



Customer 360 - Clickstream Open Source Module¹

Version 2007-10-22

Installer – New!	3
Database Slice	3
ETL Slice	3
Analysis Slice	3
Dashboard Slice	3
Report Slice	4
Prerequisites	4
Installer Deployment Steps	4
Installer Highlights	5
Manual Installation Steps	6
Manual Installation Challenges	7
Database Slice Details	8
ETL Slice Details	11
Analysis Slice Details	13
Dashboard Slice Details	14
Standard Report Slice Details	14

¹This version is distributed under the GNU General Public License (GPL) license version 2 or later.

Introduction

Breadboard BI's Customer 360°, Finance, Workforce, and Supply Chain Solution Areas include the analytic content needed to measure and improve your enterprise. This document details the open source version of our Clickstream module, a part of our Customer 360° Solution Area. It serves as a high-level introduction to the system; more detailed instructions and comments are embedded directly into the Solution Slices™ (components) where they will be most helpful.

In addition to our open source Clickstream module, we also offer a more robust, professional version. Clickstream Pro includes additional and more sophisticated analysis dimensions, reports, cubes, and dashboards. The professional version also includes a page tagging application. An open source-professional comparison is available at http://www.breadboardbi.com/clickstream_comparison.html

Justification

Stand-alone, client-based applications that perform web server log analysis do not scale, they create information silos, they are not easy to use, and the free versions do not provide detailed analysis capabilities. To date, the only alternatives have been to develop a module based on expensive business intelligence (BI) tools (generally implemented by expensive consultants) or to send your data off-site for third parties (possibly your competitors) to host.

Breadboard BI is changing this paradigm by building a fully functional clickstream module using open source BI and database tools. This module provides a low-cost, quick-start toolkit for organizations seeking scalable, robust online marketing analysis.

Business Content Overview

This module allows website metrics like page views, visits, and referrals to be analyzed (sliced & diced) by the attributes and hierarchies present in the following dimensions:

- Date & Time (server request, visit, referral)
- Geographic Location (e.g., city, postal code, telephone code)
- Organization
- Referrer
- Visitor
- Web File
- Source System (Web Server, Page Tagger, etc.)

Because of the potentially large amounts of website data, this module includes three fact tables that logically represent:

- Web Page Views
- Web Site Visits
- Web Site Referrals

New dimensions or aggregations can be added to meet your unique requirements. If you add dimensions or facts to this module, then please share this content with the SourceForge project.

Component Overview

Breadboard BI Solution Slices are the modular components that change the raw web server log data into information for display in the dashboards, analysis cubes, and reports. Because many organizations already have their own ETL or reporting tools, some choose to deploy only select pieces of this module. For example, sometimes only the DDL is used; the logic from the ETL and the functionality of the cubes, reports, and dashboards are re-created using their own tools. In this case, simply ignore the sections that are not relevant to you.

This package contains the following components:

Installer - *New!*

- **clickstream_open_source_installer.jar** - A java-based installer program that makes installation of this module far easier.

Database Slice

- ***.DDL File** - Creates the database schemas for both the raw data store (RDS) and the multi-dimensional warehouse (MDW). This file represents a small subset of our enterprise relational model. We include DDL for Oracle and MySQL databases. However, minor changes will allow it to work on virtually all major and minor DBMS systems. (The ETL Slice is database agnostic.) Consider submitting any .ddl you create for other databases to the SourceForge project.

ETL Slice

- ***.KTR Files** - Define the Pentaho Data Integration (PDI) Transforms, and Maps that load web server, MaxMind GeoIP, and location files into the RDS and MDW.
- ***.KJB Files** - Define the Pentaho Data Integration (PDI) Jobs that load data from the files into the RDS and MDW.
- ***.BAT Files** - Simple batch example file used to schedule the Jobs.
- **Maxmind.jar & *.dat Files** - Turn visitor IP addresses into geographic location and organization information.

Analysis Slice

- **breadboardbi.mondrian.xml File** - Defines the Pentaho OLAP (Mondrian) multi-dimensional database schema. This file represents a small subset of our entire enterprise OLAP model.
- **Analysis *.xaction Files** - Define the MDX queries (Cubes) run against the OLAP database (in the analysis folder).

Dashboard Slice

- ***.jsp Files** - Define the structure of the dashboards, calls the appropriate .xml files.
- **Dashboard *.xml Files** - Define the dashboard widgets, calls the appropriate *.xaction files.

- **Dashboard *.xaction Files** - Defines the SQL (or MDX) queries run against the MDW or OLAP database (in the dashboard folder).

Report Slice

- **Report *.xml Files** - Define the report structure, call the appropriate *.xaction files.
- **Reporting *.xaction Files** - Defines the SQL (or MDX) queries run against the MDW or OLAP database (in the reporting folders, also a subfolder under dashboard).
- **metadata.xmi File** - Defines the reporting metadata layer for adhoc reporting.

The Analysis, Dashboard, and Report slices also include *.properties files. These files provide metadata, (e.g., Title and Description information) for the Pentaho server demo.

Installer Slice Deployment - New!

Recognizing that the installation of this powerful module is complex and hoping to increase the number of successful implementations, we have created an installer which performs most of the manual installation steps automatically.

Prerequisites

1. A working Pentaho BI Server (Pre-Configured Installation) v1.2 (or later) installation. Download the PCI from <http://www.pentaho.org>.
2. The Pentaho Data Integration (PDI) tool. (Formerly known as Kettle.) Download this from <http://www.pentaho.org>.
3. A working MySQL database with the MyISAM engine enabled. Download this from <http://www.mysql.com>. Note: this module is certified for MySQL enterprise.

Installer Deployment Steps

1. Execute the clickstream_open_source_installer.jar file to get started.
2. The first two tabs provide initial information, and display the module GNU/GPL license.
3. The third tab asks for information about your Pentaho solutions path, JBoss path, and Pentaho Data Integration path.
4. The fourth tab queries for information about your MySQL database. Your MySQL database must be running, the schema you provide must exist, and the user and password information you provide must have "create table" privileges in the given schema.
5. The fifth tab asks for your operating system. This tab is important to correctly update the hard-coded file paths with '/' or '\' characters.

If all goes well during installation, then the steps detailed in the Manual Slice Deployment section will have been completed.

Installer Highlights

- JNDI connections have been created for use by PDI and the Pentaho server using the information you provided to the installer.
- You will notice that new versions of the Spoon, Kitchen, Pan, and Carte .bat/.sh files have been added (all suffixed with _jndi). These include a small change that assists with the use of JNDI connections.
- All ETL objects have been placed in subfolders under the breadboard folder. The breadboard folder is located under the PDI path you provided to the installer.
- A folder named source_files has been created. This is where your latest web server log file (renamed access.log) should be placed.
- A geoip folder has been created directly under the PDI path you provided to the installer. This folder includes MaxMind's GeoIP® binary files (*.dat) used to provide geographic location and organization details about your visitors. These files are free sample files, visit MaxMind at <http://www.maxmind.com/> to download the latest free versions or to purchase more accurate files. Once acquired, replace the old files with the latest versions.
- The maxmind.jar file has been placed into the libext folder under your PDI path.
- All database tables have been created automatically using the information you provided to the installer.
- All analysis cubes, dashboards, and reports have been placed in a subfolder named "breadboard" under the solution path you provided to the installer. They should now be available on your server in a new Solution entitled "Breadboard BI Solution Area Modules". You should see this new solution immediately above Pentaho's "Samples" solution.
- The dashboard .jsp files have been placed on the Pentaho application server, directly under the Pentaho.war folder.
- If you have not previously placed the MySQL JDBC driver (e.g., mysql-connector-java-3.1.12-bin.jar) in the jboss\server\default\lib path, then you must do so before attempting to view your data.

Manual Slice Installation

For those brave (or poor) souls not interested or able to use the installer, this section provides information important to a manual installation. Breadboard BI content is designed for easy deployment for those with some Pentaho experience. Care has been taken to integrate it into Pentaho's BI Suite (Pre-Configured Installation).

All Slices include significant documentation built directly into the objects. However, if you find that useful information is missing, please post it to our SourceForge project page so that we can add your feedback to future releases.

Manual Installation Steps

After meeting the pre-requisites, here is what you need to deploy the solution manually (all steps are done automatically if you use the installer):

1. Create a functioning JNDI connection (named "mdw_mysql") for use by your Pentaho server and PDI. (If not using the installer, then you will need to make changes in several places, refer to Pentaho's documentation and forums for details).
2. Minor changes to the Pentaho .bat or .sh startup files may be necessary for JNDI connections to work properly. Your data source details will need to be added to the jdbc.properties file in the PDI simple_jndi folder. (The new installer performs all these tasks automatically.)
3. Create the following folders immediately under your PDI path: breadboardbi, geoip, and source_files.
4. Move the etl folder under the breadboardbi folder under your PDI path created in step 3. The "etl" folder contains the ETL transformations, maps, jobs, and sample batch files required to move data from the web server log files into the database.
5. The transformations with reusable Map steps, transformations with file input steps, Job entries, and batch files will need to be manually re-pointed to the correct file path (they now point to a Windows file path our development server). (The new installer does all this automatically.) The best approach may be to create a PDI meta data repository for object check-in.
6. Move the contents of the geoip folder under the geoip folder created in step 3. This folder includes MaxMind's GeoIP® binary files (*.dat) used to provide geographic location and organization details about your visitors. These files are free sample files, visit MaxMind at <http://www.maxmind.com/> to download the latest free versions or to purchase more accurate files. Once acquired, replace the old files with the latest versions.
7. Move the maxmind.jar file into the libext folder under your PDI path.
8. Update the JavaScript steps in the DIMENSION_WEB_VISITOR PDI transformation to the file path from step 6 that contains the .dat files. By default, the JavaScript steps point to the following Windows path: D:/geoip/databases/.
9. Move the dma_area.csv file into the source_files folder.
10. Build the database tables and insert the "missing row" values into the dimensions using the provided .ddl file (e.g., clickstream_mysql.ddl). The provided .ddl file contains the stage and module DDL needed to create the tables, constraints, indexes, and sequences for the MySQL and Oracle databases. Pay close attention to any errors that may occur during execution. Post any problems to the SourceForge site.
11. Copy the breadboard folder directly under the pentaho-solutions folder on your Pentaho server. The "breadboard" folder structure and related files include the Analysis Cube, Dashboard, and Standard Report Slices. It should be placed directly under the "<pentaho BI server home>/pentaho-solutions/" folder.

12. The .jsp files that define the remainder of the Dashboard Slices should be placed directly under the "<pentaho BI server home>/jboss/server/default/deploy/pentaho.war" folder. If you follow the instructions above, your Breadboard URL will be http://yourserver:8080/...

Manual Installation Challenges

Yes, you may have many. Try to solve problems on your own by reviewing Pentaho's documentation, project websites, this document, and the components themselves. Post any remaining questions to our SourceForge project.

After the Installation

Here are the steps that you need to start loading data into your newly installed module:

1. Two transformations should be run once to load the date dimension and DMA lookup tables. These transformations are named DIMENSION_DAY.ktr and STAGE_GEO_ISO_FIPS_DMA_DATA.ktr. If you did not use the installer, then you will need to manually point the table input step in the STAGE_GEO_ISO_FIPS_DMA_DATA.ktr transformation to the source file.
2. To start loading your web server log data, rename your Apache extended log file to access.log and place the file in the source_files folder created under the PDI path you provided to the installer. Run the JOB_WEB_SITE_VISIT.kjb job to load your data. If you did not use the installer, then you will need to manually point the table input step in the STAGE_WEB_SERVER_REQUEST.ktr transformation to the source file.
3. **Windows Only.** If you encounter a problem with the DIMENSION_WEB_VISITOR.ktr transformation, then you may need to edit the GET_ORGANIZATION_NAME and GET_GEO_LOCATION steps. Replace the '\\' path to the GeoIP binary files with '/'. For example, change C:\kettle_2_5\geoip\databases\GeoIPOrg.dat to C:/kettle_2_5/geoip/databases/GeoIPOrg.dat.
4. In the JOB_WEB_SITE_VISIT.kjb, two job-entries have been disabled: HTTP_GET_WEB_SERVER_LOG_FILE and SHELL_RUN_ARCHIVE_WEB_LOG_FILES. Reconfigure and re-enable if you would like to automatically retrieve your web server log file from a remote web server and subsequently move this log file to an archive folder.
5. Example batch files are provided under the breadboardbi folder that may be used to automatically schedule your jobs using Windows scheduler, archive files, etc. For Linux, create shell scripts that duplicate this functionality and schedule using CRON.
6. Some *.xaction files utilize the ORDER BY DESC and LIMIT function to return the top entities from a particular category, e.g., Top Countries by Number of Visits. The LIMIT function may be limited to MySQL. If this function is not available in your database, then consider replacing this query with an MDX or alternate SQL query (e.g., if using Oracle, then use the "RANK() over" function). Post your work to our SourceForge project.

More Details About the Module

For those that want to know more, read on...

Database Slice

1. The tables use "meaningless" keys (except in the case of the day dimension). The ETL transformations will automatically increment these columns. If your client prefers to use database sequences or identity/auto-increment columns, then minor ETL changes are required. DO NOT choose to use your source system keys as surrogate keys!
2. This data model has been designed with performance, multiple data sources, and heterogeneous DBMS platforms in mind. Ensure you carefully review and understand the denormalized structures before creating additional cubes, reports, and dashboards.
3. Table and column comments, column default values, primary keys, and alternate key indexes are provided for your assistance. A significant number of columns are defined with default values. This helps to support a diverse group of organizations with complex data sources. Be sure you understand the purpose of these default values before you attempt to remove or change them.
4. It is expected that the data model will need to be expanded (or contracted) to meet your specific needs. Please make every effort to propose new data elements for inclusion in our model. Our content will improve, and your customers have an easier upgrade and expansion path. Please post new data elements to the SourceForge project with your comments.

ETL Slice

1. The ETL transformation that stages the web server log data is designed to source Apache extended log files. Some changes may be required to stage data from the file format used by your web server.
2. Slowly Changing Type 1 (SCD1) and Slowly Changing Type 2 (SCD2) functionality is built into the dimension maps. As delivered, the logic for all columns has been set to update, this update setting mimics SCD1 functionality. Based on your requirements, this can be easily changed on a column-specific basis.

Even More Details About the Module

For those that want to know even more, read on...

Database Slice Details

Breadboard BI data models are modeled dimensionally with conformed (shared) dimensions and facts. Our library of standard models can be quickly and easily customized to your business, and adapted to their technical platform (e.g, MySQL, Greenplum, Oracle, SQLServer, etc). They include all the tables, indexes, constraints, sequences, defaults, and comments needed to support your BI requirements. Each model can be deployed on its own or with others to form an integrated, enterprise model.

The Breadboard BI data architecture is logically divided into two sections: the raw data store (RDS) and the multi-dimensional warehouse (MDW). Both RDS and MDW structures may differ significantly from the tables and indexes in your online transaction processing (OLTP) system.

The table and columns used in the RDS and MDW are named descriptively, in many cases taking full advantage of the standard 30 character limit afforded by many databases. Tables are generally prefixed with `STAGE_`, `DIMENSION_`, `FACT_`, OR `ADMIN_` to help describe their purpose. Columns are suffixed with one of the following self-explanatory class words: `amt` (amount), `code`, `cat` (category), `date`, `desc` (description), `id` (source key), `ind` (indicator or flag), `name`, `number`, `qty` (quantity), and `setid`.

RDS Structure

The structure of the RDS tables includes columns for staging OLTP data (in this instance web server logs) and columns used by the ETL process. A few columns used for staging OLTP data are worth special mention. The list includes:

1. Columns suffixed with `_ID` - These are the whole or a part of the composite key from the source system. These mandatory columns are used to update the data in the warehouse. To account for systems that use `VARCHAR` keys, it is mandatory that the datatypes of these columns be converted to `varchar(32)` during your data staging process.
2. `DW_DIMENSION_LOAD_IND` - (not used in this module). This character (1) column denotes if rows in the stage table have been successfully loaded into the dimension table. It implies that rows can be loaded into fact tables. This column is only found in stage tables that maintain data for both an MDW dimension and fact table. For example, the order stage table stages descriptive data for the order dimension table and metrics for the order fact table.
3. `DW_ERROR_IND` - (not used in this module). This character (1) column denotes if rows failed a previous load attempt. It can be used by the data warehouse administrator to research rows that failed validation checks in the load process.
4. `DW_LOAD_DATE` - This date datatype column defaults to the system date when rows in the staging table are loaded. Although the RDS is designed as a non-persistent data staging area, these columns can be used to develop an incremental load strategy from a persistent RDS (details in the ETL Slice Details section).
5. `DW_SOFT_DELETE_IND` - This character (1) column can be used, like the `DW_LOAD_DATE`, to implement an incremental load strategy (details in the ETL Slice Details section). Because this is generally helpful in situations with small amounts of data, it is not advisable to use this column in the web analysis context.
6. `SOURCE_SYSTEM_ID` - This numeric column identifies the source system from which the data is extracted (e.g, web server 1, web server 2, etc.). (The counterpart for the `SOURCE_SYTEM_ID` in the MDW is the `SOURCE_SYSTEM_SK`.) If you are only adding data from one web server, then this column will default to 0. If you have multiple source systems or add an additional system in the future, then assign each of your systems a unique number. See the section entitled "More About `SOURCE_SYSTEM_SK`" for additional information.

MDW Structure

The MDW tables are designed using dimensional modeling techniques. The resulting dimension and fact tables are simple for business users to understand, and they perform optimally with large data sets. Conformed dimensions like customer, day, product, and person are shared across most Solution Areas to allow for cross-organizational reporting and to prevent information stovepipes from forming between departments in a single organization.

Surrogate Primary Keys

A single surrogate primary key, or composite primary key, is created for each MDW table to act as the primary key. Through the use of PDI, this primary key is guaranteed to be unique. Primarily because source system keys (e.g., CUSTOMER_ID) are not unique across systems, the MDW does not rely on source system keys to uniquely identify its tables. The use of a single, non-composite primary key in the dimensions optimizes join performance between dimension and fact tables. Finally, the use of surrogate keys can also protect against source system key-reuse and provides the flexibility to implement slowly changing dimensions.

Alternate Keys

Aside from the assigned primary key, an alternate key (AK) is available in dimension and fact tables to allow for ETL updates. This key is composed of the actual key from the source system. In addition, another part of some AKs may be the SOURCE_SYSTEM_SK. This column may be part of the alternate primary key in order to ensure uniqueness across source systems (web servers in this module).

More About SOURCE_SYSTEM_SK

(If you are only adding data from one web server, then you can ignore this section.) As previously discussed, the SOURCE_SYSTEM_SK is part of many alternate keys in MDW dimension and transaction grain fact tables. A similarly valued identifier, named the SOURCE_SYSTEM_ID, also exists in many RDS stage tables. Both columns should be valued exactly the same, i.e., for a given source system with a SOURCE_SYSTEM_ID = 1, the SOURCE_SYSTEM_SK will also be = 1. Both the SOURCE_SYSTEM_ID and the SOURCE_SYSTEM_SK are "owned" by your analytics team - they will have no meaning outside of this context. An administrative table named ADMIN_SOURCE_SYSTEM has been created to maintain information about each source system (web server) added to the analytics solution.

If you are adding more than one web server to the module, then each web server will have its own row in the source system dimension, but you should assign the same PARENT_SOURCE_SYSTEM_SK value to both rows. However, if you are adding another duplicative data source like our page tagger application, then you must assign this row a different PARENT_SOURCE_SYSTEM_SK value. If you accidentally assign the same PARENT_SOURCE_SYSTEM_SK to a duplicative source, then you will double count your data. This is because the same page visit will be recorded in your web log and by the page tagging application. In aggregations, a separate PARENT_SOURCE_SYSTEM_SK value prevents this from occurring.

MDW ETL Columns

A few MDW columns used by ETL are worth special mention. The list includes:

1. DW_CURRENT_IND - This character (1) column allows for simple "current" reporting in a slowly changing dimension type 2 (SCD2) dimension. This column defaults to 'Y'. **It is not currently used in this module.**
2. DW_START_DATE - This timestamp datatype column represents the start date in a SCD2 dimension. It defaults to '1971-01-01 00:00:00'.
3. DW_STOP_DATE - This timestamp datatype column represents the stop date in a SCD2 dimension. It defaults to '2036-12-31 00:00:00'.
4. DW_VERSION_NUMBER - This number column represents the version of a particular dimension in a SCD2 dimension. It defaults to 1.
5. DW_LOAD_DATE - This timestamp datatype column indicates the date a row was loaded into the data warehouse. It defaults to the CURRENT_TIMESTAMP.

Final Data Model Slice Thoughts

Creating the RDS and MDW objects from the Breadboard BI DDL file should be a straight-forward process. You may need to tweak the DDL slightly to accommodate different DBMS versions, but this effort should be minimal. Add or drop indexes as needed to enhance data load and query performance, but keep in mind that some indexes may increase ETL load times (these can be dropped at ETL load time and rebuilt post-load using ETL objects). Finally, it is important to maintain referential integrity between the fact tables and the dimensions.

ETL Slice Details

Breadboard BI significantly reduces the risk of your BI project by bundling pre-built ETL maps into a customizable solution. Our ETL maps, built using the Pentaho Data Integration tool, integrate advanced ETL processes like incremental load strategies, error processing, slowly changing dimension (SCD) type 1 & type 2 updates, surrogate keys, and reusable logic. For details read the Advanced ETL with Pentaho white paper at http://www.breadboardbi.com/white_papers/pentaho_etl_whitepaper.pdf.

RDS Data Load

The RDS data load consists of three distinct Jobs:

1. Job Web Site Visit - Loads raw web server log data into the RDS and MDW on a regular schedule (e.g., every two hours). The step that defines the web log format may need to be changed to work with your format. Please share your new step with the project.
2. Job Stage MaxMind Data - Loads MaxMind GeoIP data once per month. **The latest version of Clickstream no longer uses the Mindset .csv files for complexity and performance reasons.** This job and related objects are not required if the new binary database java api is used.
3. Job Stage Location Data - Loads miscellaneous location lookup data on a one-time (or infrequent) basis. **The latest version of Clickstream no longer uses the Mindset .csv files for complexity and performance reasons.**

Refer to the Jobs and underlying Transformations and Step for details.

Certain generic points should be kept in mind when loading the RDS tables. These include:

1. If loading multiple source systems (web servers in this context) into the staging tables, assign each source system a unique number. If you are only loading one system, the SOURCE_SYSTEM_ID column will default to 0 if it is not populated by your ETL process.
2. Trim char and varchar columns.
3. Convert character values like product names from UPPER case to lower or Initial Upper Case (if desired). This may make data more "attractive" in your cubes, dashboards, and reports.

MDW Data Load

After the RDS has been successfully loaded, the MDW data load can begin. The process of loading data begins with the dimensions, and ends with the fact tables. As delivered, each transformation performs the following basic tasks:

1. **Executes a single SQL statement against one RDS table.**
2. **In dimension maps, the SCD logic is evaluated to determine if an insert or update is appropriate.**
This configurable SCD logic is implemented using the dimension lookup/update step. As delivered, the logic for all columns has been set to update - this update setting mimics SCD1 functionality. Based on your requirements, this can be easily changed for each dimension on a column-specific basis.
3. **In dimension maps, surrogate keys are created using either database sequences or identity columns.**
As delivered, the transformations utilize database sequences. The .ddl necessary to create these sequences is provided in the Database Slice. Because databases may not have sequence objects, identity columns may be a good solution. An easy change to the dimension lookup/update step in each transformation will convert it from using sequences to using identity columns. Simply erase the value in the "Optional Sequence" field and check the "use auto increment field?" box for each dimension transformation.
4. **In fact table maps, the source system keys are used to return the dimension table surrogate keys.**
Because this logic is common across fact table maps, these lookup/update steps are embedded in reusable map objects. If performance suffers, change the fact table maps to only allow inserts.
5. **Following the successful load of the MDW fact table, the Pentaho OLAP (Mondrian) cache must be flushed (ignore if Mondrian is not used).**
A PDI job with an HTTP job entry is used to call a simple .jsp file (mondrian_cache_flush.jsp). This .jsp file flushes the Mondrian OLAP cache so that newly inserted or updated data will be reflected in the Analysis Cubes.

Resolving Visitor Organizations and Locations

The latest version of Clickstream utilizes MaxMind's Java APIs and binary data files within PDI to resolve visitor organizations and locations. This approach is far more scalable than our older relational database lookup method.

The files we included are free sample files, but they are not the most accurate available. Visit MaxMind at <http://www.maxmind.com/> to download the latest free versions or to purchase more accurate files. We strongly advise users to buy the latest GeoIPOrg.dat, it is far more accurate than the sample file we have provided. Once acquired, replace the old files with the latest versions.

The Missing Row

The dimension ETL transformations will insert one "missing" row with a surrogate key valued at 0 into every dimension table. This should automatically happen the first time a transformation is run. It is best if you inserted this row into your dimension tables prior to running the job, especially if there are NOT NULL constraints on any of your non key columns. If you have these constraints in your table and you don't have the missing row inserted prior to running the transformation, then your run will fail as a result of your DBMS enforcing the constraint. The database .DDL file includes the insert statements necessary to insert these "missing rows" into your MDW tables.

Final ETL Thoughts

- Your data will be unique, and it may stress the code in new ways. Deploy each transformation one at a time, unit test all logic paths in each transformation with your data from the Spoon interface, optimize as needed.
- Consider automatically dropping indexes or enforced constraints (i.e., those not utilized by ETL) prior to loading if you need to increase performance. You can automatically rebuild and enable these after the load completes (call a database script or procedure from your Chef job).
- Don't expect to roll-out our ETL Slices and have all of the maps work perfectly during the first test run. Post any bug fixes or enhancement requests to the SourceForge project.

Analysis Slice Details

Analysis cubes offer excellent dynamic slice and dice (drill down and drill through) ability. A few well-designed cubes allow business users to dynamically explore data without the time and energy wasted in the process of requesting and waiting for IT to build large numbers of standard reports. Our cubes are built using the open source Pentaho OLAP Server (formerly Mondrian). The Breadboard BI Analysis Slice includes a Multi-Dimensional Schema and Analysis Cubes.

Multi-Dimensional Schema

The Breadboard BI Multi-Dimensional Schema consists of two logical pieces. The first part consists of the cube, dimension, hierarchy, level, measure, and member metadata that define the multi-dimensional database. The second part of this schema includes a mapping of these objects to the Breadboard BI dimensional module (MDW). The breadboard.mondrian.xml file includes the

metadata that defines this schema. View the cubes at <http://www.breadboardbi.com/demo.html>.

Analysis Cubes

The Analysis Cubes are defined by the multidimensional expression (MDX) queries found in the .xaction files. Each MDX query references objects defined in the Multi-Dimensional Schema to build exactly one OLAP cube.

Final Analysis Thoughts

- Whenever making changes to the breadboard.mondrian.xml schema definition file, always first make a backup copy. Problems in this file can be challenging to troubleshoot, so it is good to have a working copy to which you can roll-back.
- If you need to make changes to the Schema definition file, iteratively change and test. Again, a lot of changes at once can be challenging to troubleshoot.
- For Pentaho OLAP (Mondrian) information and discussions, visit Mondrian's SourceForge project at <http://mondrian.sourceforge.net>.
- Finally, please post changes you believe would enhance the system to the SourceForge project.

Dashboard Slice Details

Dashboards are an excellent way to convey critical information to employees at all levels of an organization. They can act as the "gateway" to standard reports, analysis cubes, and data mining results; or serve as standalone components with interactive, interdependent objects. In either case, they serve as the entry point for a well-designed business intelligence system. The Breadboard BI Dashboard Slice includes pre-built interactive and gateway dashboards.

Our dashboards utilize an assortment of Pentaho dashboard chart, dial, and report widgets, as well as embedded cubes to simply convey meaningful information. Each dashboard is defined by a single .jsp page; each .jsp page references one or more widgets defined in *_widget.xml files. Finally, these widget.xml files may further reference .xaction files to retrieve data from the MDW. View the dashboards at <http://www.breadboardbi.com/demo.html>.

Standard Report Slice Details

Much maligned in favor of dynamic analytic cubes, standard reports are still an important part of most BI systems. Breadboard BI bundles pre-built reports as part of a Breadboard BI Solution that we or our partners customize with you. We build all these reports using the open source Pentaho Reporting suite.

Because you probably prefer to create your own highly customized reports, we offer sample reports that display some of the more popular features, including the use of charts and report prompts (both static and derived from database queries). Use these examples to build highly customized reports. View these reports at <http://www.breadboardbi.com/demo.html>.

Additional Resources

- Pentaho's web site at <http://www.pentaho.com> for a series of very helpful .pdf files and discussion boards monitored by skilled Pentaho resources. A few good pdf files include: Pentaho Advanced Install Guide, Pentaho Creating Pentaho Solutions, Pentaho Customizing Deployments, Pentaho Quick Start, Pentaho Report Design Wizard.

Licensing

The open source version of this module (including all associated component files) is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

It is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this module; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA