

The SeETL

Evaluation Guide

Version 3.1.00
1st January 2012
Author: Instant Business Intelligence
info@instantBI.com

Table of Contents

1.	CHANGE CONTROL LOG	3
2.	AUDIENCE	4
3.	INTRODUCTION	5
4.	RESTRICTIONS ON THE EVALUATION VERSION.....	6
5.	WHAT IS IN THE EVALUATION PACKAGE?	7
6.	INSTALLING THE EVALUATION PACKAGE.....	8
6.1.	Step 1 – Unzip the seetldemofiles	8
6.2.	Step 2 – Install the SeETL Executables	8
6.3.	Step 3 – Create the SEETL3000 Database, Tables and Views.....	9
6.3.1.	Creating the SEETL3000 Database	9
6.3.2.	Creating DBA User Id	9
6.3.3.	Create the Tables	10
6.3.4.	Create the Views	12
6.4.	Create and ODBC Data Source.....	12
6.5.	Load the Test Data	13
6.6.	Run SeETL Runtime.....	15
7.	EVALUATING SEETL RUNTIME	16

1. CHANGE CONTROL LOG

#	Date	Name	Description
1.0	20/1/2003	IBI Developer	Initial version for publication with the Instant Data Warehouse.
1.1	18/5/2003	IBI Developer	The executables for the Instant Data Warehouse were changed from being in a 'setup' file to just being the executables in winzip. The reason for this being that the download file is much smaller when not loading the Microsoft CLR.
1.2	8/02/2004	IBI Developer	Update email addresses.
1.3	25/03/2004	IBI Developer	Updates from feedback from evaluators.
2.0	1/1/2005	IBI Developer	Convert the document across to Instant Business Intelligence.
3.0	1/7/2006	IBI Developer	Convert the document for SeETL rebranding.
3.1.00	1/01/2012	IBI Developer	Release of version 3.1.00

2. AUDIENCE

The intended audience for the SeETL Evaluation Guide is:

- Technical developers evaluating SeETL for use within their organisation.
- IT managers considering the purchase of SeETL.

3. INTRODUCTION

Firstly, thank you for taking the time to evaluate SeETL!!!

If you have been developing star schema data warehouses for any length of time we are sure you will be amazed at the magic of building multi-level dimension tables and multi-level summary tables without having to write some form of 'mapping' code to map the sources to the target.

We have been working in IT for many years. SeETL is one of the 'cleverest' pieces of code we have even seen in all those years.

One of my biggest 'complaints' in the data warehousing space has been how hard it is to build processing that is so 'standard' when building large sophisticated data warehouses. As is usual, if we wanted anything done about it, we had to do it ourselves.

Now, with SeETL, a significant breakthrough has been achieved. With the magic of C++ and ODBC it has become possible to build SeETL such that it requires almost no parameters and no generated code. The same programs, using a minimal amount of information, can perform the vast majority of processing required to load data from a staging area into large star schema data warehouses.

This guide is intended to show you how to do this on the demonstration database so that you can evaluate SeETL for your company.

We are sure you will find the productivity benefits over any of the leading ETL tools or hand coding justify purchase of SeETL. We are sure because we have been building star schema databases for 15+ years and SeETL is by far the fastest development and maintenance environment available today.

Best Regards

The Team at Instant Business Intelligence

4. **RESTRICTIONS ON THE EVALUATION VERSION**

This section documents the restrictions placed into the executable code in the evaluation version. You need to be aware of these restrictions so that you do not try to use SeETL in a such way that it will fail 'mysteriously'.

1. The current evaluation version runs against SQL Server 2005, Oracle 10/11, Sybase IQ 12.5, DB2 UDB 8.
2. The database that you create for the demonstration database must be called SEETL3000.
3. The owner of the tables you create must be called dbo.
4. It is highly recommended the user id used to perform processing is called DBA and the password is set to 'password'.
5. There is a timer in the evaluation version to stop it running at the end of the evaluation period. You will be told the end of the evaluation period when you are given the password for the downloadable executables. In general the evaluation period is 4-8 weeks because the evaluation version is set to expire at the end of the 'next month' after you download the evaluation version.

If you try to use a different database name or owner name you should receive a message telling you that the input table could not be validated and the table will be named.

If you try to run SeETL^{RT} after the expiry period you will receive a message telling you that the evaluation period has expired and that you should contact support@instantBI.com to discuss product purchase or a new evaluation copy of the product.

5. WHAT IS IN THE EVALUATION PACKAGE?

Please make sure you have all the following zip files.

Name	Description
SeETL – User Guide	This is the detailed user guide for SeETL. For any information that you require in more detail than this Evaluation Guide please refer to the User Guide.
SeETL – User Guide - Appendicies	This document contains such useful information as the error messages and notes on the changes to the Instant Data Warehouse between releases.
SeETLDemoFiles.zip	These are the files you require to set up the demonstration database. The files contain all the DDL for the database, tables, and views. They also contain test data, commands to load the test data, and commands to run the SeETL ^{RT} on the test data and demonstration database.
SeETLRunTimeEval.zip	These are the executable programs that you must download and place into a directory that is visible in your path.
SeETLSetup.zip	This is the windows installer package that you run to set up SeETL ^{DT} .
SeETL Evaluation Guide	You must have this document because you are reading it.

6. INSTALLING THE EVALUATION PACKAGE

This section documents how to install the evaluation package.

6.1. Step 1 – Unzip the SeETLDemoFiles.zip file

Unzip this file with the option to create sub directories. It is recommended that you unzip the file to a directory called D:\IBISoftware\SeETL\DesignTime\3.1.00. If you do this you will not need to make changes to the files named in the command files to run SeETL^{RT}.

The major directory created will be D:\IBISoftware\SeETL\DesignTime\3.1.00\ . This is referred to as the seetlpath in the documentation below.

Check you have the following directory structure:

Name	Description
Seetlpath\command	Contains all the commands required to load test data and run SeETL ^{RT} against the demonstration database. It is recommended that you place the SeETL ^{RT} executables in this directory when you install SeETL ^{RT} .
Seetlpath\ddl\filegroup	Contains DDL to create the SEETL3000 database.
Seetlpath\ddl\tables	Contains the DDL to create the SEETL3000 tables.
Seetlpath\ddl\views	Contains the DDL to create the SEETL3000 views.
Seetlpath\documentation	Contains any additional documentation that has been recently developed.
Seetlpath\sql	Contains sql commands to select * all from all tables so you can see the test data has loaded and a delete all command so you can delete all data and reload it as you are running multiple tests.
Seetlpath\testdata	Contains test data that can be loaded into the SEETL3000 database by the loadalldemo command in the command directory. This is also the data directory for SeETL ^{RT} as specified in parameters in the command directory. It also contains an MS Access database which is attached to all the tables in SeETL ^{RT} as long as you use the ODBC Data Source Name of SEETL3000. This the Data Source Name that the tables are attached to.

6.2. Step 2 – Install the SeETL Executables

Unzip the SeETLRunTimeEval.zip file into any empty temporary directory. Copy the executable programs to a directory that is visible to the path.

6.3. Step 3 – Create the SEETL3000 Database, Tables and Views

It is recommended you update and run the standard DDL to create the database, tables and views in the first instance. As you become more familiar with SeETL^{RT} you may want to reconfigure the database to determine the effect of indexes or new columns or any other changes you might like to make to see how SeETL^{RT} responds to database changes. The backup copy of the database has been included in the SeETLDemoFiles.zip file so you may wish to just connect it to your SQL Server.

Please note. Demonstration databases are also available in Oracle, Sybase IQ and DB2 UDB. However, we have not included detailed documentation on each of these databases. Please contact us on support@instantBI.com if you would like a sample database in Oracle, Sybase IQ or DB2UDB.

6.3.1. Creating the SEETL3000 Database

The create database statement is as supplied below. You should change the filename parameter to point to your SQL Server data directory.

```
CREATE DATABASE SEETL3000

ON

PRIMARY
(
    NAME           = SEETL3000,
    FILENAME       = 'D:\SQLSERVER\SQLDATA\SEETL3000.mdf',
    SIZE           = 20MB,
    MAXSIZE        = unlimited,
    FILEGROWTH     = 20)

LOG ON
(
    NAME           = ' SEETL3000_log',
    FILENAME       = 'D:\SQLSERVER\SQLDATA\SEETL3000_LOG1.ldf',
    SIZE           = 20MB,
    MAXSIZE        = unlimited,
    FILEGROWTH     = 20 )
```

6.3.2. Creating DBA User Id

Go into the SQL Server Enterprise Manager and:

- Create a userid called 'dba'. Give it the password of 'password' and make it the owner of SEETL3000.

We use a dba userid because we prefer to execute jobs under a different userid than the userid that owns the tables. This way the same userid can be used to execute jobs that access tables with many owners.

6.3.3. Create the Tables

Rather than print all the DDL for all the tables we will just list them here. You need to run the DDL for each of these tables under SQL Query Analyser or similar tool. The details of each table required by the Instant Data Warehouse are contained in the User Guide.

There are a number of files containing ddl for tables and different versions for different databases. For SQL Server they are:

- M01 - seetl_control_tables sql server V3.1.00.sql
To create the SeETL control tables
- M03 - seetl_demo_tables_and_views sql server V3.1.00.sql
To create the demonstration tables.

All tables are fully described in the User Guide.

Name	Description
ctl_last_key_used	Records the 'last key used' for each dimension.
ctl_aggregation_control	Defines levels of summaries to be created.
ctl_dim_table_key_definitions	Definitions of the concatenated keys for dimension tables.
ctl_dim_table_load_control	Dimension table load control for attribution. Not used in the evaluation version of the software.
ctl_dim_table_type2_col_defs	Definitions of the fields to check for type 2 field changes for type 2 dimensions.
ctl_audit_table	The audit table. This table records the number of rows read/written from/to every file/table touched by the Instant Data Warehouse.
ctl_batch_control	The batch control table to tell the Instant Data Warehouse the date of the current batch, the batch number, and batch status.
ctl_message_table	The message table. If you choose to write error messages to the database (rather than cerr or an output file) they go into this table. Note that the messages table logs the userid and password used to log into the database so this table should only be visible to the dba and dwh userids in production.
ctl_month_control	The month control table. For the evaluation version it records the default high value date to place on the type 2 customer dimension in the date to field.
ctl_batch_pre_req	Batch processing pre-requisite table.
ctl_batch_run_log	The batch run log.
ctl_codes_lookup	A codes lookup table. Empty in the evaluation version.
ctl_column_defs	The column definitions table for maintaining metadata about columns that is derived from the database catalog via ODBC calls.
ctl_commands	The batch scheduler command table.
ctl_datastage_parms	The DataStage parameter table/
ctl_ddl_gen_columns	A table to store column definitions that a utility can generate tables from.
ctl_ddl_gen_table	A table to store definitions for the generation of tables.
ctl_file_cycle_control	A file cycle control table. Empty in the evaluation version.
ctl_mapping_ss	The table to store all details from the SeETL mappings Spreadsheet.
ctl_proc_grp_pre_req	Process group pre-requisite table.
ctl_proc_grp_run_log	Process group run log table.
ctl_proc_run_log	Process run log table.
ctl_process_commands	Process commands table.
ctl_data_types	A table that defines the various ODBC data types to make it a little easier to know what the numeric values of the data types mean.
ctl_sequence_nums	A table that contains sequence numbers for fact tables.
ctl_src_file_defs	A table that captures all the source file details from the SeETL spreadsheet.
ctl_mapping_doc_infa_01	A table to record Informatica mappings from the SeETL spreadsheet.
ctl_mapping_doc_infa_02	A table to record Informatica mappings from the SeETL spreadsheet.
address_dim	Address Dimension table.
cust_address_asoc	Customer Address Association table.

customer_dim	Customer Dimension table.
customer_dim2	Customer Dimension as a type 2 dimension table.
f_order_fact	The order fact table. The detailed level fact table for order transactions.
f_order_fact_summary	The order fact summary table. This is a multi-level summary fact table. This is where all the order fact summaries are stored.
f_order_fact_swk1/2	Work tables for the order fact summary table.
f_order_fact_v1_summary	A second summary fact table for demonstration purposes.
f_order_fact_v1_swk1/2	A second set of summary fact table sort work tables for demonstration purposes.
in_customer_address	An input table for customer addresses to demonstrate the association tables.
in_customer_dim	Input table for customer dimension data.
in_customer_dim2	Input table for customer dimension data for the type 2 customer dimension. I created separate inputs so that you can manipulate data to perform different types of testing.
input_order_facts	Input table for order facts. You can manipulate the data in this table to produce more order records or to produce different order records.
product_dim	The product dimension. If you want to add more products you can insert rows into this table.
time_dim	The time dimension. If you want to increase the number of order transactions and extend the time over which they are submitted you can insert more rows in this table.
vendor_dim	The vendor dimension. If you want to add more products you can insert rows into this table.

6.3.4. Create the Views

Rather than print all the DDL for all the views we will just list them here. You need to run the DDL for each of these views under SQL Query Analyser or similar tool. The details of each view required by the Instant Data Warehouse are contained in the User Guide.

Name	Description
address_dim_at	Address dimension table attribution view.
address_dim_dm	Address dimension table maintenance view.
address_in	Address dimension input data.
cust_address_asoc_ins	Customer Address Association Insert view.
cust_address_asoc_lkp1	Customer Address Association lookup view 1.
cust_address_asoc_lkp2	Customer Address Association lookup view 2.
cust_address_asoc_upd	Customer Address Association Update view.
customer_dim_at	Customer dimension table attribution view.
customer_dim_dm	Customer dimension table maintenance view.
customer_dim2_dm1	Type 2 customer dimension table maintenance view 1.
customer_dim2_dm2	Type 2 customer dimension table maintenance view 2.
product_dim_at	Product dimension table attribution view.
time_dim_at	Time dimension table attribution view.
vendor_dim_at	Vendor dimension table attribution view.

6.4. Create and ODBC Data Source

On the test server you must set up an ODBC Data Source. It is recommended you create one called SEETL3000 so that you do not need to change any example commands. It does not matter what userid or password you use in the ODBC Data Source as the userid and password is passed by SeETL^{RT}. However, if you choose to use integrated security you can just allow the user running SeETL^{RT} to connect to the database. In this way the userid and password used to access the database is not logged into the ctl message table.

It is expected that you are aware how to create an ODBC Data Source so no description is provided here.

6.5. Load the Test Data

Once you have created all the tables you can load the test data into the database.

You do this by running the loadall.cmd file in the command directory which is as follows:

```
call run001load input_order_facts
pause
call run001load time_dim
pause
call run001load in_customer_dim
pause
call run001load product_dim
pause
call run001load customer_dim
pause
call run001load vendor_dim
pause
call run001load in_customer_dim2
pause
call run001load customer_dim2
pause
call run001load f_order_fact
pause
call run001load f_order_fact_summary
pause
call run001load ctl_last_key_used
pause
call run001load ctl_dim_table_key_definitions
pause
call run001load ctl_dim_table_type2_col_defs
pause
call run001load ctl_batch_control
pause
call run001load ctl_file_cycle_control
pause
call run001load ctl_codes_lookup
pause
call run001load ctl_month_control
pause
call run001load ctl_message_table
pause
call run001load ctl_audit_table
pause
call run001load ctl_aggregation_control
pause
call run001load ctl_dim_table_load_control
pause
```

It calls the run001load.cmd command which is as follows:

```
CTLU001.exe
DBConnectionOutParameter=DSN=SEETL3000;SERVER=PETERLAP;UID=sa;PWD=password;DATABASE=SEETL3000
InCatalogName=SEETL3000 InSchemaName=dbo InTableName=%1
DataMovementOption=Load InsertUpdateOption=InsertThenDeleteInsert
WorkFileName=D:\IBISoftware\SeETL\DesignTime\3.1.00\TestData\%1.DAT2
```

Notes:

1. The %1 parameter is passed to name the table to load.
2. If you have not used the standard directory for the test data you must update the WorkFileName parameter.
3. You must update the SERVER= parameter in all commands to point to the name of the server you will be evaluating the SeETL^{RT} on.
4. The DBConnectionOutParameter is one long string that is passed to ODBC to connect to the database. You cannot put any blanks inside this string.

A number of evaluators have asked:

1. Why it is that loading the test data loads all the tables? and
2. What is this MS Access database provided with the test data.

The initial loading process loads the 'after' image of the data into the tables so that you can see what the data should look like after you have run it. The MS Access database also contains the small sampled of test data after you run SeETL^{RT}.

Once you have loaded the test data and understood the output that SeETL^{RT} produces you can delete the data from customer_dim, customer_dim2, and the fact tables and run the SeETL^{RT}. If all is installed ok when you run the SeETL^{RT} it should reproduce the results with only the keys being different for customer dimensions. You can even make the keys the same by updating the ctl_last_key_used table.

The MS Access database is just a copy of the final version of the data so that you can compare your processing results with what you should be seeing.

The idea of the test database is to provide a small sample database that demonstrates the functionality of SeETL^{RT} as well as proves that it is installed ok. After testing that everything is working ok you can create whatever tables they like in SEETL3000 and actually try it out on your own data. This way you can test the features of SeETL^{RT} on your own models with your own data. This provides you with the ability to gain a very good understanding of exactly how SeETL^{RT} works and exactly how it can process your data.

It's not very feasible to provide a sizable demonstration database in 4 different database managers as a download over the web. Also, feedback from evaluators is that they prefer the idea of testing SeETL^{RT} out on their own data.

6.6. Run SeETL Runtime

You are now ready to run SeETL^{RT}. This is done by issuing console commands or calling the console commands from within some processing job. The first time through it is recommended that you run one program at a time and determine if it ran correctly. Notice again you must change the SERVER= parameter in each command file and the directory for data if you did not use the recommended directory.

The following commands are in the commands directory.

Name	Description
runCTLBM01	Program to signal the start of the batch.
runCTLDM01	Customer Dimension table load using CTLDM01.
runCTLDM02	Type 2 Customer Dimension table load using CTLDM02.
runCTLAT01	Attribution processing.
runCTLAG01	Aggregation processing.
runCTLCL01	Consolidation processing.
f_order_fact_load	Load the order fact detail table.
f_order_fact_summary_load	Load the order fact summary table for new records.
f_order_fact_summary_update	Update the order fact summary table for changed records.
runCTLBM02	Program to signal the end of the batch.
runALL01	A single command to run all the above. However it does not check the return code is successful before running the next program.

Note that DebugLevel parameter is set to 9 in all commands which means that you will see listings of variables and procedure names when you execute these programs. If you want these to disappear set DebugLevel=0 or delete the parameter from the command. It will default to 0 and these messages will stop being issued.

7. EVALUATING SEETL RUNTIME

Once you have successfully run the demo database for SeETL^{RT} you may be interested in creating new models and making changes to tables in order to see how productive SeETL^{RT} is in creating new models and making changes to models.

You can create as many new input tables as you like. You can create as many new dimension/fact tables as you like. You could even implement some of your own existing stars and see how SeETL^{RT} handles the loading of these tables.

The evaluation version database is not 'tuned up' for large volumes. If you choose to place any significant amount of data into the tables you must remember to:

1. Place indexes on the dimension tables on both the integer key and the level column/dim_char_ky_fld combination to make lookups reasonably fast against the database.
2. Place indexes on the detailed fact table and summary fact table so that lookups and updates will be performed quickly and efficiently on these tables.

Important points to note during you evaluation are:

- No code is required to map source to target for dimension or fact tables.
- No code is changed when the underlying tables and views change. SeETL^{RT} will move columns by name. As you will be aware SeETL^{RT} does not know any difference between a view and a table.
- No code is changed when defining more levels in dimension tables.
- No code is changed to create new aggregates in the summary fact table. No new database objects are created to add a new summary.
- Only a small number of programs are required to perform this processing. Simplicity translates into reliability. Because you use the same code over and over again across all dimension and fact tables rather than writing a job for each one the opportunity for introducing 'bugs' is minimal.
- Using ODBC means you can talk to almost any source and any target database on Windows 2000 or Unix.
- Using C++ means other client connect mechanisms or other data types (eg blobs) are easy to add on demand.

If you are really interested and technically minded you can even run an ODBC trace on each program to see and understand the SQL it is actually issuing against the database to satisfy yourself that it is preparing and executing statements as you would expect. In all cases where the processing performance will be important the statements are prepared before being executed.

That's it. You should have SeETL^{RT} up and running. You should be able to make changes. You will be convinced it is far more productive than building your data warehouse however you are building it today.

We look forward to your purchase order for SeETL^{RT}.

Best Regards

The Team at Instant Business Intelligence