task.**
- Can individual extracts be parallelized to run across servers?
SQL Server 2008 supports parallelized execution of individual extract packages on one host only. Performing one extract across multiple servers requires constructing a separate package to issue agent jobs and then join the intermediate results as appropriate.
 - Are there grid / concurrent processing capabilities? If so, how do they work?
SQL Server 2008 has no out-of-the-box grid processing capabilities for ETL packages, however, concurrent processing of extracts can be implemented using techniques previously described.