

Enterprise Enabler® Map Designer Overview

www.stonebond.com

The Enterprise Enabler® (EE) Map Designer is the Graphical User Interface that easily guides a user to build interfaces between the applications. It constitutes the application interface definition capabilities of the EE suite of tools, and is designed to generate application interfaces without programming. A user does not need to be highly skilled in systems or programming in order to build interfaces. However, Stone Bond recommends the close participation of information technology personnel or consultants to ensure that the overall implementation with many corporate interfaces is managed appropriately.

Map Designer is a code-less design studio that graphically represents the process workflow while bringing rule-based intelligence in a single unified develop, test, deploy and monitor environment. To build an interface, an end user or system administrator graphically defines a template for the source data that needs to be transferred, then does the same for the destination data. Once the source and destination have been defined, a "map" is designed in a point-and-click manner to establish the relationships and transformation of the data when it is transferred. All templates can be reused as desired, and source templates can become a sanctioned source of corporate data by using EE's Source of Record Manager.

Source and destination data can be flat files (including reports), database tables, Excel, Word, XML, EDI, Web Services, electronic stream, process control data, emails, and virtually any data format that is discernible.

Transformation Engine: The EE *Transformation Engine* is the heart of EE. While EE's Map Designer can actually run the interfaces as they are built, that feature is designed only for running the data transfers on a test basis. The EE Transformation Engine is designed to handle the running of the data transfers, as instructed by EE's scheduler. Complex data transformations are effortlessly performed as defined in Map Designer.

EE's *Transformation Engine* transforms (manipulates and moves) your data from a source to a destination. It uses data transformation criteria contained in your Map that can be based on formulas, complex equations, business rules or logic, or other conditions you specify. Maps can use any combination of the transformation features described below:

- Row/column mappings (including aggregation functions)
- Unit of measure conversions
- Complex formulas (variables from other data sources)
- Business rules
- Matrix inversions
- Filters
- Constants, formulas, equations, etc.

<http://www.layer2.de/en/products/Pages/Enterprise-Enabler-for-SharePoint-BDLC.aspx#2>

Enterprise Enabler® Map Designer Overview

www.stonebond.com

Source Templates: A source template provides the ability for you to define application interfaces for any of the external applications and/or databases supported by AppComm components. Source templates contain descriptions of the source data and its format in order to extract and feed data to the Transformation Engine using your Map definitions.

Destination Templates: Destination templates can create a new data set or file (e.g. database, message, ASCII), or they can be appended to an existing data set. Conditions can be placed on the updates performed using rules or equations (e.g. update destination, but only if data has changed). Destination Templates are created in a manner similar to source templates. Where source templates gather information from the source application or database, destination templates transfer the data collected by the source templates (that have been processed by the transformation engine) to its target destination. AppComm components also apply to destination templates. As with source templates, you can define and create numerous destination templates to receive data, and all can be reused for any number of interfaces.

Application Communication Handlers (AppComms): AppComms chapter contains descriptive and usage information for all the *Application Communication Handlers* (AppComms) found in EE. The term "adapter" is widely used to mean a system's knowledge of how to connect and move data in and out of various software applications or data formats. Usually, they are billed as more or less "plug and play" modules; however, in reality, it is not that simple. EE has a slightly different concept for adapters (AppComms) that allow for completely GUI driven definitions using the Map Designer module.

AppComms automate the discovery of data schema for any application or data store. Once discovered it will save this as metadata which provides the ability to have a virtual connection with a real-time and on-demand access to the data to start leveraging the aggregation of data.

Unlike a connector or adaptor, which is hard coded, an AppComm is used to connect to the schema of the given system. By doing this, the AppComm can monitor the schema live, should anything change, and notify the owner or administrator of the change. With an adaptor or connector, should the schema change, then the adaptor or connector will need to be recoded.

The AppComm is built with a model driven architecture, when connecting to any backend system or applications they are connected live and discovering the schema of the objects of interested. Any customization to the objects recognized without any coding. That schema information is stored as metadata so there is never a need to compile and deploy assemblies when creating or modifying a dynamic data model.

<http://www.layer2.de/en/products/Pages/Enterprise-Enabler-for-SharePoint-BDLC.aspx#2>

Mapper Attributes: *Codelessly Creating a Map*

- **Lookup:** A relationship between the columns of different blocks in a source template can be created here. A primary block is set in the Lookup. This helps in deciding which block can be used as the driving block in the Map.
- **Equation Builder:** *Equation Editor* is used in the Map building process that allows you to define equations or complex formulas. Equations are performed during the data transformation process. When you right click on an unmapped destination column and select equation the equation editor appears.
- **Exception Handler:** This is code that will be used when an exception occurs during the execution of the map. *Equation Editor* is used to write the code.
- **Global Variables:** *Global variables* allow users to define certain variables, which can be used multiple times throughout the Map. These variables must be initialized with string or numeric values only. Users can update the values of the global variables using Equation editor.
- **Filters:** *Filters* are another way you can manipulate data from a source to its destination. Filters can be used when you want to prevent specific source data from reaching its destination. Filters are executed for every source record processed through the Transformation Engine.
- **Constants:** *Constant Values* are a way to instruct the Map Designer to insert a constant value into a field during the column mapping process.
- **Rules:** A *rule* is a function that will return a value that can be assigned to a destination column. Similar to other Map features and functionality, your business rules can be added to a Map for execution during the data transformation process.
- **Pre-built functions:** *Pre-built functions* are a list of functions, which are applied on the data by the Transformation Engine before the data is uploaded into the destination. Many common pre-built functions are available within the EE Suites, which the users can utilize for common tasks.
- **Aggregation functions:** *Aggregation Functions* are another way of customizing your interfaces during the map building process. You can create new functions, or apply any of the numerous row aggregation functions to your source data.
- **Constant Value:** *Constant Values* are a way to instruct the Map Designer to insert a constant value into a field during the column mapping process. For example, if a destination column is not linked to a source column, or perhaps the destination column needs to be populated with a value that remains constant for each row, then the Constant Value feature can be applied to your Map. In the following example, the value "10" is entered for every source row in the "Number" Column.

Features, Functions and Values:

- Application/data source agnostic data mapping
- Hierarchical to relational data mapping
- Cross application lookups
- Dynamic filtering of data in AppComms during transformation, based on lookups.
- Multilevel data filters
- Plugging Pre-built functions
- Complex business rules for data manipulations
- Ability to define Global variables for aggregation, sharing and other purposes
- Exception handling during transformations
- Skip or continue iterations based on complex business rules.
- Ability to define row aggregations
- Auto unit of measure conversions
- Pre and post processing before and after data extractions
- Complete audit trail of the transformations
- Preview and analyze data just before writing to destination application/data source
- Ability to share data between processes, maps and templates
- Control transaction context with commit and roll backs
- Multilevel looping on the source data
- Ability to use virtual store for temporary data storage
- Ability to use virtual parameters and global variables for building queries.
- Extensible framework for user defined functions, unit of measures and application communicators

<http://www.layer2.de/en/products/Pages/Enterprise-Enabler-for-SharePoint-BDLC.aspx#2>