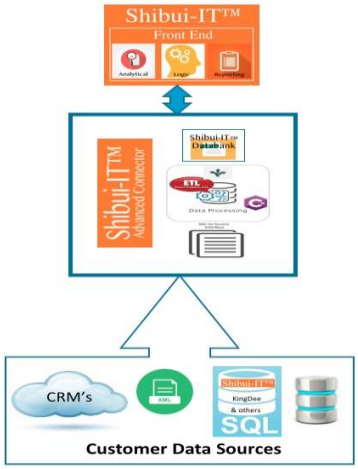
Shibui-IT™ Advanced Connector (SAC) enables easy access to any data sources

Abstract:

SAC is an Extraction-Transformation-Loading (ETL) tool which has been developed by kdm semi consulting GmbH (k•d•m) to load Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) data from multiple and different resources into the Shibui-IT™ databank. Transact-SQL logic is used to integrate this data into Shibui-IT™ Business Intelligence (BI) reports. The SAC architectural model consists of an (a) modularly extensible SAC to Source interface setting approach; (b) a SAC processing logic which uses a decision/logic-tree model and associated operations to support the target application requirements; and, (c) a configurable modular SAC CSV data repository. To support the SAC compliance diagnostic features, a data interpretation tool for debugging and monitoring of used customer data, or logging functions to report date of operations, are integrated.



kdm semi consulting GmbH
Muenchener Str. 22, Steinhoering, Germany
contact@kdmsemiconsulting.com

HRB 224819

Glossary

"Shibui-IT™" is the k•d•m trademark for an advanced analytics and forecasting Business Intelligence suite.

"CRM" stands for Customer Relationship Management and is a process and software tool which allows businesses efficient interactions with customers and a better data and information management with them.

"ETL" is short for Extract, Transform, Load, three connector functions that are combined into one Shibui-IT™ Advanced Connector tool to pull data out of one data source and shift it into the Shibui-IT™ databank or in our case a CSV repository.

"ERP" stands for Enterprise Resource Planning which integrates business processes across Sales, Finance, Operations, Supply Chain, etc into one single system supported by specific ERP software tools.

"BI" stands for Business Intelligence and is a tool to collect huge raw data and create meaningful reports, charts or analysis to better monitor a business

"Data Source" is in computer term the source where the data comes from. It can be a database, a dataset or a spreadsheet .

"CSV" is short for Comma-Separated Values and is a plain text file.

"SaaS" stands for Software as a Service and is purchased on a subscription base hence no hardware or software investment necessary.

"Transact SQL" is Microsoft's proprietary name to the SQL (Structured Query Language) used to interact with relational databases.

This paper outlines the k•d•m approach for how to identify and process various data formats, access any data source, and optimize data flow from source to Shibui-IT™ BI target applications.

Issued by: kdm semi consulting GmbH

Authors: Klaus Mueller

Status: Issued

Issue Date: 10.12.2018

Revision History

Date	Change	Author	Comments
Dec, 10th 2018	Issued	KM	

Table of Content

1 Introduction

Business data has become too complex and voluminous for executives, investors and analysts to keep track of. They run the risk of no longer being able to properly monitor, compile and analyze the most vital and relevant financial data. In today's business world financial reports lack organizational clarity, and waste important company resources that could be utilized elsewhere. More importantly, reports never seem to be handy when they are needed most.

k•d•m's Shibui-IT™ SaaS solution collects business data from any existing CRM and ERP systems, or even from simple Excel sheets. At the push of a button, Shibui-IT™ converts any data set into user-friendly and quickly accessible charts or other forms of visual representation, while maintaining an extraordinary level of detail for further analysis.

Traditional front-end tools cover the basic business support feature and offer function-standard solutions, but often lack operational knowledge to filter and report the necessary customized information from big data. For that reason, many companies often use spreadsheets and other *ad hoc* tools to get the job done. As a result, more people have to partake in the data collection and analysis process than needed. This approach is very time consuming and labor intensive, and can thus become exceedingly expensive.

The traditional ETL process of extracting data from different sources into the front-end databank can best be described as a simple moving of files, In case the target application (BI Reports) requirements change or access to a different database is needed, a rewrite of the entire ETL process logic is required.

Shibui-IT™ Advanced Connector (SAC), on the other hand, is characterized by a strict separation of source, processing logic, and target. On the source interface SAC uses a modular field definition approach which can handle various data types (String, Integer, Date, etc), and can be synced with any data sources.

The SAC-to-Source Interface specifications are determined by the Shibui-IT™ target application requirements, and programmed to onboard only those data sets which are of immediate relevance to the Shibui-IT™ BI reports. The SAC processing logic uses a decision/logic-tree model approach to make sure that the data being transferred into the SAC databank complies with Shibui-IT™ BI specifications. SAC uses a CSV data repository because CSV files a) are easy to access, b) enumerate the kinds of data being loaded, c) are stable, and d) do not cause any unwanted data manipulation.

2 SAC Architecture

The SAC architecture (see Figure 1.) consist of three major SAC building blocks to address the Shibui-IT™ data acquisition requirements.

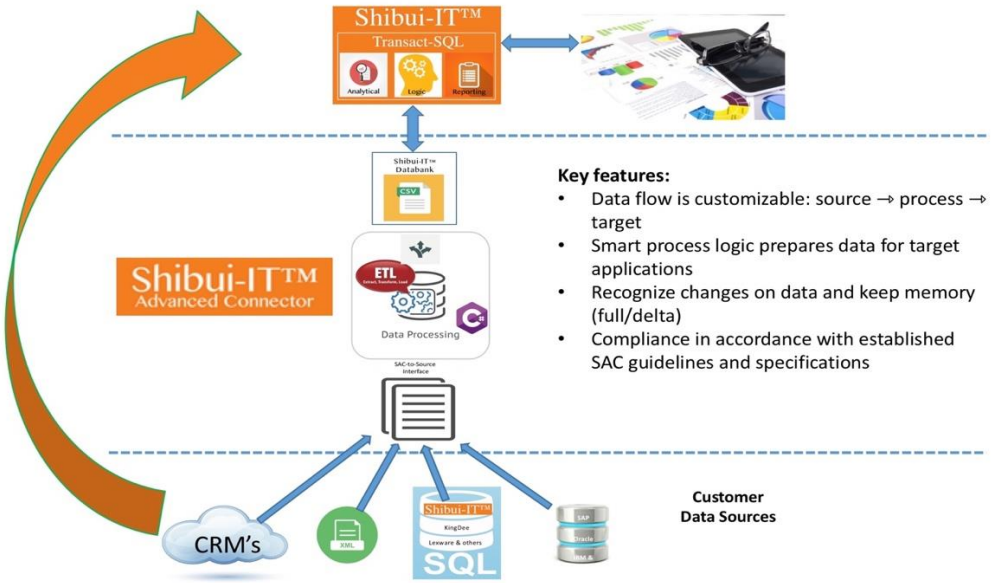


Figure 1. SAC Architecture

2.1 Interfacing SAC to Sources:

- SAC uses a modular field definition approach which can handle various data types (String, Integer, Float, Character, …) to access any popular data source
- SAC system assistant is able to configure any data source interface settings
- SAC detects any data changes in source system by supporting time variants like date, time, counter or delete-flag

2.2 SAC Data Processing Logic:

- k•d•m has chosen Microsoft for the Shibui-IT™ architecture
- SAC uses C#, a programming language developed by MS, which is designed to write the process logic for the SAC.
    - C# programming example (see below):

```
FCUCDSEHsSelection.cs*   FCUCDSHEdit.cs   FCUCDSECSrcSQLEdit.cs   FCUCDSEHEdit.cs
ShibuiAppDtaExchg                                                 KDM.ShibuiAppDtaExchg.FormCentralU
25          private bool dGVDSEHsBindSrcSortInProgress;
26
            1 reference | Johannes Buergmayr, 20 days ago | 1 author, 1 change
27          internal FCUCDSEHsSelection()
28          {
29              InitializeComponent();
30              dGVDSEHsselected = -1;
31              fCUCDisplayText = "DataSource Exchange Header Selection";
32              FCUCCreationState = FCUCCreationStateENUM.isNew;
33          }
34
            1 reference | Johannes Buergmayr, 15 days ago | 1 author, 2 changes
35          internal void FCUCInit(int SelectDSEHID, int DSHIDwp)
36          {
37              loghandler.AddLog("Information",
38                  "UC " + fCUCDisplayText + " started",
39                  "UC " + fCUCDisplayText
40                  + "for DSH "
41                  + DSHIDwp
42                  + " startetd with DSHE Selection "
43                  + SelectDSEHID,
44                  false);
45
46              DSHID = DSHIDwp;
47
48              dGVDSEHsRunUID = General.CollectionsKDM.RunUINT;
49              Settings.DSExchgDefHeadersMnt.SBLAdd(DSHID,dGVDSEHsRunUID);
50              dGVDSEHsBindSrc = new BindingSource();
51              dGVDSEHsBindSrcSortInProgress = false;
52              dGVDSEHsBindSrc.SuspendBinding();
53              dGVDSEHsBindSrc.DataSource = Settings.DSExchgDefHeadersMnt.SBLGet(DSHID, dGVDSEHsRunUID);
54              Settings.DSExchgDefHeadersMnt.SBLEventSortStartedAdd(DSHID, dGVDSEHsRunUID, SBL_SortStarted);
55
56              dataGridViewDSEHs.AutoGenerateColumns = false;
57              dataGridViewDSEHs.DataSource = dGVDSEHsBindSrc;
58              DataGridViewColumn column = new DataGridViewTextBoxColumn
59              {
60                  DataPropertyName = "ID",
61                  Name = "ID",
62                  Visible = false
```

- SAC processing logic uses about 30 different decision/logic-trees and associated operations to create the pre-defined Shibui-IT™ CSV databank tables for Sales & Purchase Orders, Customer, Products, Country, Account Manager, among others
- SAC processing logic uses a universal and modular approach to a) set up, use, and manage different customer data sources, and b) add or change data fields of the Shibui-IT™ CSV databank

2.3 Configuring SAC to Shibui-IT™ Databank

- SAC uses a CSV data repository to store the pre-defined tables needed for the various Shibui-IT™ BI reports
  - CSV table snapshot:

SOLI.csv - Notepad

File Edit Format View Help

```
KDMSHIBUI;KDMSTYPE;KDMS;KDMSL;KDMSLCGTS;KDMSLSHPD;KDMSLPRD;KDMSOCS3;KDMSLCRD;KDMSLDA;KDMSLDC;KDMSLDP;KDMSLSPK;KDMSLSPC;KDMSLQTK;KDMSLQTU
A0;IC;SO_10001;SO_LI_001;20110921010000;0;PR_0011;CU_0069;20110921;;20111102;20111102;69,64;USD;150;ST
A0;IC;SO_10001;SO_LI_001;20111102010000;1;PR_0011;CU_0069;20110921;20111102;20111102;20111102;69,64;USD;150;ST
A0;IC;SO_10002;SO_LI_001;20111017010000;0;PR_0011;CU_0080;20111017;;20111205;20111205;69,29;USD;150;ST
A0;IC;SO_10002;SO_LI_001;20111100010000;0;PR_0011;CU_0080;20111017;;20111205;20111205;69,29;USD;200;ST
A0;IC;SO_10002;SO_LI_001;20111205010000;1;PR_0011;CU_0080;20111017;20111205;20111205;20111205;69,29;USD;200;ST
A0;IC;SO_10003;SO_LI_001;20111117010000;0;PR_0023;CU_0080;20111117;;20111215;20111215;21,58;USD;330;ST
A0;IC;SO_10003;SO_LI_001;20111215010000;1;PR_0023;CU_0080;20111117;20111215;20111215;20111215;21,58;USD;330;ST
A0;IC;SO_10004;SO_LI_001;20120117010000;0;PR_0011;CU_0069;20120117;;20120228;20120228;69,64;USD;200;ST
A0;IC;SO_10004;SO_LI_001;20120228010000;1;PR_0011;CU_0069;20120117;20120228;20120228;20120228;69,64;USD;200;ST
A0;IC;SO_10005;SO_LI_001;20120210010000;0;PR_0011;CU_0080;20120210;;20120504;20120504;69,29;USD;200;ST
A0;IC;SO_10005;SO_LI_001;20120504010000;1;PR_0011;CU_0080;20120210;20120504;20120504;20120504;69,29;USD;200;ST
A0;IC;SO_10006;SO_LI_001;20120215010000;0;PR_0023;CU_0024;20120215;;20120406;20120406;48,56;USD;60;ST
A0;IC;SO_10006;SO_LI_001;20120323010000;0;PR_0023;CU_0024;20120215;;20120411;20120411;48,56;USD;60;ST
A0;IC;SO_10006;SO_LI_001;20120411010000;1;PR_0023;CU_0024;20120215;20120411;20120411;20120411;48,56;USD;60;ST
A0;IC;SO_10007;SO_LI_001;20120313010000;0;PR_0011;CU_0069;20120313;;20120501;20120501;69,64;USD;150;ST
A0;IC;SO_10007;SO_LI_001;20120405010000;0;PR_0011;CU_0069;20120313;;20120501;20120501;69,64;USD;200;ST
A0;IC;SO_10007;SO_LI_001;20120418010000;0;PR_0011;CU_0069;20120313;;20120501;20120501;69,64;USD;150;ST
A0;IC;SO_10007;SO_LI_001;20120501010000;1;PR_0011;CU_0069;20120313;20120501;20120501;20120501;69,64;USD;150;ST
A0;IC;SO_10007;SO_LI_002;20120307010000;0;PR_0011;CU_0069;20120307;;20120418;20120418;69,64;USD;100;ST
A0;IC;SO_10007;SO_LI_002;20120405010000;0;PR_0011;CU_0069;20120307;;20120418;20120418;69,64;USD;150;ST
A0;IC;SO_10007;SO_LI_002;20120418010000;1;PR_0011;CU_0069;20120307;20120418;20120418;20120418;69,64;USD;150;ST
A0;IC;SO_10008;SO_LI_001;20120309010000;0;PR_0011;CU_0061;20120309;;20120323;20120323;68,94;USD;150;ST
A0;IC;SO_10008;SO_LI_001;20120312010000;0;PR_0011;CU_0061;20120309;;20120323;20120323;68,94;USD;100;ST
A0;IC;SO_10008;SO_LI_001;20120323010000;1;PR_0011;CU_0061;20120309;20120323;20120323;20120323;68,94;USD;100;ST
A0;IC;SO_10009;SO_LI_001;20120309010000;0;PR_0023;CU_0080;20120309;;20120330;20120330;21,58;USD;450;ST
A0;IC;SO_10009;SO_LI_001;20120330010000;1;PR_0023;CU_0080;20120309;20120330;20120330;20120330;21,58;USD;450;ST
```

- Additional or different CSV data repository information needed for new BI reports can be easily added or changed, and requires no rewrite of the SAC process logic

3 SAC Compliance

SAC compliance is a test whether all processed SAC data maintains its integrity in the entire data flow. To achieve the objectives of compliance requirements SAC has integrated a data interpretation tool for the debugging and monitoring of used source data, and a logging feature to report date of operations or server interruptions.

SAC uses the following standards-based and simple-to-use compliance testing methods:

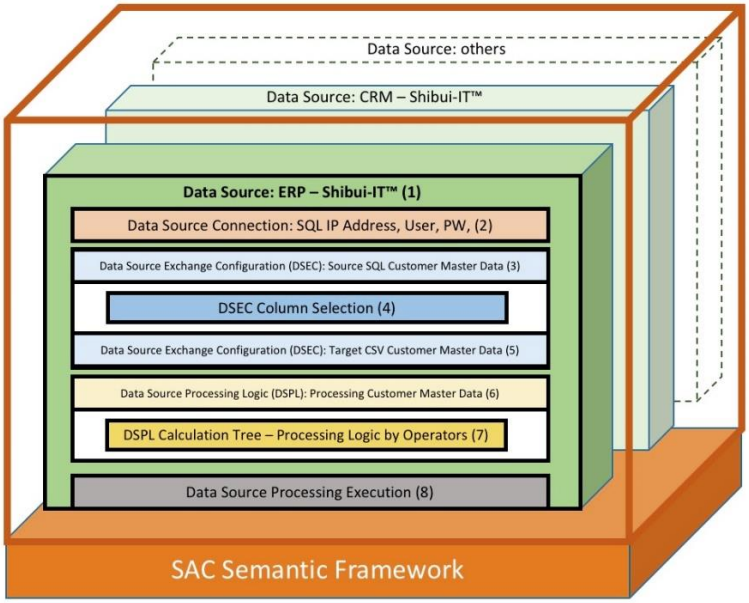
a) Speed & Volume
   - SAC uses a “first Full and then Delta” data read approach which allows faster data reading.

b) Accuracy
   - SAC checks the accurate extraction, transformation and loading from the data source to the Shibui-IT™ CSV databank

c) Syntax & Reference
   - Checks for poor data; e.g. invalid characters, erroneous character patterns, incorrect character cases and the correctness of the data source in relation to the required Shibui-IT™ databank attributes

4 Semantic SAC Framework

k•d•m’s semantic SAC framework includes a semantic model of the various data sources (ERP, CRM, others) organized under the SAC model. This model illustrates the semantic link of the various data sets, the data preparation, the mapping of data fields, the processing of the data sets and the data source processing execution in compliance with the used data model of the Shibui-IT™ Advanced Connector (SAC).

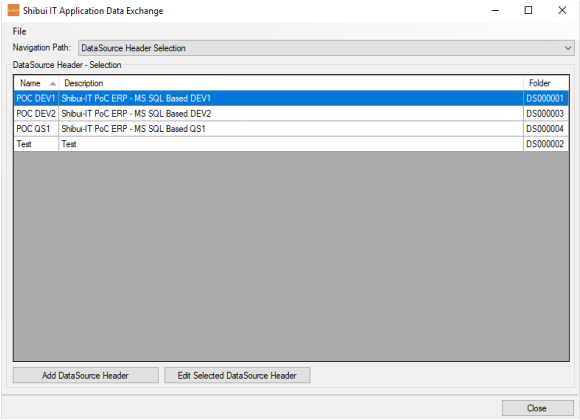


5 SAC Semantic Mapping

The use of a SAC semantic model illuminates every step, from the data source header selection, the various connection settings, the logic tree approach for about 30 different data sets which are needed to create the Shibui-IT™ BI reports, to the loading of the Shibui-IT™ CSV databank.

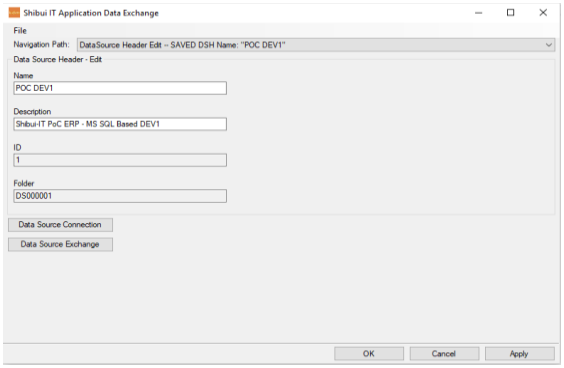
5.1 Data Source Header selection

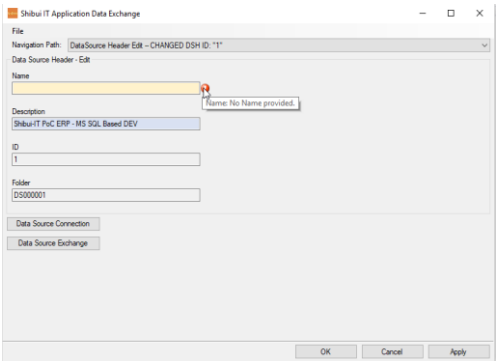
Next we see which data source the SAC has to deal with. Is the data source an ERP, a CRM or does the SAC have to deal with multiple data source.



5.2 Semantic System Link

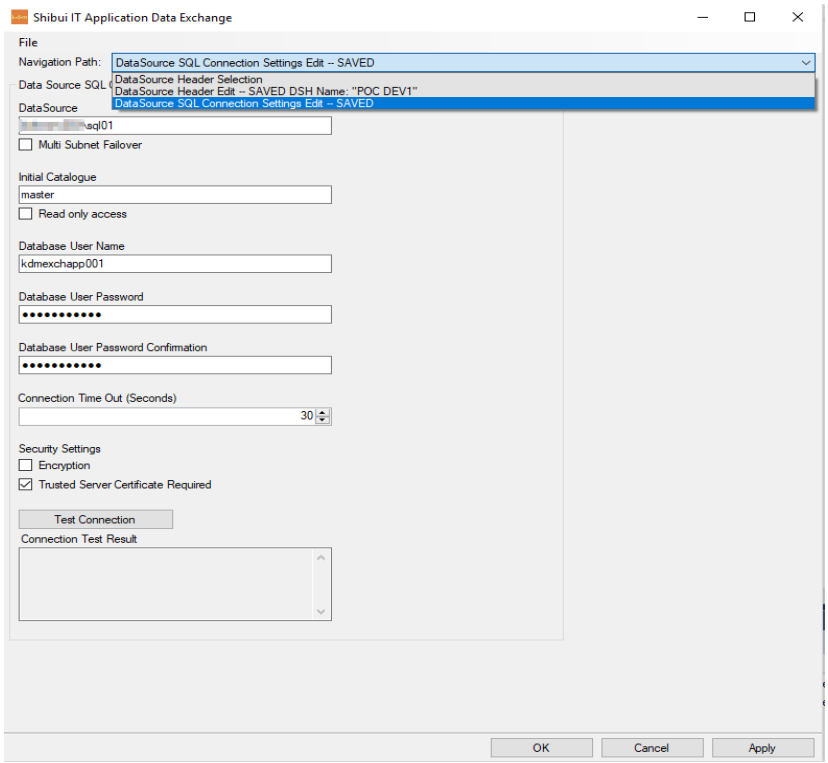
Below is the header description of the connected specific data source. In this case the SAC is connected to an ERP which is a fictive and simplified Shibui-IT™ ERP running on a SQL data base.





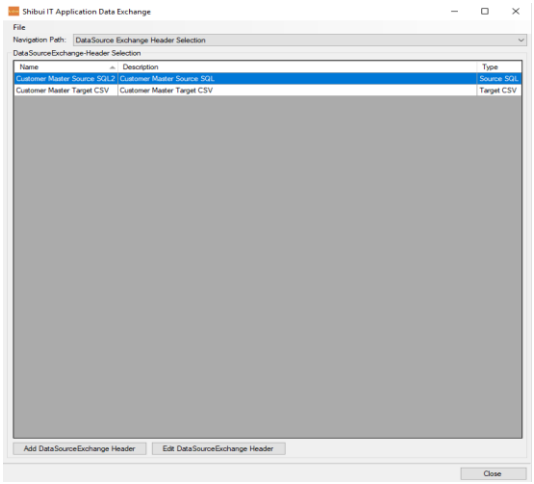
5.3 Semantic Connection to Data Source

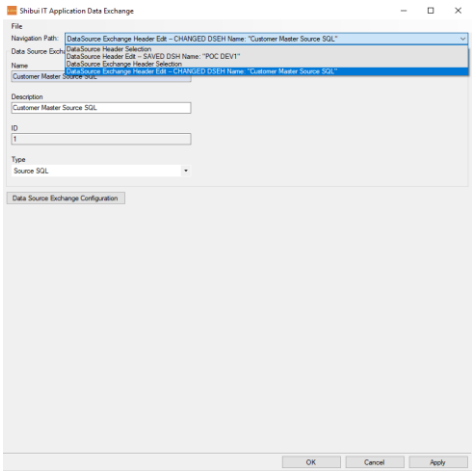
To connect the data source SQL input fields like SQL IP addresses, user name and user name passwords are needed.

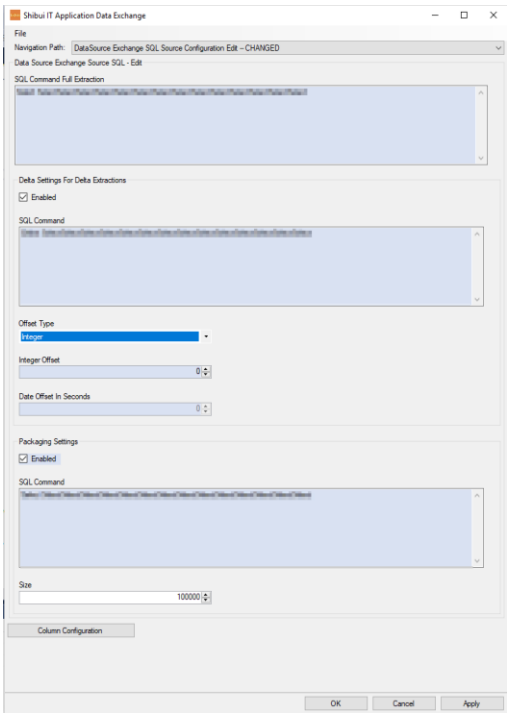


5.4 Semantic Link to Master & Transactional Data

This step illustrates how to configure one of the 30 different Shibui-IT™ Master Data sets. In the following configuration example, the "Customer data set" is selected.

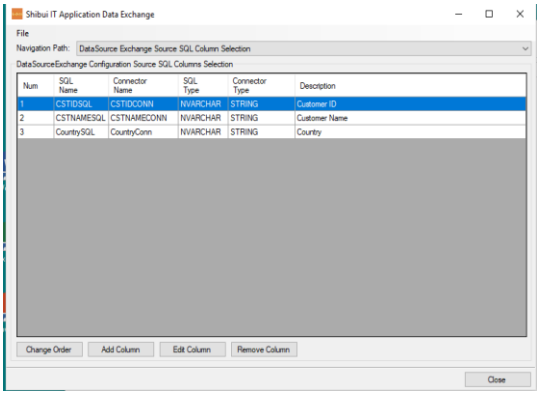


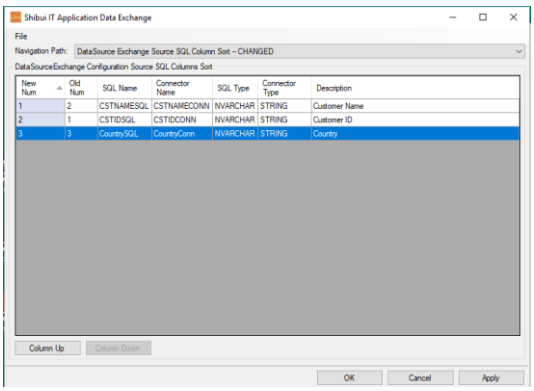


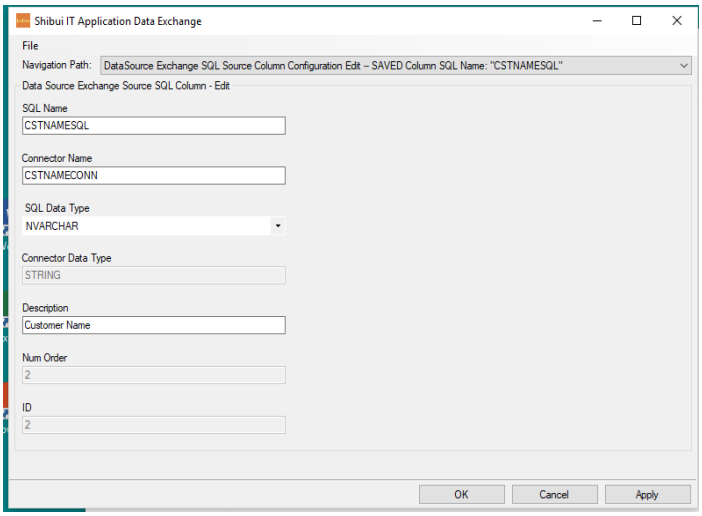


5.5 Semantic Link Data Properties Translation

The illustrations below preview how to translate the SQL Data Columns of the Customer Master data set into the corresponding SAC columns names needed for the Processing Logic Operators. This translation feature of SAC guarantees that the SAC Process Logic can be used for different data sources hence no re-write of the Logic is necessary.

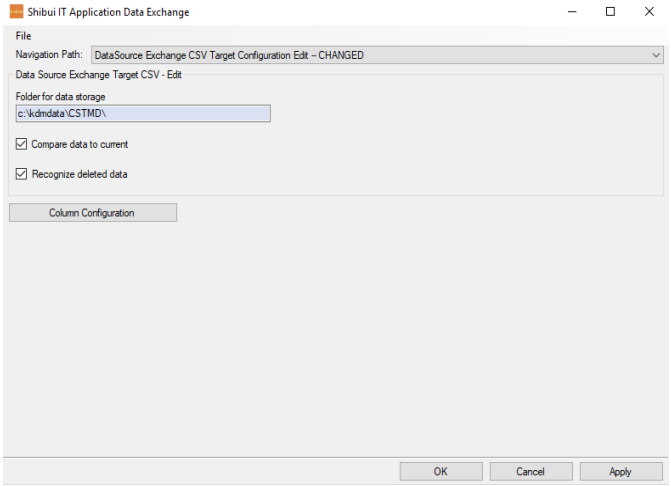






5.6 Semantic Connection to Target Source

Next we see how the connected specific target source in which the processed master data sets (e.g. “Customer data set”) are stored. Shibui-IT™ SAC uses a CSV data repository which is the data source of the Shibui-IT™ front-end tool.

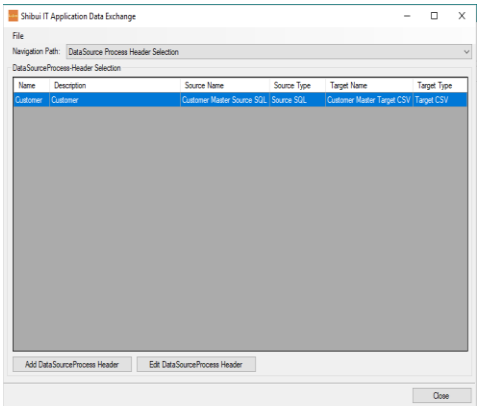


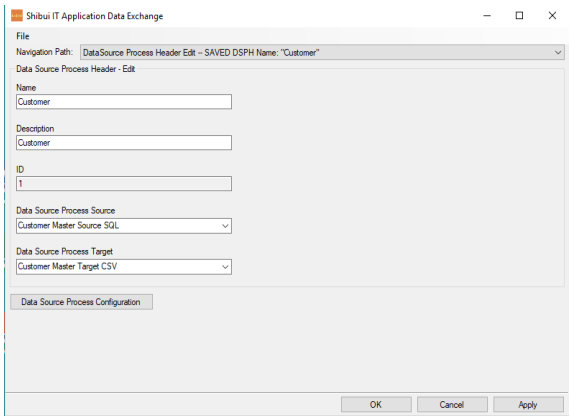
5.7 Semantic SAC Processing

SAC processing logic uses a decision/logic-tree model and associated operations to process the specific SAC algorithm for each of the 30 different master data sets.

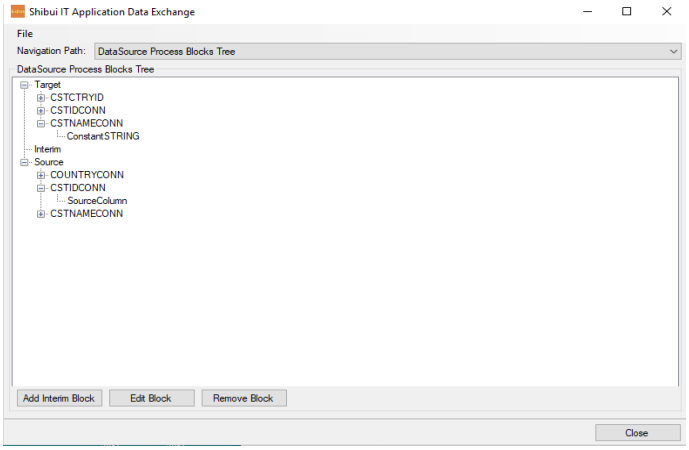
5.7.1 Definition of Logic Trees

The following header shows how to define and name in the SAC user system the 30 different process trees for all needed master data sets.



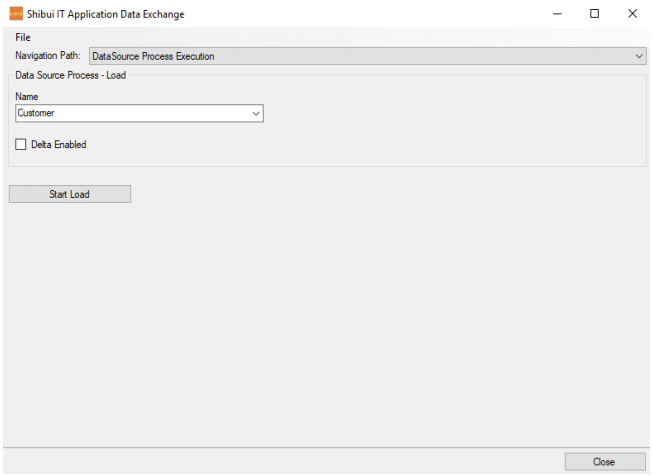


5.7.2 Semantic SAC Process Tree



5.8 Semantic Processing Execution

The last step of the SAC semantic model is the execution of a Master Data Set. In the above-described case it is the "Customer data set". After the execution the result will be stored in Shibui-IT™ CSV databank.



6 Summary

k•d•m has built a Proof of Concept for the Shibui-IT™ Advanced Connector (SAC). The connected customer data source is a simplified Shibui-IT™ ERP/CRM running on a SQL server. For the PoC k•d•m uses the data set of a fictitious Tier 2 components company with 100 customers in 12 countries and 5 regions, with 4 product lines and 30 products which includes mature products, new products, opportunity funnel and end of life products as well as IP (Intellectual Property) business. The company has 12 Account Manager, a data history back to 2001, has an ERP system and a CRM tool as data source and a mature business of about $10M in 2014 and a growth potential to $50M in 2016. The processed 30 different master data sets are stored in the Shibui-IT™ CSV databank. After they are processed in the Shibui-IT™ front-end tool to generate the various custom Shibui-IT™ BI reports.