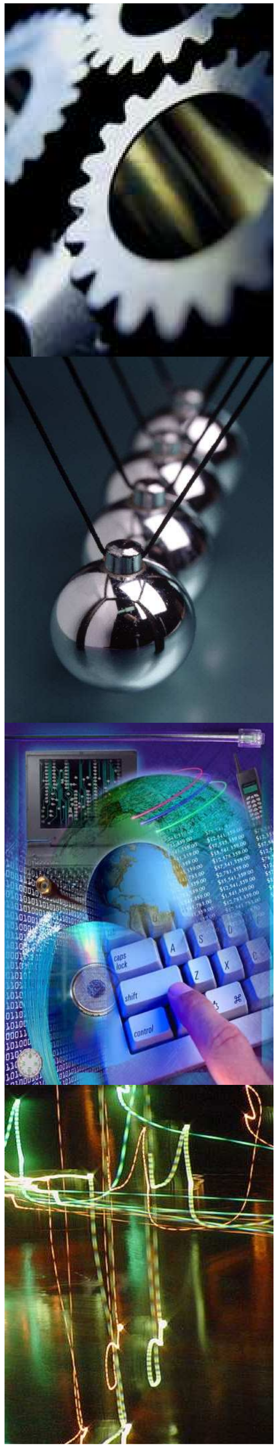


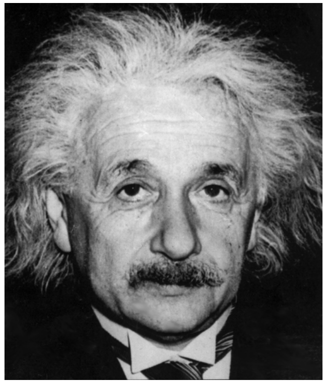
What you need, when you need it

ETL Tool
Evaluation Guide

Public Information
Version 3.1: 1/1/2012

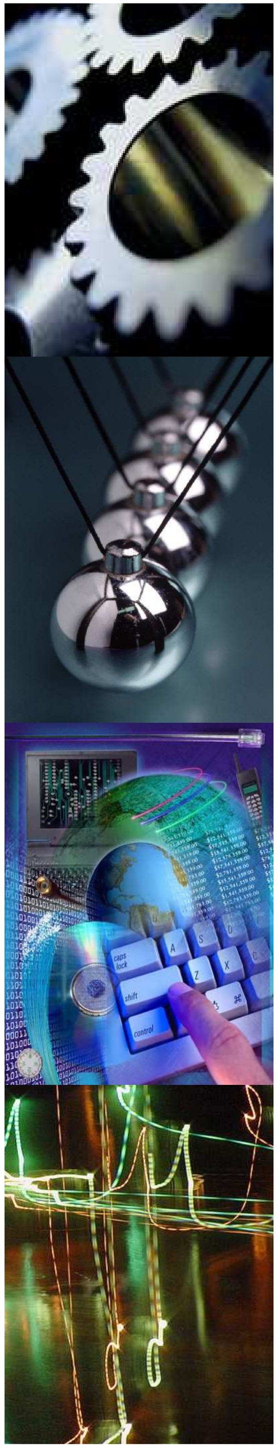
Why Instant Business Intelligence?

“The significant challenges we face today cannot be resolved by the same level of thinking that created them.”



Albert Einstein

Table of Contents



Introduction

Introduction

A

White Paper

by

Instant

Business

Intelligence

www.instantbi.com

Hello and Welcome to our ETL Tool Evaluation Guide.

Over recent years we have received some inquiries from prospective clients asking us to compare SeETL to other ETL products. We have also had a number of our partners ask us to provide a detailed comparison with other ‘vendor’ ETL products.

We feel this is a little strange because we believe the most important two features of SeETL are:

- SeETL is less than 10% of the price of ‘vendor’ ETL tools.
- SeETL is around 10x more productive than the ‘vendor’ ETL tools.

We advise our clients and prospective clients that, as good as SeETL is, they cannot expect SeETL to be the feature / function equivalent of a vendor tool costing 10 times as much.

We advise our clients that the ETL tool vendors are in the ‘feature/ function’ stage of their battle and today. Just like Microsoft Word, these tools are having features built into them that most companies will never use, yet they are included in the price.

Here at Instant Business Intelligence we are building ‘what you need’ to build large dimensional Operational Data Stores and Data Warehouses. Nothing more, nothing less. We are not adding lots and lots of features you will never use.

We believe that your SeETL decision criteria should be focused on:

- Can I build what I need to build with SeETL? Yes or No?
- Price.
- Productivity.

To help you determine if SeETL can do all that you need we provide the Mapping Spreadsheet, SeETL DesignTime (SeETL^{DT}) and a fully functioning version of SeETL RunTime (SeETL^{RT}) software for your evaluation. We provide all this at no charge. We also provide assistance to clients for ETL tool evaluations for a standard consulting rate.

When building your EDW prototype you must define your source to target mappings, and the most likely place you will define them is inside a spreadsheet. The mapping spreadsheet we provide is the evolution of 15 years of experience at documenting ETL mappings.

Now that SeETL can generate 95%+ of all ETL directly from the mapping spreadsheet you can build your EDW prototype faster with SeETL than you can with any other ETL tool.

We propose that you might as well use our ***free*** evaluation version of SeETL to build your prototype because it is the fastest way to build a prototype.

The selection of an ETL tool can be a significant investment if a vendor tool is chosen and it is one you cannot easily change once made.

We advise our clients to make their ETL decision after they have built their prototype SeETL.

You will know much more about your true ETL needs when you have built your prototype. Whether you then buy SeETL or not you will have gained great benefit from using SeETL evaluation version for your prototype.

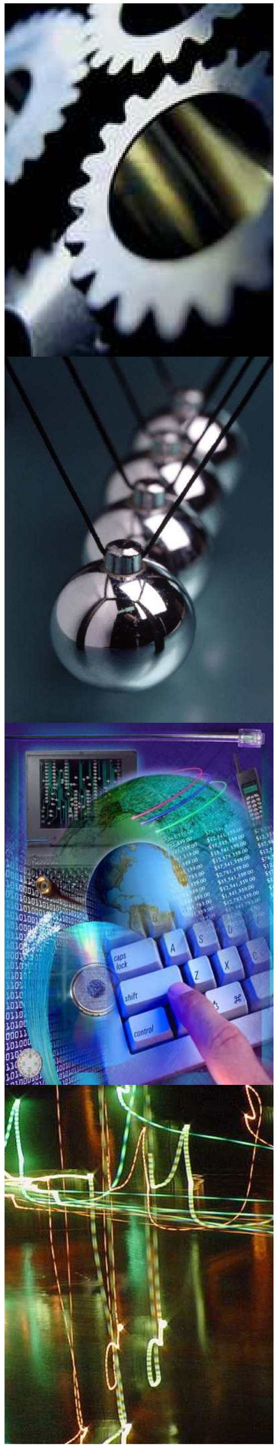
In cases where our consulting clients purchase a vendor ETL tool we actually use SeETL to build the prototype EDW because we cut 6-8 weeks off the elapsed time of the project ***and*** we take ETL development off the critical path of the project.

As you consider SeETL we ask that you please keep in mind:

- We wrote SeETL for ourselves.
- We use it every day at our clients.
- We are experts in many of the vendor ETL tools.
- We know that we are offering a product with compelling features at a compelling price.

Having said all that, we have decided to release this ETL Tool Evaluation Guide to assist you to select your ETL tool based on the real world needs of the many large EDW clients we have worked with over the last 20 years.

We hope you find this Evaluation Guide valuable.



Features of an ETL Tool to Evaluate

Public Information
Version 3.1: 1/1/2012

instantBUSINESS INTELLIGENCE

Overview of Detailed Features

The way we present the ETL Tool Evaluation Guide is in two parts.

- In this document we will provide detailed descriptions and examples of features which are valuable in an ETL tool based on our experience.
  We have made every effort to group the features and explain them in such a way that people with little experience with ETL tools will understand.
- In a separate spreadsheet we have documented these features in a matrix such that you can evaluate a number of ETL tools based on the spreadsheet and this document.

We do not maintain licenses for the latest versions of all the vendor ETL tools. They are very expensive.

Hence, at any given time we are not able to provide a detailed comparison of the 'latest release' or the 'marketing release' of the vendor ETL tools. This is why we have not rated the vendor tools against each feature we have documented here.

We are experts in many of the vendor ETL tools. If you would like, our consultants are available to perform an on-site evaluation of the latest and greatest versions of the vendor ETL tools if you can get the vendors to provide you with the evaluation versions!!

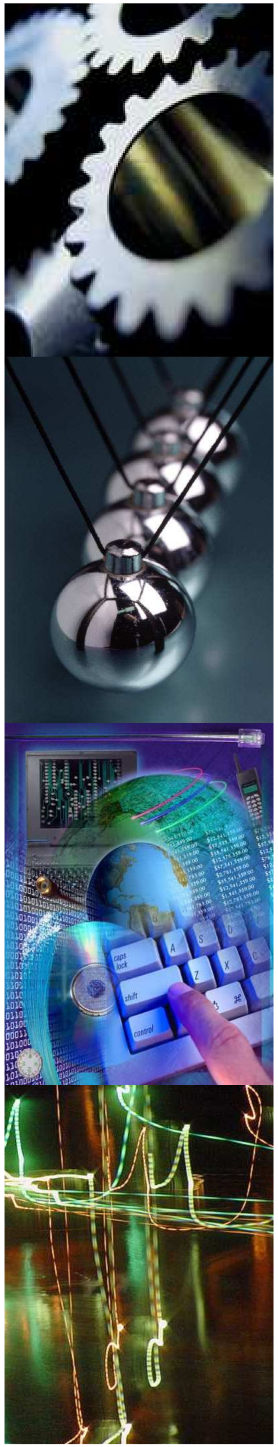
We would be pleased to perform such a consulting service for you.

Please note all SeETL *features described in this paper refer to the 3.1 version of the* SeETL *which was released in January 2012.*

Summary of Groups

We have defined the following groups of features and they are documented in detail on the following pages.

- Commercial Considerations
  - Pricing, Open Source, Support, consequences of 'take overs' which have been rife in the ETL market.
- Productivity Features
  - After all, the vendors all claim ETL tools 'improve productivity'
- Portability Features
- Scalability Features
- Degree of control of the ETL Designer
- Ability to have new features added



Commercial Considerations

Public Information
Version 3.1: 1/1/2012

Commercial Considerations and Features

CC01: Price.

The price of an ETL tool must be carefully considered. Some vendors require you to purchase a license for all the CPUs visible through the operating system instance on which the ETL tool runs. Nearly all the ETL vendors link their price to number of processors or number of machines. Inspect the price carefully. Remember that maintenance is often calculated on list price and is usually in the range of 15-25% of the list price not of the price paid for the license.

A major feature of SeETL is the price. We are taking the path of providing our product at the lowest possible price to make it possible for more companies to gain the benefits of Business Intelligence.

SeETL is licensed on three difference schemes:

1. Purchase of a single run time copy for windows to be used on one machine only.
2. Purchase of a source code copy which can be used on as many machines as you would like inside one company.
3. For Software Developers.
   1. Purchase of a source code copy with subsequent purchase of licenses for each 're-sold' version. You must inform Instant Business Intelligence of each licensed client.
   2. Purchase of an unlimited source code license.
      You can implement as many copies of the software as you like in as many clients as you like.

CC02: Vendor Stability/Reliability/Openness

Carleton, Prism Solutions, ETI. These were the 'big three' in the ETL space not that long ago. We then saw an explosion of ETL tool vendors followed by an implosion of consolidation and buy outs. This process has cost thousands of users of ETL tools significant amounts of money. Now that significant consolidation has taken place the vendors IBM (Ascential), Oracle (Carlton), Cognos (Decision Stream) and Business Objects (Acta), then Business Objects into SAP, all have reason to reduce the 'openness' of their tools to further consolidate their position with the clients who are using the tools as well as new clients who can be sold these tools.

Informatica remains the only major independent ETL tool vendor, but for how long?

When considering your ETL tool investment you should consider:

- The stability and reliability of the vendor providing the tool. Consolidation of small ETL vendors will continue and the 'buyer' may or may not continue support/development of the tool. Some vendors did not provide a migration path between the tools they purchased and their own tools.
- The track record of 'openness' of the vendor and the likely costs should you want to change other components of your EDW environment that you may have also purchased from the same vendor. Some vendors are quite notorious for 'locking' clients into a product and then charging high fees.
- If you have purchased multiple dependent components from a single vendor the deterioration of the bargaining position that you will be in when it comes time for upgrades and maintenance fees. If it is expensive to move and your vendor knows this, they have no motivation for providing future discounts.

A major feature of SeETL is that it is available as 'open source'.

Whether Instant Business Intelligence continues on or is bought out, your license is a perpetual open source license. There is a growing user community that knows the source code to look to for support. The source code is written in C++/ODBC and these skills are readily available. We believe that 'open source' is the model for the future for software development.

'Open Source' is your guarantee that you can continue to use and get support for SeETL for years to come.

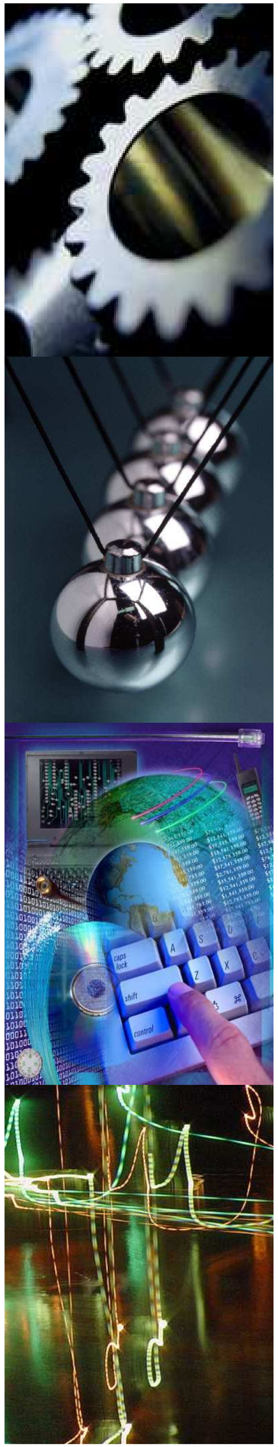
Commercial Considerations and Features

CC03: Development System Licenses.

Because most ETL tools are 'code' based and changes to code require testing it is usually necessary to maintain some sort of development and testing environments. Using the production license/system of your ETL tool for development and testing introduces the possibility of testing affecting your production system. Most vendors will (rightly in our opinion) advise clients to run a second copy of the ETL tool on a second machine. And they will ask for a development license fee.

A major feature of SeETL is the price. We offer a source code license and with the source code license we allow our clients to run as many copies of our software on as many machines as they like within one company.

For clients who buy runtime only licenses, we do not charge for development licenses.



Productivity Features

Productivity Features

PR01: Can you generate the ETL jobs from the mapping spreadsheet?

No other ETL tools can generate 95% or more of the ETL 'code' from the mapping spreadsheet. With other ETL tools the mapping spreadsheet is passed from the ETL Architect/Designer to the ETL Developer. (A link that introduces many 'slow downs' by itself.)

The ETL developer then designs and builds a job based on the ETL Specification. Then the 'ETL Specification' and the 'ETL Job' are defined in two places requiring dual maintenance.

SeETL is currently the ***only ETL tool in the world*** that can generate 95%+ of all the ETL required directly from the mapping spreadsheet. This has boosted the productivity of SeETL to more than 10x that of any other tools.

If you cannot generate your ETL jobs directly from the mapping spreadsheet you cannot get within an order of magnitude of the development productivity of the ETL jobs possible SeETL.

PR02: Cost of learning the tool.

The first indication of productivity for a tool is 'How long does it take to learn it'? Ask your vendor what the standard classes are. We advise our clients not to believe the line "You don't need to go to the standard class the tool is so easy to use." The standard classes are there because people need them to learn the tools. Otherwise no-one would pay for them and they would not exist.

Most ETL tool vendors have 4-5 days training to learn the tool. And they charge good money for attendees to learn the tools!

In contrast there is no 'training' for SeETL. If you can use a spreadsheet and you understand how to create tables and views you can use SeETL. The main reason for this is the 'simplicity by design' of SeETL. The major areas of 'training' we do with our clients are in the development of the data models and the development of reports. We run no training on the usage of SeETL. We provide all that support remotely by web.

PR03: Are there new interfaces to learn?

Most ETL tools have a new graphical interface to learn. They vary in quality and productivity. However, each new person using the tool must learn to use this new interface. This costs time and money for every new person that uses the tool. Further, if a person does not use the tool for a period their skills in using the interface declines and they may be less productive when they start to use the tool again.

In contrast there are no new 'interfaces' when using SeETL. We have used Microsoft Excel, Word, Access, text editors and a simple 3 button application to load the mapping spreadsheet and perform the generation of the ETL. We have provided extensive and detailed documentation most of which our clients tell us they never read!!!

PR04: How many developers and development licenses are required?

The answer to this question is a 'dead give away'. ETL vendors often charge extra fees per developer license. Though it depends on the complexity of the EDW being built your prospective ETL vendor should be able to give you an idea of how many developers you are going to need based on the number of source systems, number of source files, number of source fields, approximate number of rows in each file and likely number of target fields in the EDW.

Experienced EDW consultants can usually provide an estimate accurate to within a work month or two to build an EDW given these details.

If you really want to know how many developers will be required and what the likely cost is ask the vendor for a fixed price estimate for the ETL development using their tool and their consultants. If you ask for the proposal like you might really buy it you will learn a lot more about the productivity of the tools!

In contrast SeETL does not require ETL developers. SeETL makes ETL developers redundant. The ETL Designer/Architect defines all mappings in the spreadsheet and the DBA builds the database. The ETL Designer/Architect can then validate the ETL with the aid of the DBA if necessary.

Productivity Features

PR05: What is the effort required to build 100 dimension tables and 60 fact tables?

Ask your vendor to provide references for how much effort it takes to build 100 dimension tables with an average of 20 columns each and 60 fact tables of approximately 10 keys and 20 data fields each.

For SeETL RunTime prior to the SeETL DesignTime this process took 15 work days total for one ETL developer. With the SeETL DesignTime it takes the ETL Designer/Architect approximately the same amount of time to add and test every detail of the mapping spreadsheet so that it generates code correctly. That is, about 15 work days!! The rest is actually getting the mappings correct, which must be done no matter what ETL tool is used. Regeneration of all the ETL rules takes just a few minutes. This is only because we do not execute the creation of views directly against the database. We allow the DBA to cut/paste the views so he/she can see any errors.

PR06: Is the tool really codeless?

Most ETL tool vendors sell their tools as 'codeless'. But, if you scratch the surface, you will find 'code'. It is often generated code, but it is still there. We advise our clients to look for the 'code'. Whether it is a 'mapping' written in a GUI and stored as a set of complex parameters in a database or whether it is a 'job' written in a GUI that generates some underlying code that is then executed, this is 'code'.

Well, what's the problem with 'code'? You may ask.

Nothing really. Except…

When you write an ETL job and some 'code' is written somewhere in order to run that job a number of productivity 'problems' arise:

1. There is an opportunity to introduce 'bugs' in development as well as in later changes made to the 'mapping/job'. Therefore you need to perform extensive testing to make sure the job works in the first place and you must perform extensive testing to make sure the job works after any changes.
2. The code and the data structures are collapsed. Therefore the code must be reproduced for every instance of the data structures. So if you have 100 dimension tables you need 100 dimension table processing jobs/mappings.
3. When you want to make changes, and EDWs are all about 'changes to data structures', you must change your code and you must test it again.

These are very expensive and time consuming exercises.

SeETL is truly 'codeless'. There is no code generated, there is no separate 'repository' defining mappings. There is only the mapping spreadsheet and the database catalog storing views. There is very little testing required because there is so much less opportunity for the ETL Architect to make mistakes. Indeed, because the views are generated even a 'typo' in the field names of views is no longer an error. The view will be generated, the data will be moved, it is just the name of the field in the view will be a 'typo'. Being truly 'codeless' is a major difference and a major factor in SeETL being so productive.

PR07: Are the 'code' and the data structures separated?

Experience tells us that separating algorithms and data structures enhances productivity and system quality/reliability. This has been known since the late 70s. For example, ERP developers have gone to great lengths to separate Business Logic and Data Structures.

But most ETL tools collapse data structures and processing. This means you must have one job/mapping per input/output combination. That is 100 dimension tables requires 100 jobs. With most ETL tools it is common to have a 1-1 relationship between tables and jobs. This is a huge drain on productivity.

SeETL separates data structures and algorithms. Each 'type' of processing has one and only one executable program. This program is controlled by parameters and the program is intelligent enough to 'discover' the mapping of data between sources and targets at run time. So, for 100 dimension tables SeETL requires one program and 100 sets of parameters invoking that program.

Usually this is as simple as 'RunType1Dimension <table name>' being entered into the batch schedule tab of the mapping spreadsheet.

Clearly, this separation of algorithms and data structures is the more productive way to write ETL jobs.

Productivity Features

PR08: Does adding new columns require 'code' changes?

One of the most common tasks in maintenance of an EDW is the addition of columns to dimension/fact tables. Most ETL tools require that you edit the 'job/mapping' at a field level to add the new field.

Any 'wizard' that helped build the job in the first place is not normally used to edit the job in the second place. Further, since the 'code' has changed it is prudent to 'test' the change in a test environment before moving it to a production environment.

SeETL separates algorithms and data structures. Therefore to add a new column you only need to add it to the mapping spreadsheet and regenerate the appropriate views. (You also need to change the physical database of course.) As long as the create view statements are properly generated and can execute properly there is nothing else to 'test' or to 'change'. You may like to make sure that the data types of the target and source are compatible by running our metadata checking utility for the table. However, because of the separation of algorithms and data structures you do not need to retest the algorithm because of a change to the data structure. Thus maintenance of ETL jobs is far more productive in SeETL .

PR09: Does the tool codelessly support multi-level dimensions?

Most data warehouses have many levels of dimensions. For example, day, week, month, quarter, half year, full year are all 'levels' of time. Postcode, county, city, stateregion, state, countryregion, country, economic block are all 'levels' of geography. Most ETL tools require that you write a job/mapping for each level of each dimension if you want to be able to navigate into the fact tables at that specific level rather than summarise a lower level to the higher level. This means the number of jobs to be written and managed is proportional to the number of levels of dimension data you want to be able to have. This can be a multiplier of 2-5 times. We have seen large companies implement large numbers of jobs like this on the advice of their ETL/BI tool vendors.

SeETL supports the creation of separate tables per level just like all other ETL tools. It also supports the ability to manage 10 'levels' of a dimension table inside the one dimension table with no coding required, just the definitions of the summary levels in the mapping spreadsheet.

Further, SeETL supports the creation of these multiple levels inside the dimension table processing programs. There is no separate processing to run, there are no extra jobs to run and manage. One invocation of one dimension processing program can manage the updates of up to 10 levels of summary. With SeETL , the examples of time and Geography can be created and managed by one program invocation each and one physical dimension table. This is a major productivity improvement.

PR10: Does the tool 'codelessly' support summaries and the addition of summaries?

Most tools require the developer write some form of code to create summaries. The tool must be told what dimensions to summarise on and what fields need to be summarised. This information must be encoded somehow inside the tool.

When a new summary is required, most tools and designers will advise their clients to create new tables and new jobs to manage the new summaries.

SeETL supports the ability to 'codelessly' generate summaries based purely on a control table. Now, to add a new summary you simply add a new row to the mapping spreadsheet 'Aggregation Control' tab. There is no code to change, no tables to create and no IT support required to add a new summary level to the data warehouse.

PR11: Does the tool 'codelessly' support incremental update of summaries?

Most tools require the developer to write some form of code to update summaries. Indeed, in many cases we see summaries are rebuilt every run of the batch because of the complexities of doing incremental update in the ETL tool.

SeETL supports the ability to 'codelessly' incrementally update summaries on all supported databases.

Productivity Features

PR12: Is the tool truly 'typeless'?

One of the areas that costs a lot of time in development of ETL jobs/mappings is making sure all the data types of all the columns are correct. Some tools generate warning messages when there is some sort of 'data type mismatch'. Getting rid of these messages can be quite time consuming and a significant drain on productivity.

SeETL is truly 'typeless'. SeETL moves data from source to target based on the data being converted to a C character string. When it sends data to the ODBC driver it requests the ODBC driver perform an explicit data conversion from the character string to the target data type.

If the string can be converted it will be and SeETL will send the data to the database. If it cannot be converted an error message is issued and processing is stopped.

We have also provided a metadata checking utility which checks sources and targets separately and issues messages where it considers there might be a problem with the data being moved. This has proven to provide a great productivity boost in finding such things as truncated fields or fields where characters were being accidentally moved to numeric fields in complex mappings.

PR13: Does the tool codelessly support common data correction and reformatting tasks?

Dates and numerics!! When being sent data that is in flat file format extracted from some old file processing system we often see dates in all sorts of formats and we see numerics coming to us with ',' and '.' inside the string.

Most ETL tools require you to write a code fragment for the reformatting of each different format of such things as dates and numerics coming into the ETL system as strings. Further, they require you to insert the call to this code fragment inside the mapping/job.

SeETL provides the 'Data Correction Utility'. This tool can reformat 52 different date formats. It can also remove leading blanks, remove ',' or '.' inside strings that are being sent to numerics, all codelessly. This tool was developed for a client and saved weeks of detailed coding work.

Productivity Features Summary

We set out to build the worlds most productive ETL tool.

We built the current version of SeETL from the ground up to be the productivity tool of choice for our consultants when building large Enterprise Data Warehouses.

We constantly look for cases where the SeETL tools can be extended to save time and money for our clients.

As a result, SeETL is at least 10X more productive than any other tool we have had the pleasure to use.

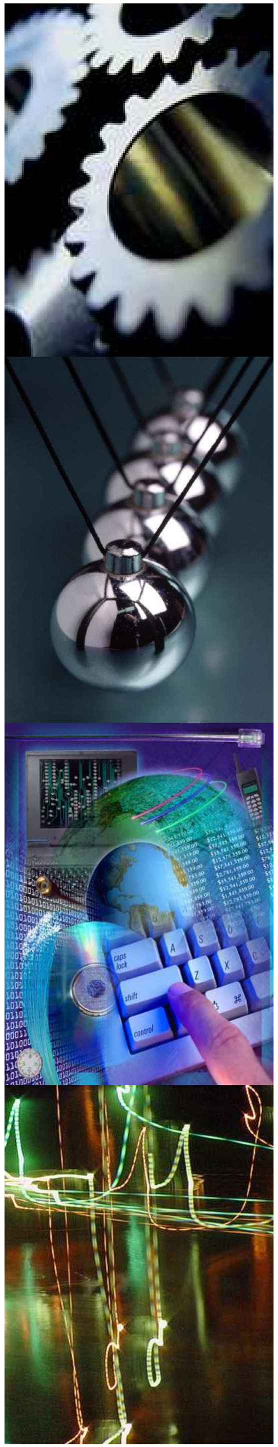
On a productivity basis, SeETL is the world's leader by far.

SeETL will now remain the world's productivity leader because it takes no time at all to generate the ETL, even for the largest and most complex Enterprise Data Warehouses we work on.

The productivity breakthroughs of SeETL have removed the writing of ETL jobs/mappings as a substantial cost in the development of ETL in Enterprise Data Warehouses.

The productivity breakthroughs of SeETL are the foundation component of our vision of:

Enable all companies everywhere to benefit from the ability to profitably implement Enterprise Business Intelligence Solutions.



Portability Features

Portability Features

Portability Features

Fifteen to twenty years ago data warehouses were predominantly built on mainframes using Cobol as the ETL language. In the last 15 years we have seen the rise and rise of Oracle, Sun, HP and AIX as various platforms for Data Warehouses. Today, in Business Intelligence, we are seeing Lunix and MySQL do to Oracle/Unix exactly what Oracle/Unix did to mainframes running DB2. We are also seeing appliances come along.

We are also seeing Microsoft making efforts in the data warehousing area with SQL Server and their appliance product.

A major lesson to be learned from the last 20 years of IT is that portability of applications is important.

If the application can be quickly, easily and cheaply ported to a new operating system or new database we might have the opportunity to reduce our costs as new vendors bring new and innovative products to the marketplace.

Unless you want to be tied to your EDW HW/OS/ETL/Database vendors for a long time, paying premium prices because they know you cannot move, portability is a major feature you should look for in your ETL tool.

PO01: Can a ‘mapping/job’ be moved from one operating system to another with no change at all in the job/mapping?

We advise our clients on the importance of no change at all. Many tools are aware of the differences in newlines between windows and unix based operating systems. Also, many ETL tools have file names defined inside jobs and these file names must be changed when moving from one operating system to another. Most tools cannot move ETL from one operating system to another with no changes.

SeETL is written in C++/ODBC and compiles natively on each supported operating system. At ‘run time’ SeETL is not aware of the operating system that it is running on and so treats all operating systems in exactly the same way. We have written SeETL to the ‘lowest common denominator’ of the underlying operating systems.

The only change required when moving SeETL from one operating system to another is to change the parameters to the commands because the names of files and ODBC connections are likely to have changed. This is a 5 minute job for the ETL Architect and it is done in the mapping spreadsheet.

PO02: Can the ETL be moved from one database to another with no change at all in the job/mapping?

We advise our clients to inspect this area closely with ETL tools. It is somewhat surprising to us that most of the leading ETL tools do not allow you to easily move the ETL jobs between databases.

We advise our clients to actually try out the ETL tool and move jobs between databases to see the issues that arise for themselves.

Further, most ETL tool vendors push their ‘native’ database drivers over ODBC, especially the Oracle drivers. However, if you use the native Oracle drivers of any ETL tool you will need to spend significant amounts of money to migrate away from that environment.

The usual claim is that native drivers are faster than ODBC drivers. This is no longer true. DataDirect now offer ODBC drivers that outperform the Oracle Client (OCI). It is true that ODBC drivers have slightly less functionality than the Oracle OCI, however, it is quite rare to use these extra functions.

Some ETL vendors will claim that you can use their ODBC drivers and therefore get easy movement from one database to another. This is often not true and you should test the truth of any such claim. (We have had experiences where the vendor supplied ODBC drivers truncated decimal places where the OCI driver did not.) You should also be aware that different databases present their data types differently to ODBC and these differences can generate data type mismatches in ETL tools. For example, in Oracle an integer is really NUMBER (10,0) and in SQL Server it is really an integer and these are considered different data types by ODBC.

SeETL is completely ‘typeless’. It retrieves the data type of the fields from the database via ODBC at run time. As long as the database and the ODBC driver supports the ODBC 3.51 specification SeETL will work.

Therefore the database can be moved between any of the supported databases. We even provide the ‘Data Transfer Utility’ to assist in any such data movement. You will incur no ETL re-write cost in moving SeETL between supported Operating Systems and Databases.

Portability Features

Portability Features

PO03: If we move the EDW database can the ETL tool move the data as well?

All ETL tools can assist with moving the EDW data from one database to another.

Most tools will require you to write one job per table and will insist on moving data at field level. Therefore it is usually quite a time consuming job to move the data for large numbers of tables from one database to another.

We advise our clients to test these capabilities to assess for themselves the effort required for ETL tools under consideration.

SeETL provides the free Data Transfer Utility. This utility is a high function utility that can move data between ODBC data sources with great ease and no coding. There is even a 'Transfer' option which will move data directly from one table in one database to another table in another database if the data types are compatible on a field by field basis.

Further, the Data Transfer Utility can generate 'Load Interface File' format data that can be passed into the loading utility of the target database and so further reduce load times for the migration of data.

We have been able to write the jobs and load statements to move 100+ tables in a large DW in a day. Most of that time was writing the load statements.

We advise our clients that there are cases when data does not move smoothly because of the difference in data types and representation of data in those data types. Oracle NUMBER is one of these. NUMBER means 'float' to Oracle and can cause problems when passed to SQL Server. In some cases a view must be placed over the source table to reformat the data being extracted to a string format that is acceptable to the target database via the ODBC driver being used.

PO04: What Operating Systems are supported?

Every vendor will have a list of the operating systems that are supported. We advise our clients to be fully aware of Operating Systems Supported. Further, we advise our clients to ask the 'deployment order' of operating systems. This will reveal the 'relative importance' of the operating system that you are considering using. For example, many vendors place HP-UX last in their deployment order. And you should know this if you are considering HP-UX.

SeETL is written in C++/ODBC 3.51. On Windows it compiles using Visual Studio .Net 2003.

On Solaris/AIX it compiles using GNU C++ 2.9 or higher. It also compiles on the IBM AIX Visual Age compiler.

SeETL has been written to the 'lowest common denominator' for operating system support and has been written in ansi standard C++ as much as is possible.

We are confident it will run on any platform that supports the GNU C++ 2.9+ compiler.

We are adding support for various operating systems according to demand from clients. Our deployment order is windows, Solaris, AIX.

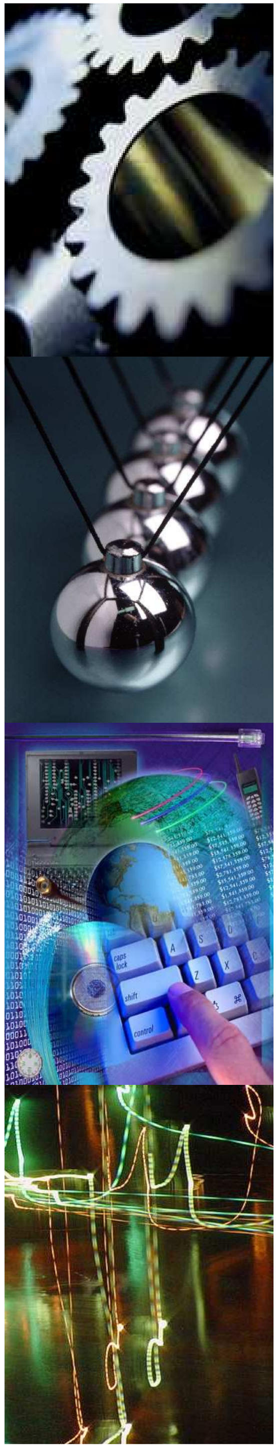
The operating system most used by our clients is windows 2000+. Since we provide source code our clients can move the SeETLRT to any operating system they would like and still receive support.

PO05: What target databases are supported?

Every vendor will have a list of the databases supported.

As a target EDW database SeETL supports Oracle 9, SQL Server 2000, DB2 UDB 8, Sybase ASE 12.x, Sybase IQ 12.x, MySQL 5.x. It will support any later versions of these databases where the ODBC 3.51 interface continues to be supported. Most of our clients use SQL Server or Oracle. We have also used DataDirect ODBC text drivers as a data source quite successfully.

We are starting to see more interest in MySQL because it is free.



Scalability Features

Scalability Features

As the volumes of data being placed into EDWs increases it is not enough to just 'throw more processors and memory' at the ETL processing problem.

Given the same amount of 'processors and memory' an ETL tool that implements some form of scalability features will process more data than an ETL tool that does not. In the worst case, some ETL tools have inherent bottlenecks/restrictions that must be programmed around as volumes increase.

We have been working in large systems programming for 23+ years. We are fully aware of where the bottlenecks and problems occur. And we have developed SeETL to avoid these problems.

SC01: The price of scalability.

We advise our clients to check to see if there is an extra 'price tag' on the 'high scalability' features of an ETL tool. It is often the case that the vendors will talk about 'high scalability' but not mention that there is a separate fee to turn the option on or that the 'high scalability' product is actually a separate product.

The scalability features of SeETL are part of the purchase price. We do not charge extra for larger volumes. We do not charge extra for more processors.

We have had some clients recommend to us that we should be charging larger clients extra, usually on the basis that they should then be charged less!!

Today, larger clients can buy the unix source code license of SeETL for exactly the same price as smaller clients buy the windows source code license and therefore obtain a proportionally larger benefit.

This is just good luck for the larger clients!!!

Our position is that our prices are subject to change without notice.

SC02: Does implementing high scalability require 'code' differences?

With many ETL tools, not only is the 'high scalability' feature set an 'optional extra' it is often the case that the job/mapping that is written must be altered to take advantage of the 'high scalability' features.

Further, in at least one case we are aware of the overall flow of data must be written differently, some custom code must be written, and complex jobs streams and job checking must be written to achieve 'straight line' scalability from the ETL tool.

In at least one case we were surprised to find a significant limitation on the amount of data that could be placed into an 'in memory file' which severely limited the scalability of the ETL written. We were required to significantly re-write our ETL jobs to avoid this 'surprising' limitation.

We advise our clients to check whether taking advantage of the 'high scalability' features of the vendors ETL tools requires changes in the code.

SeETL does not require any extra code to be implemented to take advantage of 'endless scalability'.

The main 'endless scalability' features are turned on simply by making changes in the 'Dimension Table Load Control' tab of the mapping spreadsheet.

With the implementation of 'endless scalability' at no extra cost we now recommend to our clients that they turn on the memory mapped IO features of SeETL all the time.

The only areas where we do not recommend this is where clients want to run very small batches of records. This is because the time required to load the dimension tables into memory exceeds the processing time of simply reading the rows needed directly from the database.

Scalability Features

SC03: What scalability features are available?

We advise our clients to ask their ETL tool vendors to document their scalability features. You should check to see if the vendor has truly implemented a wide set of features to take care of the major bottlenecks in processing large volumes of data into a data warehouse. The process that is the bottleneck is the ‘attribution process’.

Instant Business Intelligence has published a public document called ‘SeETL in a Large Scale Environment’. This document provides details of all scalability features. It is available from our downloads page.

SC04: What ‘in memory’ options are available for the attribution processing?

The attribution process is the most expensive single operation in a dimensional data warehouse. We advise our clients to check that the following minimum options are available:

1. Read the lookup table from the database.
2. Load the lookup table into memory so that the lookup can be performed by a binary search, as a minimum, by the process that loaded it.
3. Load the lookup table into a shared memory area (memory mapped IO) so that all processes requiring access to the lookup table can access it and only one copy is required in memory.

We recommend to our clients (if they are a sizable company) that they should not consider any ETL tool that does not support options 1 and 2. We advise our clients to carefully review claims of support for option 3. Some vendors claim this support but it is quite limited. Some vendors only use the database for lookup and claim that the database keeps the row in memory and this is just as fast as a binary search in memory. It is not. Not even close.

SeETL supports all three mechanisms with no limitations.

SC05: Can the ETL system load a subset of the lookup tables into memory mapped files independently of the fact table processing?

For very large clients this is a key feature. Even some of the very advanced/expensive ETL tools to do not support this.

The ETL Architect should have total control over what fact table processing is occurring at any one time and that also means total control over loading into memory mapped files just those dimension tables required by the fact tables being processed at any one time. There is no point loading tables into memory mapped files unless they are being used.

Most ETL tools rely on the idea that the ‘first fact table’ that asks for a dimension table to be loaded will cause the load and then other fact tables processing jobs that require that data will find it. However, this places control of this loading with the ETL tool and not the ETL Architect. It has the nasty side effect that in the case of a failure of the fact table processing for something simple like a full tablespace these in memory tables will be purged from memory and they must be reloaded when the process is restarted. This takes precious time.

We advise our clients to make sure they are aware of whether the control for loading memory mapped files lies with the tool or the ETL Architect.

The benefits of increased control include faster processing times, faster restart times after failure and the ability to place attribution processing onto a machine which does not require database licenses.

SeETL provides the ability to load any named subset of any lookup tables into memory under the full control of the ETL Architect. This control is exercised via the ‘Dimension Table Load Control’ tab of the mapping spreadsheet.

These memory mapped files can be made persistent and will not be deleted just because of a failure in attribution processing. This makes for much faster restarts after failure for very customers with very large dimension tables.

Scalability Features

SC06: Can attribution processing be reasonably run on a machine other than the EDW machine?

There is always a question as to whether the ETL tool is deployed on the EDW machine or on a separate ETL server. This is particularly so if the ETL tool and the database are both licensed per processor. Two smaller 8CPU machines are often much less expensive in software licenses than one larger 16 CPU machine.

However, some ETL tools suffer severe performance degradation if the attribution process is run on a separate machine to the data warehouse. We advise our clients to ask their ETL tool vendors to make recommendations as to whether the ETL tool should be placed on the same machine as the data warehouse itself.

SeETL can be purchased as a source code license and no extra fees are payable no matter how many machines the software is installed on. The attribution processing can be implemented on a second machine with the only slow down being the one time load of the dimension tables into the memory mapped files. It is quite reasonable to implement the staging area on a different machine and even a different database using SeETL. For example, there are great advantages to using MySQL as the staging area database. It is free and it is not queried so often. Doing so avoids license fees for the staging area database. This can save tens of thousands of euros.

SC07: Can batch processing be reasonably distributed across many machines?

In most cases ETL tools contain a scheduler and batches of jobs are run using the scheduler. Most tools can also have their jobs run by an external scheduler. We advise our clients to ask the ETL vendor how batch processing can be distributed across many machines if the ETL tool and scheduler is used on many machines. Often this requires the ETL tool to be installed on each machine with the accompanying license charges. We also advise clients to ask how the batches are co-ordinated to make sure that the ETL Architect has full control over the processing of the batches.

SeETL can be distributed across many machines when using the source code license. Further, SeETL contains a scheduler which can schedule and run any valid command. Using ftp and files as semaphores it is possible to co-ordinate distributed processing of SeETL. If desired the staging area can be on one machine, the dimension table processing on the EDW machine, the attribution processing on the same machine as the staging area and the loading occurring on the EDW machine. All the distributed processing can be fully controlled by the scheduler.

SC08: Can the ETL tool support writing to files for load processing by the database loader?

Most ETL tools support the ability to write data to a load file. However, most tools required more sophisticated programming if any of the fact table records will be updated. This is especially true when performing incremental updates of summary fact tables on a regular basis such as daily. The weekly, monthly, quarterly summaries need to be updated and some records will be updated and some will be inserted. We advise our clients to check with the ETL tool vendor to make sure that load image formats are supported and how updating rows in place is supported.

SeETL provides the ability to produce Load Image Format files directly from the attribution process. Further, it is possible to translate any internal file format file into a Load Image Format. So any file and any table can be reformatted to be of the same format as the Load Image Format for the target database.

Scalability Features

SC09: How well does the tool support updates to large fact tables?

In some cases rows in a large fact table may need to be updated. Most tools have options such as insert then update or update then insert to be able to send a file to a database and update rows if they already exist.

However, for large volumes this is not practical because the volume of logging is excessive. It is better to be able to separate out the inserts to a load file and perform the updates separately. Or it might even be better to delete the rows that will be updated and perform a load for the whole of the new fact file.

We advise our clients to check how the ETL manages the issue of updating rows in a large fact table.

SeETL free Data Transfer Utility provides the ability to delete rows to be loaded if they already exist. This is done by setting the DeleteRowToBeLoaded flag to “Yes” when creating the Load Image File Format for a file.

The Data Transfer utility will perform a lookup for each row being sent to the load image file and if it finds the row in the fact table it will perform a delete of the row. In this way, all rows that are sent to the Load Image File are inserts and can be loaded using the database loader.

For users of Oracle this is no longer needed as the Merge statement can perform a similar function. Therefore, for Oracle, it is possible to send all rows to the working table as a load and then perform a merge to the real fact table.

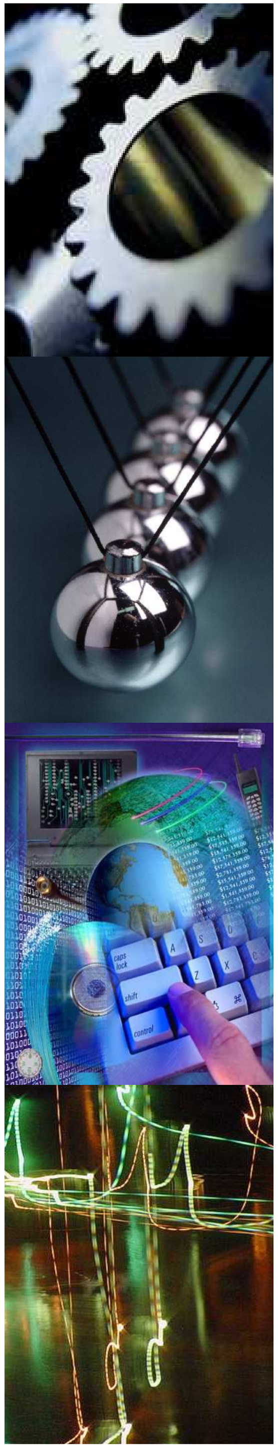
SC10: Can commit frequencies be controlled at connection level?

Some ETL tools do not allow you to easily control the frequency of commits as data is loaded into the database. They force the use of the load utility or force each row to be committed as it is loaded. This is especially true of ETL tools that rely heavily on ODBC because virtually all ODBC drivers default to committing each statement as processed.

Some ETL tools have this commit frequency parameter defined on each icon that connects to a database. This means that ETL programmers must be trained to set the commit frequency when writing the job. It also means it can be quite time consuming to change the commit frequency for a large number of jobs.

We advise our clients to check how commit frequencies are set.

SeETL supports the ability to set the commit frequency for all programs that perform updates to tables. This commit frequency is set at the command level. It is therefore trivial to change.



Other Features to Evaluate

Other Features to Evaluate

A

White Paper

by

Instant

Business

Intelligence

www.instantbi.com

Other Features to Evaluate

We have described many features separately. However, there are some features which are extremely useful and we advise our clients to determine if these features are included in the product, are optional extras, or are not included.

OF01: How do you support detection of deltas from upstream systems?

We are surprised that many ETL tools do not have any inherent ability to determine deltas from upstream systems. Or that it costs extra. 'E' really does seem to mean just 'Extract'. We advise our clients to ask about delta generation support.

Many tools rely on timestamps for rows updated or triggers. This does not help if upstream systems are file based. Many tools do nothing to help you detect deletes. Detection of deletes is a constant problem when defining the extraction process from systems because often these systems do not retain the fact a record was deleted.

Many vendors claim CRC/Hash algorithms are 'good enough' to detect deletes. We advise our clients that use of CRC or hash algorithms to detect changes is not acceptable if the data warehouse must accurately balance to the source systems.

Lastly, we advise our clients to determine how the delta detection handles nulls. The detection of fields changing to/from nulls is often poorly managed.

SeETL provides a 'Generate Delta File Utility'. This utility can compare two files using the internal file format of SeETL and generates the deltas that occurred to the 'old' file to produce the 'new' file. It is fully null aware. The source code is public.

OF02: How do you support nulls?

Many ETL tools do not inherently support nulls in the files transmitted inside the ETL tool. They rely on placing a value inside the field that is interpreted as a null. This, of course, means that that character cannot occur in the data itself. Others define a zero length character string to be null. This, of course, has the problem that there is a difference between a zero length character string and a null to all databases except Oracle.

SeETL uses a 'self describing internal file format'. This is a file which encapsulates in it both the definition of the data and the data itself. When writing the SeETL we considered using XML to define the internal file format but the volumes of data was too great and the speed of processing was too slow.

This 'self describing internal file format' is fully null aware for every field. This is achieved by having a separate null indicator field for every field. Though this does introduce a significant overhead it is the only way to ensure that nulls are effectively handled.

This mechanism is the standard mechanism employed by all databases. Hence we are surprised that it is not also the standard mechanism employed by all the ETL tools.

OF03: Do you use your own separate metadata store to store the details of your jobs/mappings?

Most vendors, by design, use a complex model implemented into a relational database to store jobs/mappings. This separate database is required to provide the very high level of functionality inside their complex tools as well as a way to support the GUIs that are used develop jobs/mappings.

However, what is created must be maintained. And much of the data in these repositories is exactly the same data that is stored in the database catalog. The very fact that a separate repository is implemented means that productivity levels are lowered by the time required to maintain that repository.

SeETL was designed from the ground up to absolutely minimise the amount of metadata stored in order to be able to implement an ETL tool. Where ever possible we used the database catalog at run time to fetch required metadata such as data types of fields. Now, we use the mapping spreadsheet to maintain all metadata and we will continue this for all future metadata.

Using the mapping spreadsheet simplifies the update of our simple repository and vastly increases the productivity of the SeETL as an ETL tool.

Other Features to Evaluate

OF04: How do you bring unformatted binary data into the ETL tool?

Many companies have binary unformatted data coming out of various hardware. The classic is the telco switch. However such things as process controllers and environment sensors also send out unformatted and sometimes binary data.

We are surprised that many ETL tools do not handle files with unformatted binary data. We advise our clients with unformatted binary data to ask the ETL tool vendors how they propose to handle that data.

We ran into this problem on a recent client. We created a tool framework where the tool can watch a directory for arriving files. When the size of the file has not changed for N seconds it assumes the file transfer is complete. It then picks up the file and passes it to a decoding routine. The specific decoding routine called is defined by a parameter to the program. The decoded file is then written to a second directory for further processing and the original file is zipped and moved to an archive for later deletion at the discretion of the ETL architect.

Therefore, for any 'different' binary unformatted file all we need to write is the C++ routine to read segments of the file and decode it according to the internal format of the file. This can usually be achieved in a day or two.

OF05: How do you support fixed format files coming into the ETL tool?

When fixed format data with no header record is coming into an ETL tool the usual way that it is handled is that the tool has some form of 'file definition editor' and the ETL developer types in the field names and field lengths into the file definition editor. However, we have seen some tools that do not even have this. We advise our clients to ask how fixed format files are moved into the ETL tool.

SeETL provides a 'Fixed Format File Reformat Utility'. The ETL Architect develops a 'heading row definition' (or gets one generated from the target table the file is going to be loaded into) defining the field names and field length. (Defining field names is optional. They can be field01, field02 etc).

The Fixed Format File Reformat Utility then reads the header record and the fixed format file and reformats the file into the self describing internal file format of SeETL. Further, such items as the value to be interpreted as null can be specified as a parameter. This is required because fixed format files and delimited files do not inherently support nulls. When reformatting fixed format files (or delimited files) the data can be reformatted based on column name or column position. This selection is provided by parameter. This is why field names are optional in the reformatting process if the columns are in the same order in which they will be loaded into a staging table.

OF06: How do you support delimited files coming into the ETL tool?

When a delimited file with a heading row is coming into the ETL tool the ETL tool generally has a wizard/utility to read the file and guess at data types. Even Microsoft Access has such a wizard.

We would be surprised to see an ETL tool that does not do this, but we still advise our clients to check.

SeETL provides a 'Delimiter Separated Values File Reformat Utility'. The utility is generalised in that the separator can by any ascii character not just commas, tabs, pipes etc.

The ETL Architect develops a staging table which has the same columns and the input file. He/she uses this table as the 'reference' for the 'heading row definition' to define the field names and field length. It is also possible to just move by column position. The file is then reformatted into the self describing internal file format of SeETL by the utility according to the staging table.

We recommend all files are reformatted into the self describing internal file format of SeETL before any further processing inside the ETL system.

Other Features to Evaluate

OF07: How do you handle newlines inside data fields?

Newlines inside data fields like addresses are quite frustrating. The extract process seems to work fine but the load processes fail. Most ETL tools have the ability to detect and translate newlines as part of the extraction process. We advise our clients to check. Sometimes it can be quite difficult to put this translation into the ETL tool and often it must be placed into the extract process at field level which means you must know which fields might contain newlines!!

SeETL provides the ability to translate the newline to any other ascii character as the data is extracted and placed into the self describing internal file format of SeETL.

This feature is provided at file level. So if a file might contain newlines inside data you can force the translation of newlines for any fields in the file.

The Data Transfer Utility can then convert that character back to a newline when the data is being placed inside the database.

OF08: How do you support truncation of leading and trailing blanks?

You might find this hard to believe, but some ETL tools do not inherently and easily provide support for removing leading and trailing blanks!!!

Indeed on one project we had to write a program to edit the unloaded XML representation of all jobs to include the removal of leading and trailing blanks from fields coming into the ETL tool!!!

We advise our clients to check how the ETL tool might support removing trailing or leading blanks because they can be a real problem inside the data warehouse.

We advise our clients that the only leading or trailing blanks that should be included in the data warehouse are the ones the EDW Architect has decided should be there. There should be no leading or trailing blanks left in the database 'by accident'.

SeETL provides a 'Data Correction Utility' which can remove leading/trailing blanks. This removal of leading and trailing blanks is performed at file level. We recommend that leading and trailing blanks are removed prior to the data being loaded into the staging area.

Further, all calls to the internal ODBC class of SeETL removes trailing blanks by default.

OF09: How do you handle delimiters inside the data itself?

Finding the delimiter that you would like to use inside the data can be something of a 'pain'. The classic nowdays is the '~' character. We have used '~' for years and years because it so rarely occurs in data. However, with the advent of WAP CDRs we see these characters in web addresses.

We advise our clients to ask their vendors how they handle receiving fixed format files or delimited files where the delimiter the vendor might like to use is inside the data itself.

We advise our clients that it is not enough to just 'use another delimiter'. There must be a way of guaranteeing that the delimiter being used in flat files does not affect processing.

SeETL provides the ability to translate any specific character found in a file to another character. Therefore, if we still want to use the '~' character as a delimiter for a WAP CDR file we can force the translation of '~' to something else such as '¬' and then use '~' as the delimiter. In this way we can guarantee that if the delimiter occurs inside the data it does not affect processing.

Other Features to Evaluate

OF10: Show us how you do data linearage.

A lot of the ETL vendors make a lot of claims about how the tool can provide ‘data linearage’ and can show you a full trace of data right from the source to the target. Though true in most cases, this is much more difficult to implement than it might first appear. We advise our clients to actually see real jobs implemented in the tool and to be shown how data linearage can be read and understood.

In our experience, to be able to get good data linearage information from the metadata store of some ELT tools requires:

1. Learning the data model of the metadata.
2. Strongly enforcing standards in jobs/mappings so that name changes in field names from source to target can be detected in the metadata store.

These limitations are because the job/mapping is encoded into some sort of relational model that can be quite complex.

SeETL provides a mapping spreadsheet. Certainly not the worlds most sophisticated meta data store!! But every DW project has a mapping spreadsheet somewhere (or word document).

In the mapping spreadsheet you can see the data linearage right there on the line of the spreadsheet. The source of the data must be expressed in a cell and the presentation view it comes out of at the other end is on the same line!

Because the spreadsheet is an XML document you can ‘report/analyse’ it any way you would like. We look forward to seeing how some people ‘analyse’ the metadata in the mapping spreadsheet.

Because the mapping spreadsheet is ‘everywhere’ our view of the future is to place more and more metadata into the mapping spreadsheet so that it grows to become the one and only place for gathering static metadata. It does not get any simpler than that.

Is this the equal of other much more complex metadata solutions? No.

Does it work? Yes!!!

OF11: How difficult is it to do the difficult jobs/mappings?

When demonstrated all the ETL tools ‘look simple’. This is because what is demonstrated are ‘simple jobs/mappings’. But an Enterprise Data Warehouse is not entirely constructed from ‘simple jobs/mappings’. We advise our clients to find out how hard is it to build the difficult jobs. You might like to ask your vendor to show you some example jobs that do things like:

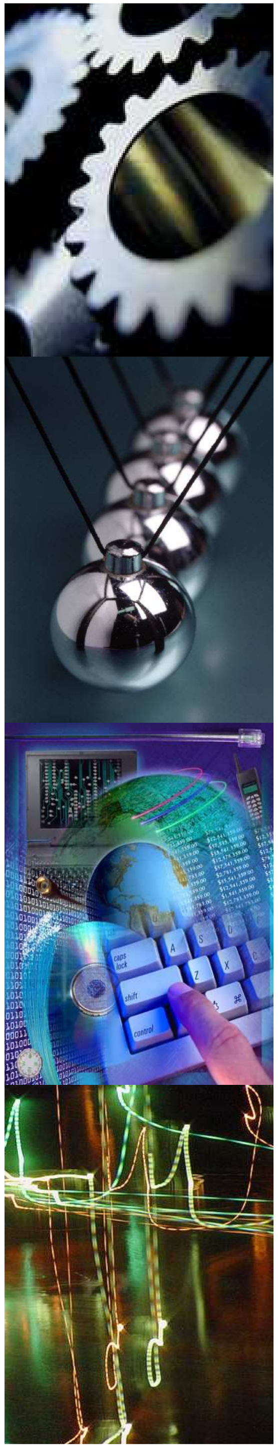
- Incrementally update a summary of transactions.
- Merge 4 or 5 sources of customer information together to build one Type 2 customer dimension.
- Check and perform updates to large fact tables.
- Bad record error handling.

We have found that often times the vendors are unable to provide examples of even the simplest jobs. We advise our clients that if they are considering making such a large investment as an ETL tool then their staff should see and use real world examples of how the tool works in complex situations.

SeETL :

- Inherently supports incremental updates of summaries.
- Uses a staging area, views, and automatically detected existing records in target dimension tables to enable merging of many disparate data sources. And if you want to use other tools you can simply call them from within the Scheduler.
- Inherently understands the primary keys of tables and can perform insert/update, update/insert, insert/delete/insert as well as delete/load to update large fact tables.
- Uses the principle of filtering data at the staging area for significant errors and zero keys plus placing the real key on the row for lookup or data integration errors.

We make our full function software available along with a small test database as a download. You are free to ‘try it out’ for as long as you feel you need to in order to make a decision for/against our product.



Degree of Control of ETL Architect

Public Information
Version 3.1: 1/1/2012

Degree of Control of ETL Architect

instant business intelligence

A

White Paper

by

Instant

Business

Intelligence

www.instantbi.com

Degree of Control of ETL Architect

We have included this section to note some particular areas where we feel the ETL Architect should be able to have greater rather than less control.

DC01: Explain how large batches of dependent groups of processing is managed.

When implementing a large Data Warehouse one of the areas where you should pay particular attention is the total elapsed time of the batch processing. Delays in batch processing can cause delays in availability of the data warehouse.

Consider the diagram to the right. Many files will come into the data warehouse and you will have many CPUs in your DW machine. The trick is to get all the CPUs working all of the time.

It must be possible to split the processing of input files into ‘Process Groups’ that can be processed in parallel while also observing any dependencies inside individual process groups.

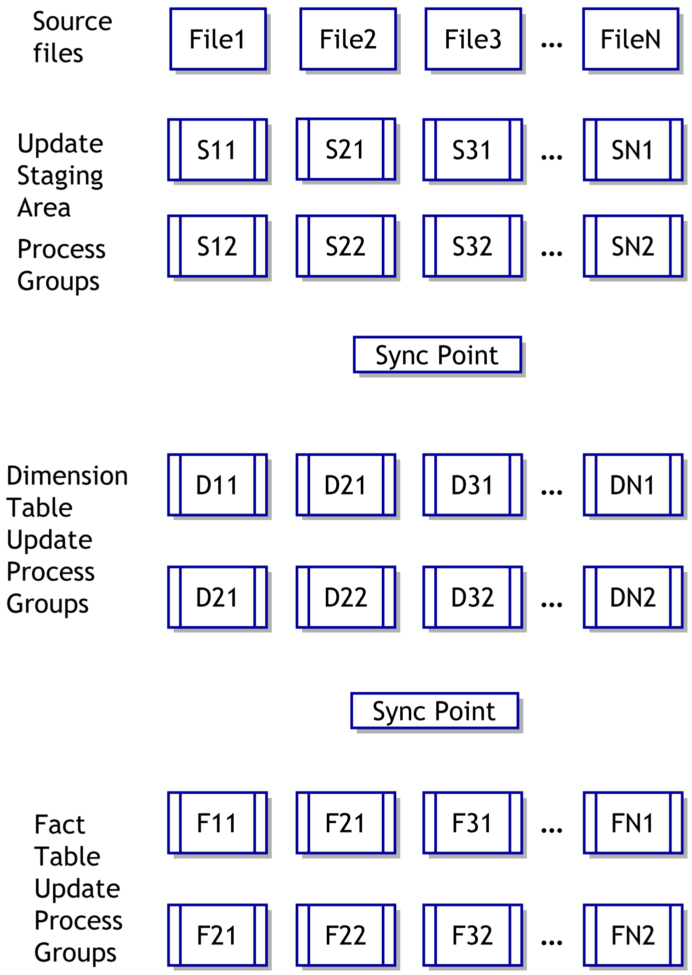
For example, in the diagram S11, S12 are a Process Group where S11 must be executed before S12. S21 and S22 form another process group but they may be run at the same time as S11 and S12 and have no dependency on S11 and S12. However, all the SNN processes must complete before any DNN processes are executed.

The number of process groups should be at the control of the ETL Architect so that he/she can control the batch such that the machine does not thrash with too many concurrent processes.

After all the files are loaded into the staging area there is typically a ‘sync point’ to make sure all staging and QA processes are successful before loading data into the DW.

At this point all Dimension table processing can run and the ETL architect may want to manage them as groups of processes as well as observe any dependencies between dimension table updates.

Again, there will be another sync point prior to fact table processing and fact tables may be processed in groups with dependencies observed.→



Source files	File1	File2	File3	...	FileN
Update Staging Area	S11	S21	S31	...	SN1
Process Groups	S12	S22	S32	...	SN2
			Sync Point		
Dimension Table Update	D11	D21	D31	...	DN1
Process Groups	D21	D22	D32	...	DN2
			Sync Point		
Fact Table Update	F11	F21	F31	...	FN1
Process Groups	F21	F22	F32	...	FN2

instant BUSINESS INTELLIGENCE

Degree of Control of ETL Architect

DC01: Explain how large batches of dependent groups of processing is managed. (contd)

To achieve maximum throughput for the machine for the data to be processed the ETL architect must be able to build batches of jobs and process groups such that the absolute minimum time is spent 'waiting' for 'late running processes' to finish.

The problem with having 'lots of slow CPUs' is that one slow process can cause significant 'wait' time for all the other processes.

The ETL Designer needs ways to measure the elapsed times for all processes and must be able to easily move processes between process groups to get the closest elapsed time balance as possible across all process groups between sync points or 'wait' points.

Or, even better, the ETL designer might like to have such fine control that he/she can break up the overall batch such that there are many sync points for different areas of data flowing into the DW.

We advise our clients to ask their ETL tool vendors to show them how this is done in their tools. In many cases, to achieve this kind of control requires that some significant amount of code be written or a third party scheduler be used.

We advise our clients not to accept the vendor statement 'We have a full function scheduler and any dependencies you want can be built in.' Though this is true for many ETL tools it is often much harder than one would imagine.

SeETL comes with a full function scheduler. The code is public, not just open source. So you can read exactly what it can do.

We have developed the scheduler based on our extensive experience in extremely large systems to be able to maximise the throughput of very large and complex batches.

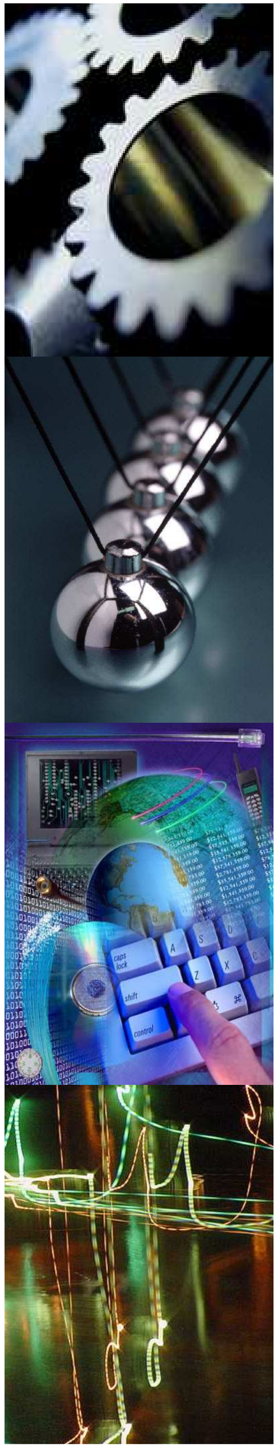
We have included an example schedule on our downloads page which shows how sophisticated a schedule can be. With SeETL you simply put the schedule into the mapping spreadsheet and SeETL will put it into the database for you.

The scheduler gives the ETL architect complete control over exactly how the batch will run. It includes extremely complex dependency checking as well as the ability to continue processing partially failed batches, only stopping the portions of the batch that have a dependency on the failed portion of the batch. It then allows you to restart portions of partially failed batches or restart full batches.

The scheduler allows a batch to start based on the normal variety of conditions such as a signal file, daily, weekly, monthly, day of month, first of month etc. New conditions can be easily added as desired because the source code is public.

We have found that SeETL can process the same workload as some ETL tools in 20-40% less time because of the greater degree of control the ETL Designer can exert over the processing of the individual jobs within the process groups and batches.

Simply put, with greater control exerted more easily the ETL Designer can keep more CPUs more busy for more of the time during batch processing without the machine thrashing from too many jobs running at once.



Ability to Add New Features

Adding New Features

Adding new features to the ETL tool is important. If you want to do something that the tool cannot do you need to go ‘outside’ of the tool and that introduces complexities such as ETL developers needing to know the environment that is ‘outside the tool’.

Further, the time lag between needing a feature and the feature being available can be significant.

Lastly, but certainly not least, because ETL tool vendors have thousands of customers and for many the ETL tool is only a tiny portion of their revenue (Microsoft, Oracle, IBM, Business Objects, Cognos), features you ask for may or may not ever be included in the tool.

AF01: What is the process of getting new features added to the tool?

We advise our clients to ask this question of the vendor.

At Instant Business Intelligence we perform custom development of new features on an as requested basis and we can usually have them done in a few weeks. Sometimes as quickly as a week.

If the new feature/function will not be included in future releases of SeETL we charge consulting rates for our time to develop the feature. If the feature will be included in future releases of SeETL we do not charge for our time. It is billed to ‘maintenance’.

Source code customers are free to add new features to their copy of the product. Given that new features are usually new programs this is simple to do. Since the language is C++ and the class library delivered with SeETL it is easy to develop new utilities.

AF02: How are new features prioritised?

We advise our clients to ask this question of the vendor.

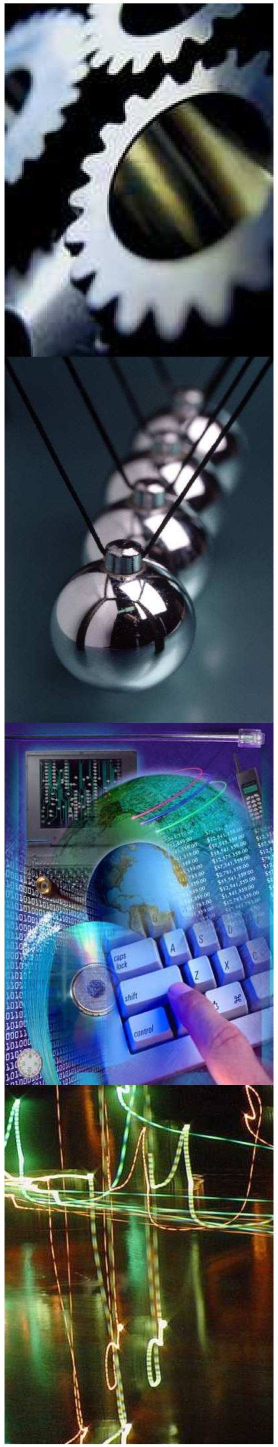
At Instant Business Intelligence we prioritise new feature development based on the value we perceive the feature has to our clients. We often ask our clients about new features suggested by other clients. Our development priorities are driven directly from our client base.

AF03: How do we extend the functionality of the software ourselves?

We advise our clients to ask this question of the vendor.

SeETL is written in C++/ODBC. Any source code client can upgrade any program or develop new programs to add new features.

In practice this has been rare. In practice our clients have asked us for new features and we have written the code for them and sent it back to them. They can read the code and also perform testing for themselves to ensure the new feature works properly.



Support, Availability of Skills and User Community

Support, Availability of Skills and User Community

SU01: What are the levels of support you offer?

We advise our clients to ask this question of the vendor.

At Instant Business Intelligence we provide global web based support through our forums on www.instantbi.com. We operate on a Monday to Friday, 9-5 working day based in Ireland. However, we frequently check the forums outside of the normal working day. We also check the support@instantbi.com email address on a regular basis.

We have resellers in the USA and Asia Pacific who also keep an eye on the forums.

We do not provide 24x7 telephone support. This is very expensive. (We prefer to build reliable software to taking support calls!!)

With the latest release of SeETL we have had clients installed for over 2 years with 5 major releases across that time and we are yet to see our first major production problem.

Our clients who are consulting companies or software companies generally provide their own code level support and only need to contact Instant Business Intelligence when they do not understand the code. This is rare as the code is very easy to understand and very well internally documented.

We do not maintain machines loaded with Solaris and AIX at our Dublin development centre. Large companies using our unix source code version are expected to provide their own code level support. It is one way the price of the software is kept to a minimum.

SU02: How available are people with skills in the ETL tool?

We advise our clients to ask this question of the vendor.

SeETL does not introduce any ‘new’ interfaces or ‘new’ environments. The only ‘new’ piece is the processing performed by the programs. There is very little to ‘learn’ directly associated with the tool. Any ETL Architect of any real ability can pick up and use SeETL by looking at the example mapping spreadsheet, browsing the manual and reading through the examples provided.
Further, no ‘ETL Developers’ are required when using SeETL. We have made the ‘ETL Developer’ redundant. Thus ‘skills in the ETL tool’ are readily available.
We believe that the very fact that some of the vendor ETL tools are suffering a ‘shortage’ of people ‘skilled’ in the tool are clear indicators that:

- The tools are not as easy to use as might be claimed.
- The tools are not as productive as might be claimed.

SU03: What User Community exists?

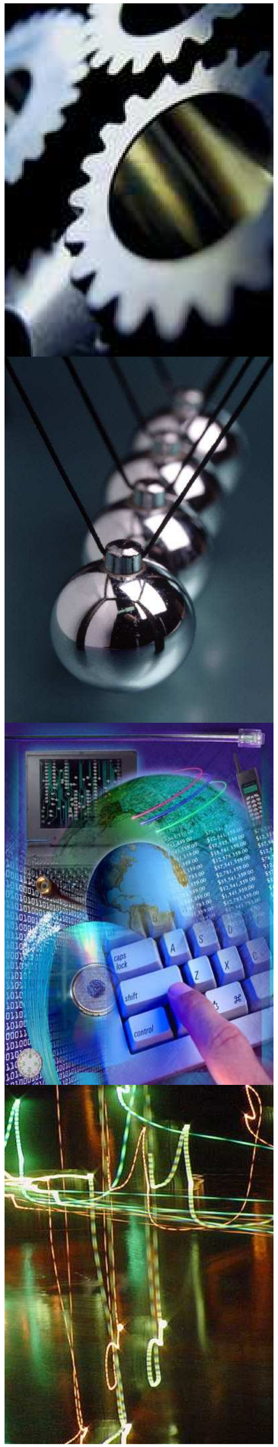
We advise our clients to ask this question of the vendor.

Our main SeETL community is our forum on www.instantbi.com.

Within our forums there are various groups depending on what products the client uses. There are forums for our free products for free support and there are also ‘fee’ based forums for clients who want priority support for free products.

For clients who own licenses for our software we have separate forums.

Any person interested in evaluating or using our products is welcome to join our user community.



Thank You For Your Time!