Project ‚BI Introduction‘

Specification and Documentation

Author : T. Uhlenkamp, M. Bellmann, P. Staack, C. Panschog

Date : 13.8.13

Status : Final

Updated : 15.08.07 (LQV)

Content list

[1.1 Project goals 4](#_Toc438129181)

[1.2 Project scope and business Intelligence (BI) system landscape 4](#_Toc438129182)

[1.3 Project organisation 5](#_Toc438129183)

[2.1 Overview of required Reports and Data entry masks 6](#_Toc438129184)

[2.2 Detailed Design of Reports and Data entry masks 6](#_Toc438129185)

[2.3 Portal Layout and Navigation 7](#_Toc438129186)

[3.1 Rieckermann order terms and definitions 8](#_Toc438129187)

[3.2 Currency 9](#_Toc438129188)

[3.3 Data Value Definitions 9](#_Toc438129189)

[3.4 Definition of Order Stages 9](#_Toc438129190)

[3.4.1 Order intake 9](#_Toc438129191)

[3.4.2 Order Backlog 10](#_Toc438129192)

[3.4.3 Order completed 10](#_Toc438129193)

[3.5 Workflow of Budgeting process 10](#_Toc438129194)

[3.6 Budgeting frequency and horizon 11](#_Toc438129195)

[4.1 IT infrastructure 12](#_Toc438129196)

[4.2 BOARD folder structure 14](#_Toc438129197)

[4.3 Backup BOARD Server 14](#_Toc438129198)

[4.4 Data Warehouse (SQL Server) 15](#_Toc438129199)

[4.4.1 Nomenclature for database tables / views 15](#_Toc438129200)

[4.4.2 Source tables of pre-systems (like ERP) 15](#_Toc438129201)

[4.5 Technical Integration 19](#_Toc438129202)

[4.5.1 Preamble 19](#_Toc438129203)

[4.5.2 Tables 19](#_Toc438129204)

[4.5.3 Views 22](#_Toc438129205)

[4.6 ETL Process 23](#_Toc438129206)

[4.6.1 Loading tables in the landing zone 23](#_Toc438129207)

[4.6.2 Loading the Facttables 23](#_Toc438129208)

[5.1 Order Intake 23](#_Toc438129209)

[5.1.1 Views OrderIntake for trade values in Ventas 23](#_Toc438129210)

[5.1.2 Views OrderIntake for commission values from Ventas 25](#_Toc438129211)

[5.1.3 View for OrderIntake from LOD 25](#_Toc438129212)

[5.1.4 View for OrderIntake from HKG Access orders before 2013 27](#_Toc438129213)

[**(For reference only, as HKG Access orders before 2013 are disconnected to Data warehouse)** 27](#_Toc438129214)

[5.1.5 View for OrderIntake from HKG Access orders since 2013 onwards 27](#_Toc438129215)

[5.1.6 Facttable „Fakt\_OrderIntake“ 27](#_Toc438129216)

[5.1.7 Facttable “HIST\_OrderIntake” 28](#_Toc438129217)

[5.2 Order Complete 30](#_Toc438129218)

[The fact tables “Fakt\_OrderComplete” and “Fakt\_OrderComplete\_DB2” are populated by the SSIS package “Facttable\_OrderComplete.dtsx”. 30](#_Toc438129219)

[5.2.1 View Order Complete for Trade orders in Ventas 31](#_Toc438129220)

[5.2.2 View Order Complete for Commision from Ventas 33](#_Toc438129221)

[5.2.3 View OrderComplete from HKG Access orders before 2013 – Solomon 35](#_Toc438129222)

[5.2.4 View OrderComplete from HKG Access orders since 2013 onwards 35](#_Toc438129223)

[5.2.5 Order Complete from LOD 36](#_Toc438129224)

[5.2.6 Loading Factables “FAKT\_OrderComplete” and “FAKT\_OrderComplete\_DB2” 37](#_Toc438129225)

[5.3 Order Backlog 38](#_Toc438129226)

[5.3.1 Order backlog actual (shipment) 38](#_Toc438129227)

[5.3.2 Order Backlog Stock per month 38](#_Toc438129228)

[5.3.3 Order Backlog Stock Historical per month 39](#_Toc438129229)

[5.4 TS KPIs 39](#_Toc438129230)

[7.1 Load Capsule in BOARD 40](#_Toc438129231)

[7.2 linkFISH ETL Starter 40](#_Toc438129232)

[7.3 Scheduled Tasks 42](#_Toc438129233)

[9.1 User license profiles of BOARD - an overview 44](#_Toc438129234)

[9.2 Authorization Concepts within capsules and databases 44](#_Toc438129235)

[9.3 How to create and maintain user profiles 44](#_Toc438129236)

[9.4 Current Authorization Matrix 45](#_Toc438129237)

[10.1 Detailed specification of Reports and Data entry masks 46](#_Toc438129238)

[10.2 Detailed specification of portal layout and navigation 46](#_Toc438129239)

[10.3 Detailed specification of authorizations per BOARD user 46](#_Toc438129240)

[10.4 List of Definitions 46](#_Toc438129241)

1. Introduction

Project goals

The project goals across several project modules are:

1. Transparency increase about decision-relevant Sales and Finance Key Performance Indicators (KPI´s)
2. Quality and efficiency increase of reporting and budgeting processes by implementing a state of the art business intelligence solution
3. Implementation of a rolling forecast process for budgeting of sales and finance KPI´s
4. Extension of the required budgeting scope with further decision relevant KPI´s
5. Extension and standardization of contribution margin scheme with other business models (merchandises like Services) to enable

The current project module 1 until mid of October 2013 focuses on the first three project goals.

Project scope and business Intelligence (BI) system landscape

The project scope is defined in the attached project plan. The figure below shows the scope and building blocks of the new Rieckermann Business Intelligence Solution. Two new software packages will be implemented: BOARD and MS SQL

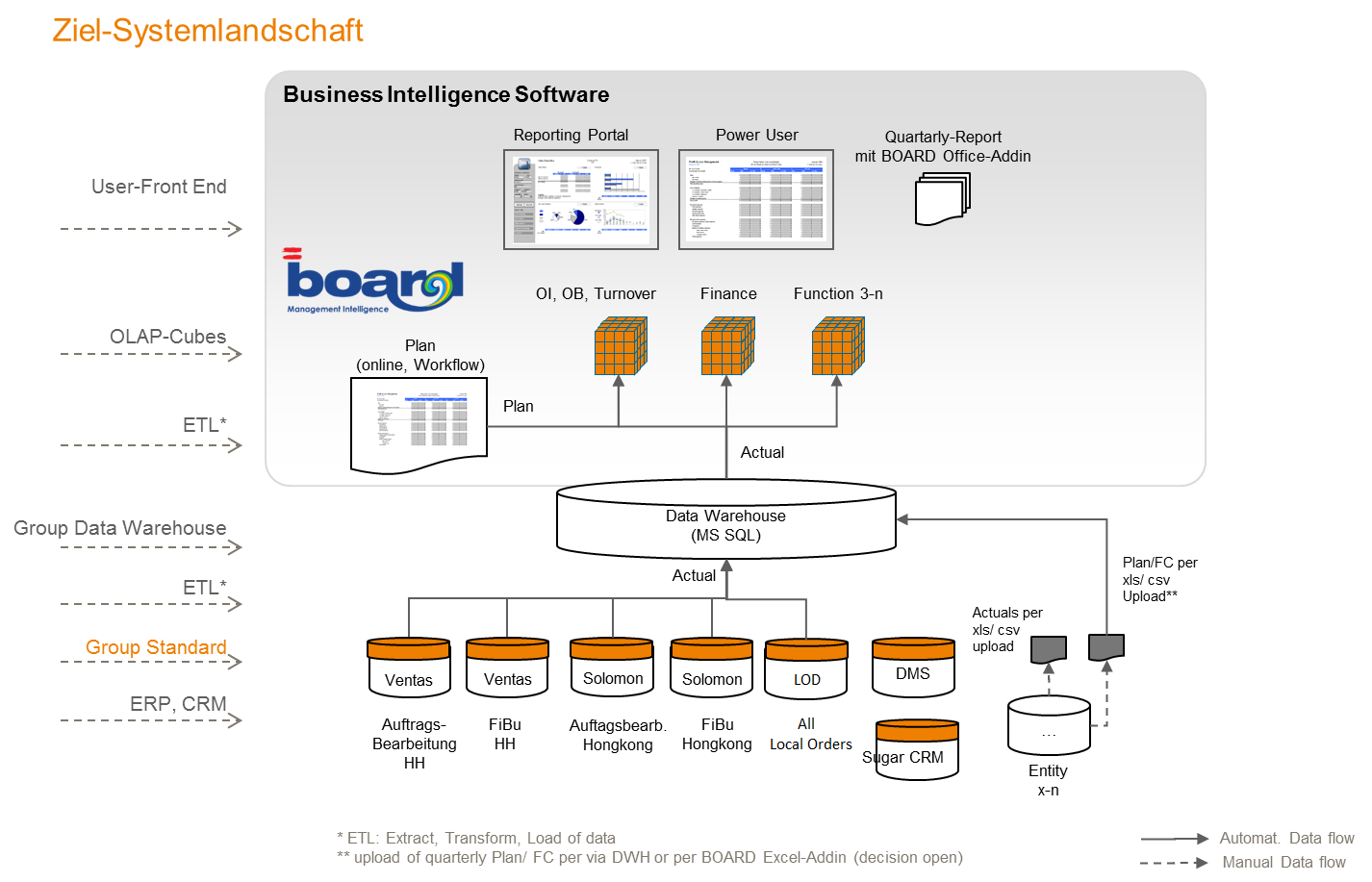


Figure 1: Scheduled Business Intelligence System landscape of Rieckermann Aug. 03rd 2013

The relational database (MS SQL) will be connected with ETL processes with five ERP systems/ modules: Ventas Auftragsbearbeitung, Ventas Fibu, Solomon Auftragsbearbeitung, Solomon Fibu and LOD (comprising all local orders).

In BOARD three different building blocks will be implemented: multidimensional Database and Cubes, Reporting Portal, Planning workflow and data entry masks.

Any connections to other Rieckermann systems and databases are out of scope of the current project module 1 until Oct. 16th 2013.

Project organisation

The below shown project organisation delivers the new reporting and budgeting solution until October 2013. The sub-project teams are staffed with internal and external experts to ensure a successful project.

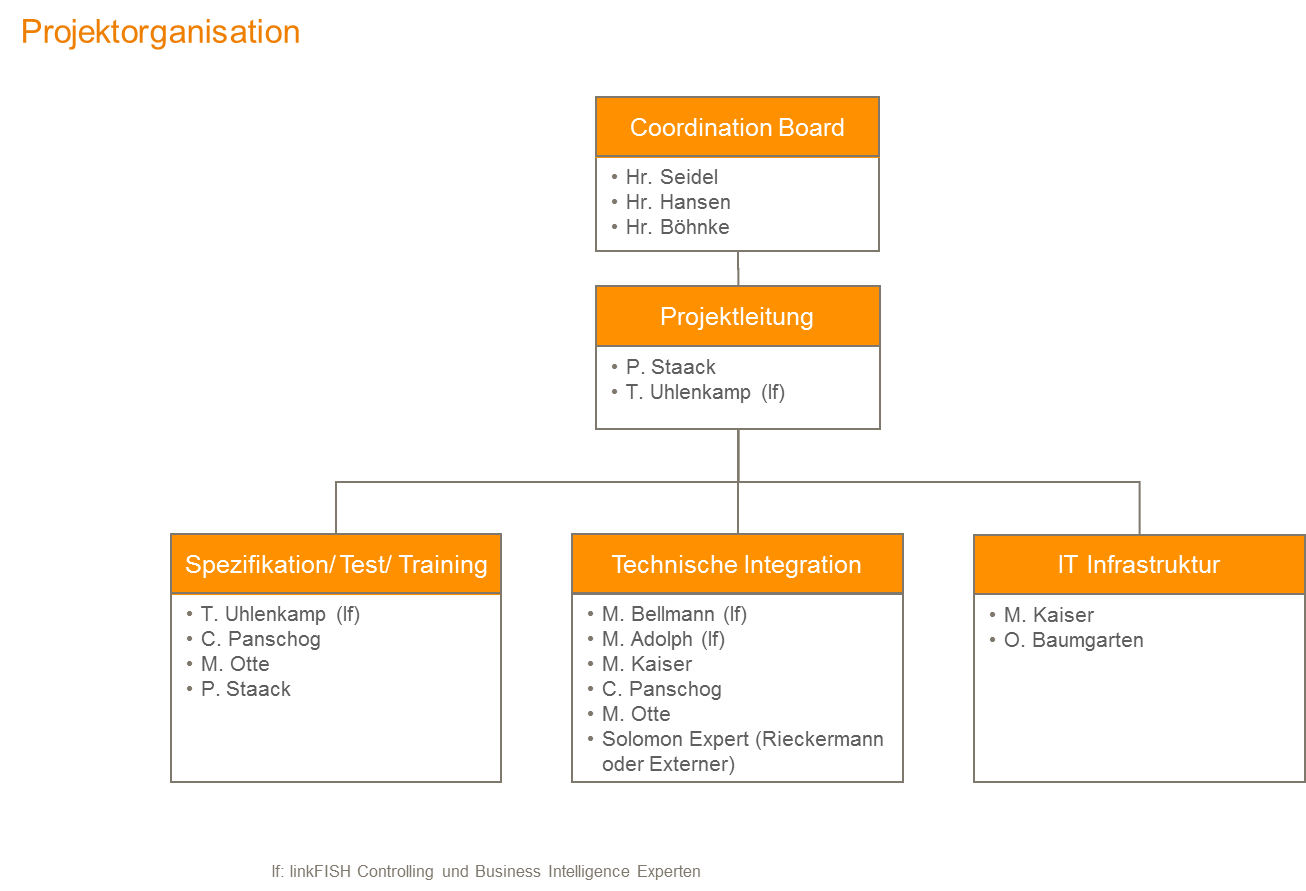


Figure 2: Project organisation

1. Module 1 : Reporting and Budgeting Design

Overview of required Reports and Data entry masks

The new Budgeting and Reporting Solution (BOARD) of Rieckermann provides several reports and data entry masks. The current project scope delivers seven reports (incl. three reports to support the budget process) and one data entry mask (see also figure below).



Figure 3: ID and title of data entry masks and reports in BOARD

Detailed Design of Reports and Data entry masks

Each report and data entry is specified in terms of columns, rows, selections, functions and title. The specifications were defined in several interviews and workshops with the Rieckermann Controlling Team. One example of a specified data entry mask is shown below. Please see attachment 6.1 for all detailed report and data entry masks specifications.



Figure 4: Budgeting process – data entry mask of business unit plans (Technical ID: E01)

Portal Layout and Navigation

The portal structure and layout with its start page, masks and screens is defined in attachment 6.2 ‘Navigation structure BOARD’.

The attachment contains the following definition:

* Tab 1 Layout: font, size, colors etc.
* Tab 2 Site map: definition of reports per user group
* Tab 3 all start screen
* Tab 4-8: Mask and screens for each user group (FCC, CB, BU, IN)

The figure below illustrates the available reports for the user group FCC.



Figure 5: Screenshot of FCC Mask and Home

1. Module 2: Corporate Standards and processes

Rieckermann order terms and definitions

This chapter is supposed to explain the key terminology used in the BI-Tool (also basis for a glossary to be established in the BI-Tool). This refers to entities used in the database as well as other key descriptive terms. It does not describe terms used by the underlying software. These can be looked up under help.board.com

Please refer to attachment 6.4 for all definitions.

Tab: Abstract of definitions in attachment 6.4



Currency

All order values will be reported in EUR. In addition the original order-currency (e.g. HKD) is also available for reporting.

Data Value Definitions

There are two figures per order that must be determined:

* order value
* DB2 (order value less order-related costs)

These figures will be determined on a order position-basis. Therefore the order-related costs will be spreaded to the order positions based on the value.

An order runs through three different stages, which will described in the following:

Definition of Order Stages

### Order intake

When an order is committed by a customer, the value will be shown as order intake. The calculated costs for the order will be used for determination of the DB2.

Changing of order data results in a change of order intake on the date of change and not on the order date.

**Example**:

1. **Order registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **order no** | **order pos** | **order date** | **change date** | **Value** | **order-driven costs** |
| 1000 | 01 | 01.08.2013 | - | 2000 | 1800 |
| 1000 | 02 | 01.08.2013 | - | 1000 | 900 |

resulting order Intake

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **order no** | **order pos** | **order intake date** | **value** | **DB2 (value - costs)** |
| 1000 | 01 | 01.08.2013 | 2000 | 200 |
| 1000 | 02 | 01.08.2013 | 1000 | 100 |

1. **Order change**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **order no** | **order pos** | **order date** | **change date** | **Value** | **order-driven costs** |
| 1000 | 01 | 06.09.2013 | 06.09.2013 | 2000 | 1900 |
| 1000 | 02 | 07.09.2013 | 07.09.2013 | 500 | 450 |

resulting order Intake

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **order no** | **order pos** | **order intake date** | **value** | **DB2 (value - costs)** |
| 1000 | 01 | 01.08.2013 | 2000 | 200 |
| 1000 | 02 | 01.08.2013 | 1000 | 100 |
| 1000 | 01 | 06.09.2013 | 2000-2000 = 0 | 100 -200 = -100 |
| 1000 | 02 | 07.09.2013 | 500-1000 = -500 | 50 – 100 = -50 |

### Order Backlog

As long as the order is in the stage of „Order intake“, the value und DB2 is in the order backlog. This is a stock value.

### Order completed

When an order, or a position of an order, is completed (shipped) the order position changes to “completed”.

Workflow of Budgeting process

During the budget process four levels of responsibilities were defined: Business Unit Manager, Industry Manager, Controlling and Coordination Board.

The first step of the budget process is the initialization of the planning system BOARD (by Controlling), followed by data entry of business unit plans (by Business Unit Manager). All following process steps are related to ‘Review and sign off’ of the plan. Please see also the illustrated workflow diagram below.

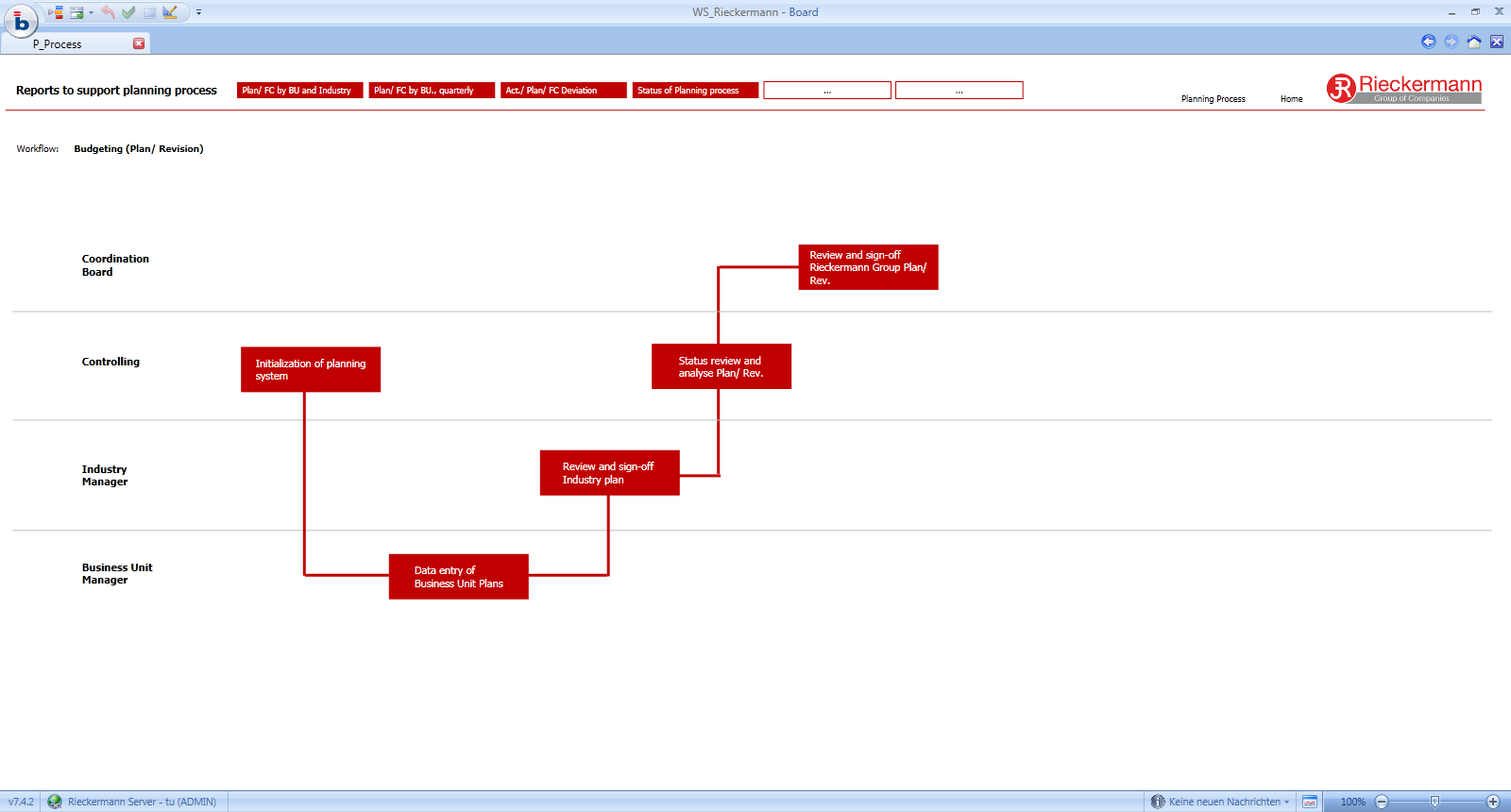


Figure 6: Workflow of budget process

The workflow and its red buttons contain links to all budget related reports and data entry masks.

Budgeting frequency and horizon

The new budgeting solution BOARD and the above defined workflow enable Rieckermann to execute four budget/ revision processes per year (see also figure below).

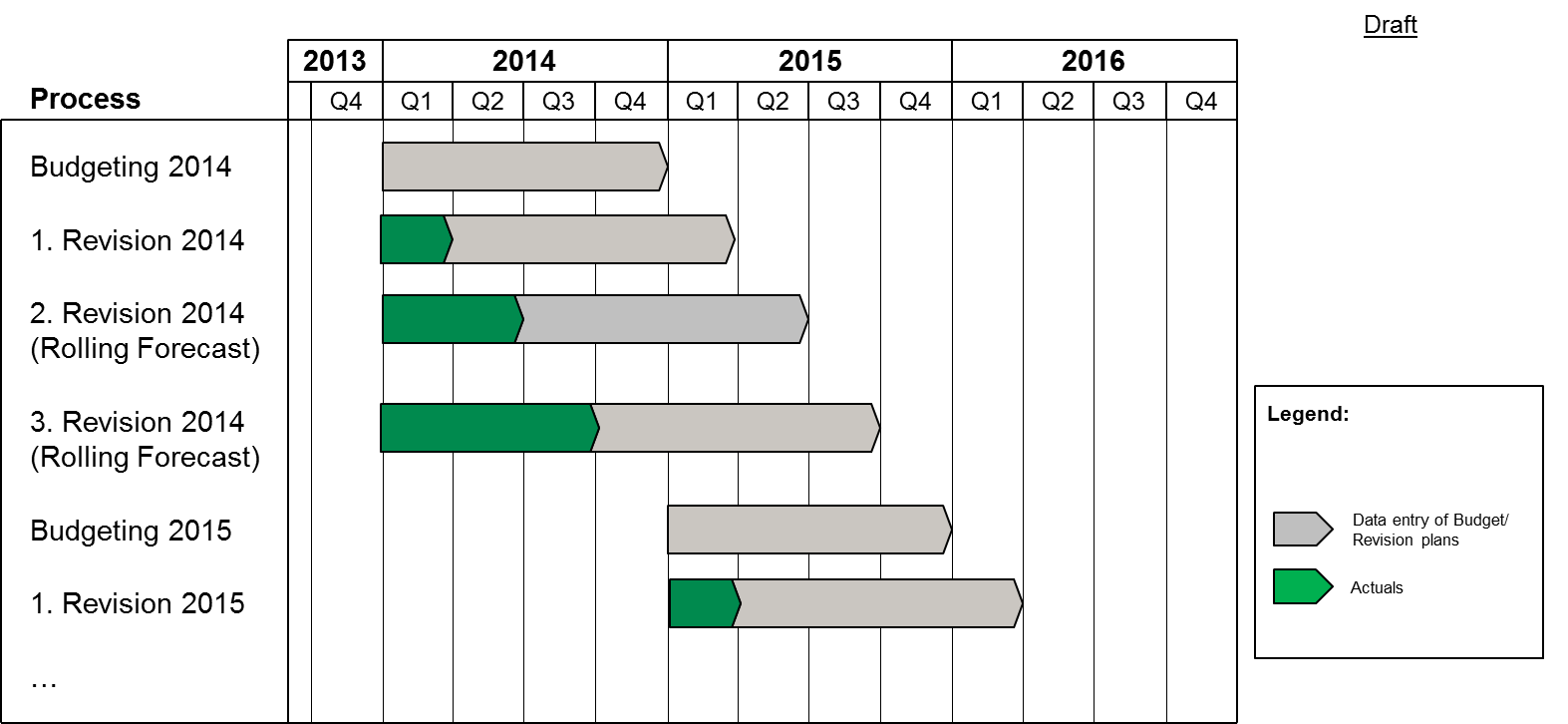


Figure 7: Frequency and horizon of the budgeting/ revision process

**Frequency:**

The new budgeting processes schedule four budgeting/ revision cycles per year. The implemented budgeting solution BOARD will allow these four cycles, but will also function if one or several cycles drop out. For example: it is possible to execute the Budgeting process 2014 and only the 3. Revision process.

**Horizon:**

To improve the far-sightedness and to enable earlier decision making Rieckermann establishes with the new budgeting solution a rolling forecast process. The planning horizon of every budgeting/ revision process covers 12 months.

1. Module 3: Overview Technical Integration of BOARD, MS SQL

IT infrastructure

The source data for the reporting portal is located in three different application systems. „Ventas“ is the main order and finance application system which is used in Hamburg. „Solomon“ (finance) are located and operated in Hongkong, and LOD (order registration).

All relevant data will be collected in one relational database, the „Data Warehouse“ (DWH). In the DWH the data will be transformed and conditioned für reporting and analysis, e.g.:

1. adaption of data (e.g. different charts of accounts)
2. currency conversion
3. identification and cleansing of false data

The DWH is running on a Windows server und a SQL Server 2008 R2. For building, operating und monitoring of the data flows, Microsofts SQL Server Integration Services (SSIS) is used.

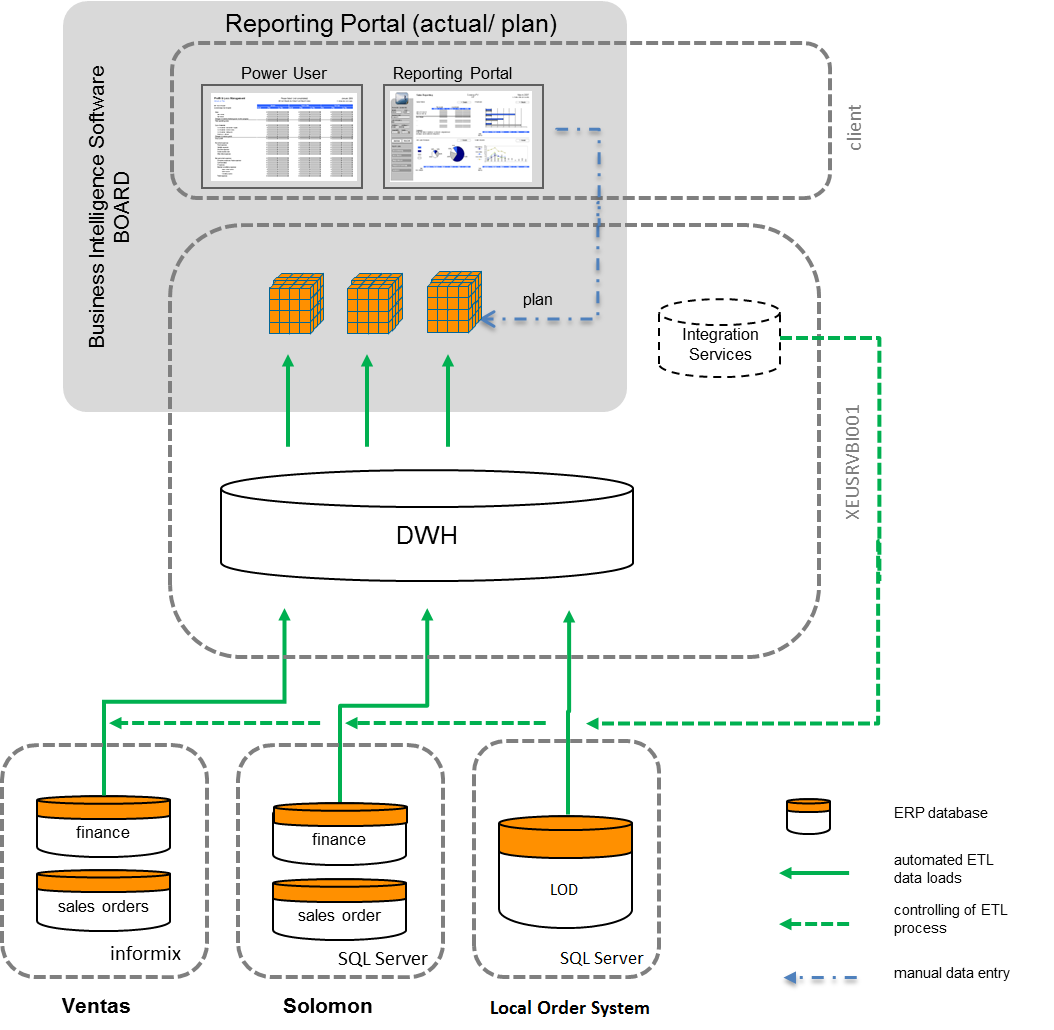


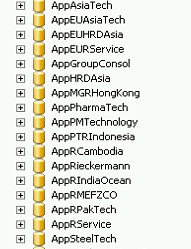
Figure 8: Components of the Business Intelligence IT architecture

**Ventas**

Ventas is based on a Informix Database System. The database is located on Server “jru1” (1.70.159.1). Both database instances “jr1” and “jr2” are source databases for the DWH.

**Solomon**

The Solomon database (SQL Server) is located on Server “10.1.1.3”. Each Solomon Client use its own Database.



All Solomon databases, except AppSteeltech, will be transferred in the DWH.

**LOD**

The LOD database (SQL Server) is located on Server 10.99.1.36, database name is LOD

BOARD folder structure

The BOARD folder is located in “D:\BOARD” on the BOARD server. The capsule folder has the following structure:



The “live” capsule is located in the main capsule folder and the current development capsule in the folder “development”. When the current development is finished und tested successfully, the capsule will bei copied to the main capsule folder.

Backup BOARD Server

For a fully backup of the BOARD and staging data, the following items must be included in the server backup:

1. **BOARD Folder**  
   The folders “d:board” and “d:\ssis” with all subfolders must be backuped. Before backup the service “board7” must be stopped.
2. **SQL Server Database “Staging”**

The SQL Server Database “Staging” must be backuped.

Data Warehouse (SQL Server)

Nomenclature for database tables / views

The following table shows the nomenclature for the database table and views in the Data Warehouse. The intention of the nomenclature is that the name of the tables describes the origin and the function of the table/view:

|  |  |  |
| --- | --- | --- |
| **Type** | **Nomenclature** | **Example** |
| complete table of pre-system | LZ\_<shortname source system>\_<name of table in source system> | LZ\_VENTAS\_apos (order positions from Ventas), LZ\_ LZ\_SOLOMON\_GLTran (general ledger from Solomon) |
| intermediate temporary data | TMP\_ |  |
| view | V\_ |  |
| historical data | HIST\_<figure> | HIST\_OrderIntake |
| target table fact data | FAKT\_<figure> | FAKT\_Turnover |
| target table dimensional data | DIM\_<dimension> | DIM\_BusinessUnit |

Source tables of pre-systems (like ERP)

**Ventas:**

Within the data model of Ventas, trade order and commission orders are stored in separate tables. The following picture describes the tables and their relations.



|  |  |
| --- | --- |
| **name** | **Description** |
| vkopf | order header for trade order |
| vpos | order positions for trade order |
| epos | shipped order positions for trade orders |
| akontr | order header for commission orders |
| akpos | order positions for commision order |
| apos | shipped order positions for commision orders |
| kalk | order calculations |
| ksttr | join table from order to calculation |
| rgko | invoice |
| rgpo | invoice positions |
| konto | account master data |
| sach\_buch | accounting transactions |
| adress | adress data (customer, supplier) |

**LOD / Solomon:**

LOD is the client application which allows all eligible Rieckermann users (regardless of location) to register and update information of orders. LOD data is divided into three groups:

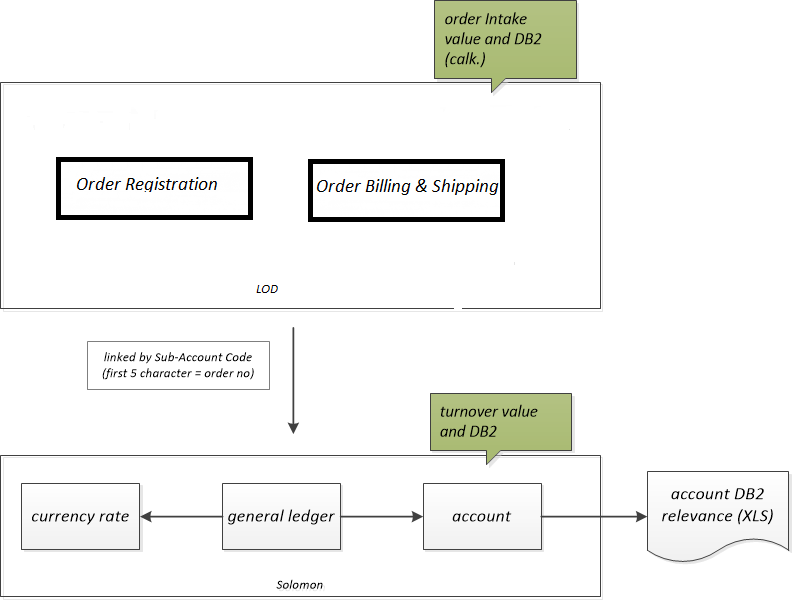
* Newly registered Hong Kong orders which are registered by Hong Kong users – called HKG orders.
* FOXPRO orders which are actually historical FOXPRO data (FOXPRO was replaced by LOD, therefore FOXPRO data was also migrated to LOD) – called FOXPRO orders.
* Orders from offices other than Hong Kong – called non-HKG orders.

There are some requirements for an order in LOD in order to be transferred to data warehouse. First of all, the order must be confirmed and not-cancelled. Secondly, the order is direct-purchased or its procurement center must meet following conditions:

* If non-HKG order => It should not be procured by “Rieckermann GmbH”, “Asia Technik Ltd.” and “Rieckermann Services Ltd.”
* If HKG order => It should not be procured by “Rieckermann GmbH”
* In case it is FOXPRO order => It should not be procured by “Rieckermann GmbH” and its division is not '00','01','11','12','13','14','15' or '1S' and its product is not 'Spare Part Onward Charges'

For FOXPRO and HKG orders, turnover information come from Solomon, therefore a lookup to Solomon databases to get necessary information is called when transferring these types of order. For non-HKG orders, all information is directly extracted from LOD.

(Please see 5.2 for more details).



For the order intake value / DB2 the order registration in LOD is used as data source. The determination of the turnover value / DB2 is different based on the business type.

**Trade order:**

value: The net total of the sales bookings

DB2: The net total of all DB2 relevant bookings

**Commision order:**

value: The order intake value (LOD)

DB2: The net total of all debitor sales bookings

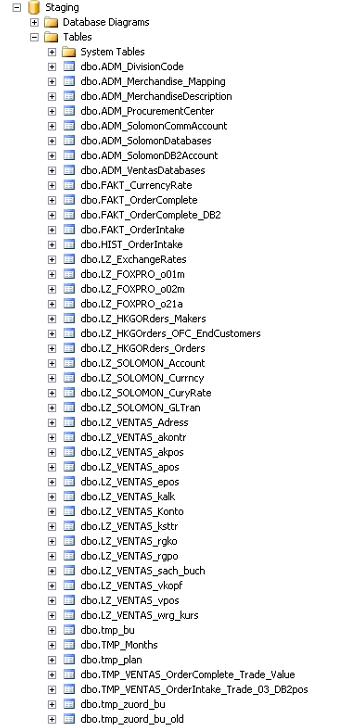
For every book entry the values will also be translated from local currency to “EUR” (curreny exchange table in Solomon). The definition of the chart of accounts with the DB2-relevant accounts is defined in a excel workbook.

Technical Integration

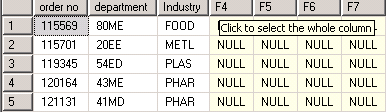
### Preamble

In this Documentation you can see the ETL Processes wich get the data from the Quellsystems Ventas, LOD and Solomon in our Datawarehouse. The Datawarehouse is stored in an SQL Server and the Name of the database is Staging Area. This documentation will help to get quick information how the key figures in Board are calculated and where they come from.

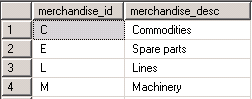
### Tables



* Tables that contains the präfix “ADM” are administration tables. With this tables we can attach certain informations to our Board System. All administration tables are managed by Mr. Panschog. Administration tables include following tables:
* ADM\_Mapping\_Department\_Industry: mapping from former department to industry for VENTAS orders.



* ADM\_MerchandiseDescription: contains merchandise code and description.



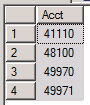
* ADM\_ProcurementCenter: contains code and description of main procurement center.



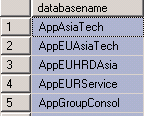
* ADM\_SolomonCommAccount: contains local Solomon commission accounts code (not including Solomon Shanghai’s ones). In Solomon, comission orders are booked by these accounts.



* ADM\_SolomonCommAccount\_SH: contains local Solomon Shanghai commission accounts code.



* ADM\_SolomonDatabases: contains list of all Solomon source databases name.



* ADM\_SolomonDB2Account: contains all Solomon DB2 accounts (not including Solomon Shanghai’s ones).



* ADM\_SolomonDB2Account\_SH: contains all Solomon Shanghai DB2. accounts.

|  |
| --- |
| acct |
| 41110 |
| 46000 |
| 47000 |
| 47100 |
| 48000 |
| 48004 |
| 48100 |
| 48200 |
| 48250 |
| 48300 |
| 49970 |
| 49971 |
| 56000 |
| 57000 |
| 58000 |
| 58007 |
| 58200 |
| 58300 |
| 58350 |
| 58353 |
| 58400 |
| 58800 |
| 58850 |
| 59900 |

* ADM\_VentasDatabases: contains list of all Ventas source databases name.



### Views



## ETL Process

### Loading tables in the landing zone

In the first step of our loading process we load all relevant tables from every sourcesystem in the landing zone (Tables that begins with LZ). To get a 1:1 copie of every relevant table we use SSIS packages. These packeges are stored in this file: “RieckermannStaging.sln“ which you can find in the folder „D:\SSIS\RieckermannStaging“. In this file there are three SSIS packages to get the date from the source system in our Datawarehouse:

* StagingVentas
* StagingFoxpro (LOD data)
* StagingSolomon
* StagingHKG\_Orders

### Loading the Facttables

In the second step of our loading process we fill the Factables in our Datawarehouse. There are SSIS packages for these step. It is necessary to get the right execution order to load the data in our Facttables:

1. Execute SSIS package „FAKT\_OrderIntake“

2. Execute SSIS package „FAKT\_OrderIntake\_HIST“

3. Execute SSIS package „FAKT\_OrderComplete“

1. Structure of Views for getting the data in Facttables

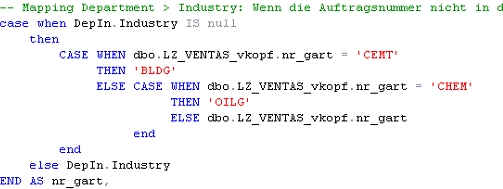
Order Intake

### Views OrderIntake for trade values in Ventas

To get the Order Intake values for trade orders from Ventas in our Facttable “FAKT\_OrderIntake” we use the following Views:

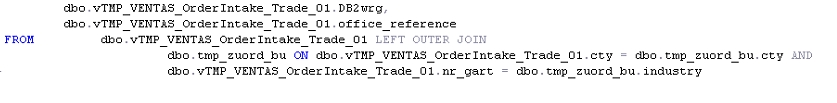
* vTMP\_VENTAS\_OrderIntake\_Trade\_01
* vTMP\_VENTAS\_OrderIntake\_Trade\_02\_BU
* vTMP\_VENTAS\_OrderIntake\_Trade\_03\_DB2pos

The View “**vTMP\_VENTAS\_OrderIntake\_Trade\_01**” attach all relevant Ventas tables that contain Values for trade orders in Ventas. In this View we use an administration table to get the right industry on the basis of the department code:



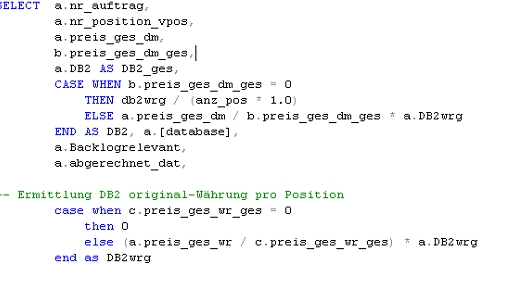
„detection of industry“

Buildung up on View „vTMP\_VENTAS\_OrderIntake\_Trade\_01” we create the View **vTMP\_VENTAS\_OrderIntake\_Trade\_02\_BU**“. In this view we identify the Business Unit to the Order Intake trade Values in Ventas. To get the right Business Unit we use the administration table “tmp\_zuord\_bu”. In this table we identify the Business Unit over the combination of industry\_code and country code:



„detection of bu““.

At least we need the Order Intake DB2 Value. To get this, we use the view „**vTMP\_VENTAS\_OrderIntake\_Trade\_03\_DB2pos**“ which is also build up on the view „vTMP\_VENTAS\_OrderIntake\_Trade\_01”:



„detection of DB2 in View „vTMP\_VENTAS\_OrderIntake\_Trade\_03\_DB2pos““

Based on the Views „vTMP\_VENTAS\_OrderIntake\_Trade\_02\_BU“ and „vTMP\_VENTAS\_OrderIntake\_Trade\_03\_DB2pos“ we fill the Facttable “FAKT\_OrderIntake”. Please read point 5.1.4 in this documentation for further information.

### Views OrderIntake for commission values from Ventas

To get the Order Intake Values for Commision Orders from Ventas we use the same procedure, that we explain in point 5.1.1. The only difference is, that we use different source tables to get the first View. The following views are used to indentify the Order Intake commission values:

* vTMP\_VENTAS\_OrderIntake\_Comm\_01
* vTMP\_VENTAS\_OrderIntake\_Comm\_02\_BU
* vTMP\_VENTAS\_OrderIntake\_Comm\_03\_DB2pos

### View for OrderIntake from LOD

When transferring LOD data to BOARD, orders are categorized into three groups: newly registered Hong Kong orders, old FOXPRO orders and newly registered non-HKG orders due to differences in their logics.

#### View vLOD\_HKGOrderNo for newly registered Hong Kong orders

This view contains list of newly registered HKG order numbers which meet following conditions:

* Not an old FOXPRO order (HistoricalSource <> ‚FOXPRO‘)
* Confirmed and not-cancelled order. (Status <> 2 and Confirmed = 1)
* Direct purchase from PC or procurement center is not Hamburg

(PCPurchase = 1 or ProcCenter <> “Rieckermann Gmbh”)

#### View vLOD\_FOXPROOrderNo for the old FOXPRO orders

This view contains list of old FOXPRO order numbers which meet following conditions:

* Confirmed and not-cancelled order. (Status <> 2 and Confirmed = 1)
* Division Code not 00,01,11,12,13,14,15 and 1S
* Product not ‚Spare part onward charges‘
* Direct purchase from PC or procurement center is not Hamburg.

#### View vLOD\_NonHKGOrderNo for newly registered non-HKG orders

This view contains list of order numbers which are not registered by HKG users and meet following conditions:

* Confirmed and not-cancelled order. (Status <> 2 and Confirmed = 1)
* Not registered by HKG users (lookup table LZ\_LOD\_HKGUsers)
* Direct purchase from PC or procurement center is not one of following – ‚Rieckermann Gmbh‘, ‚Asia Technik Ltd.‘ and ‚Rieckermann Services Ltd.‘

#### View vTMP\_LOD\_OrderIntake

For Order Intake Values from LOD we use view below to get data:

* View „vTMP\_LOD\_OrderIntake“

This View is squarely based on the table „LZ\_LOD\_OrderReg“. The table is joined with three views above (vLOD\_HKGOrderNo, vLOD\_FOXPROOrderNo and vLOD\_NonHKGOrderNo) in order to transfer only eligible orders to data warehouse.

For LOD service orders, only chargeable orders are eligilble to be shifted to BOARD, non-chargeable service orders are filtered out. (See the comment in the view content below)

Backlog information is determined by ‘Shipped Date’ field on LZ\_LOD\_OrderBillingShipping table.

CREATE view vTMP\_LOD\_OrderIntake

as

select o.Industry, o.Merchandise, o.EstDeliveryDate, o.Currency, o.OrderIntakeDate

, o.GoodDescription, o.OrderVal\_EUR, o.DB2Val\_EUR, o.OrderVal, o.DB2Val, o.LocalRef

, c.CustomerID, s.SupplierID, o.GroupNo as OrderNo\_New

, cast( 'LOD' as nvarchar(5)) as source, tb.bu\_id

, case when vl.source = 'HKG' then 'J'

else case when bs.ShippedDate IS not null then 'N'

else 'J' end

end as backlogrelevant

, '1' as OrderPos

, bs.ShippedDate, LEFT(o.BusinessType, 1) as OrderType

, cast(pc.code as nvarchar(3)) as ProcurementCenter

from LZ\_LOD\_OrderReg o

left join LZ\_LOD\_Customer c on o.Customer = c.CustomerName

left join LZ\_LOD\_Supplier s on o.Supplier = s.SupplierName

join tmp\_bu tb on o.BusinessUnit = tb.bu\_desc

join ( select OrderNo, 'HKG' as source from dbo.vLOD\_HKGOrderNo union all

select OrderNo, 'HKG' from dbo.vLOD\_FOXPROOrderNo union all

select OrderNo, 'nonHKG' from dbo.vLOD\_NonHKGOrderNo

)

vl on o.GroupNo = vl.OrderNo

left join LZ\_LOD\_OrderBillingShipping bs on o.OrderNo = bs.OrderNo

left join ADM\_ProcurementCenter pc on pc.description = o.ProcCenter

-- filter out non-chargeable LOD service orders

left join LZ\_LOD\_OrderRegService ors

on o.OrderNo = ors.OrderNo and o.Merchandise = 'S' and ors.Chareable = 0

where ors.OrderNo is null

### View for OrderIntake from HKG Access orders before 2013

### **(For reference only, as HKG Access orders before 2013 are disconnected to Data warehouse)**

The view below is used to gather Order Intake Values from HKG Access orders before 2013 (not begin with “x”):

* View „vTMP\_HKGOrders\_DB2“

The view is basically pulling data from table “LZ\_HKGORders\_Orders” through another view “vTMP\_HKGOrders\_OrderValue”.

### View for OrderIntake from HKG Access orders since 2013 onwards

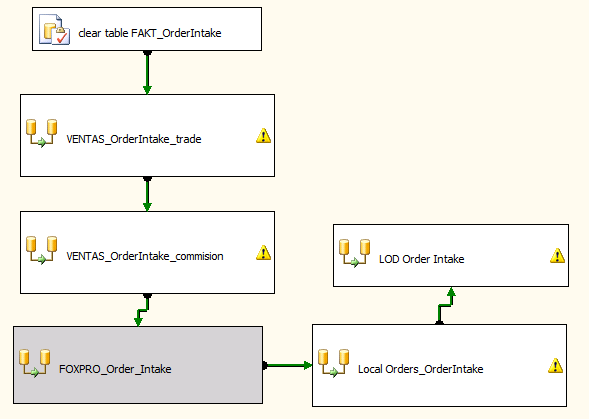
The view below is used to gather Order Intake Values from HKG Access orders since 2013 onwards (order numbers begin with “x”):

* View „vTMP\_HKGOrders\_DB2“

The view is basically pulling data from table “LZ\_HKGORders\_Orders” through another view “vTMP\_HKGOrders\_OrderValue”.

### Facttable „Fakt\_OrderIntake“

The Fact table „FAKT\_OrderIntake“ is populated by the SSIS package “FAKT\_OrderIntake”:



„SSIS package „Facttable\_OrderIntake.dtsx““

Data from the Views for Ventas Commision and Trade (see Point 5.1.1 and 5.1.2) is used to transfer Ventas orders since 2008 onwards to data warehouse. Data source of these orders are marked with “HBG.

LOD view “vTMP\_LOD\_OrderIntake” (see Point 5.1.3.4) is used to transfer LOD orders to data warehouse. Data source is named as “LOD”.

For HKG Access orders, data source is marked with “HKL”.

Facttable “HIST\_OrderIntake”

This table shows the historical data of all order intakes. When an order is changed the changed attributes or values of the order are not lost. The source of this table is the facttable “FAKT\_OrderIntake” which represents the current state of the order intake values.

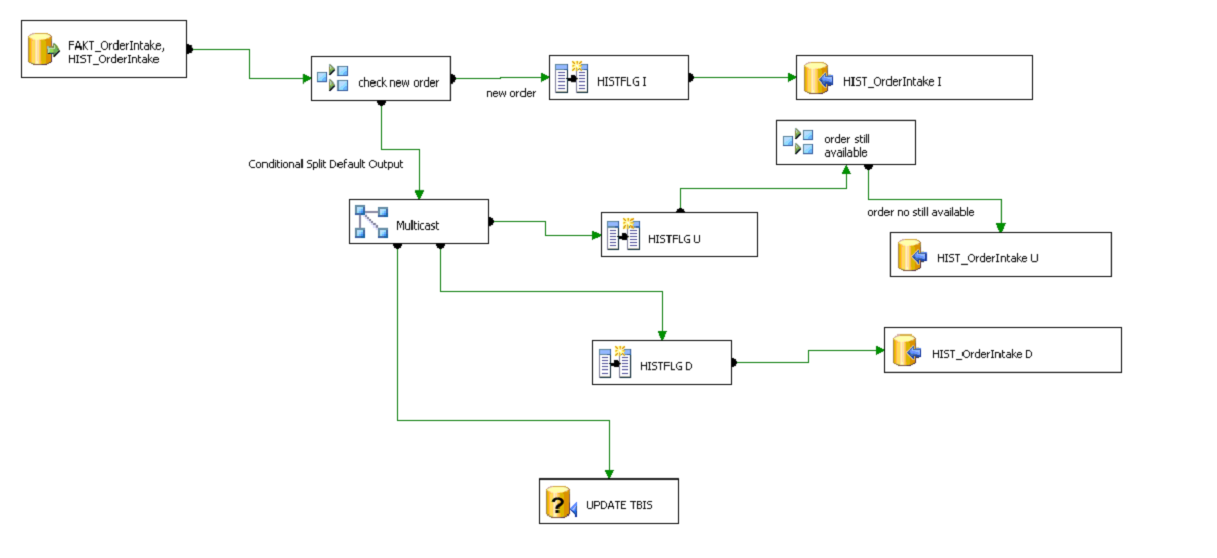
It has the same columns as the “FAKT\_OrderIntake” but three more columns:

HISTFROM Date field, represents the date since when this data row is valid

HISTTO date field, represents the date until when this data row as vald

HISTFLG flag field, represents the type of historical date (I = first inserted row, D = deleted row, U = updated row)

The table is updated with an SSIS Package, which is executed just after the update of the underlying order intake.

**Screenshot of SSIS Package “HistOrderIntake.dtsx”:**

For the calculation of the historical order intake the following procedure is implemented in the SSIS Package:

1. **Order is in FAKT\_OrderIntake, but not in HIST\_OrderIntake**  
   🡺 Order will be inserted in HIST\_OrderIntake with  
    - HISTFROM = order date  
    - HISTTO = 31.12.9999  
    - HISTFLG = “I” (insert)
2. **Order in in FAKT\_OrderIntake and in HIST\_OrderIntake and is not changed (all column values are the same)**  
   🡺 nothing needs to be done
3. **Order is in FAKT\_OrderIntale and in HIST\_OrderIntake and is changed (e.g. value, supplier, …)**  
   🡺 existing (old) Order in hist table will be updated (closed)  
    HISTTO = yesterday  
   🡺 old order is inserted in hist table with negated value and DB2  
    HISTFROM = today  
    HISTTO = 31.12.9999  
    HISTFLG = “D” deleted  
    VALUE = old VALUE \*-1  
    DB2 = old DB2 \* -1  
   🡺 new order is inserted in hist table  
    - HISTFROM = today  
    - HISTTO = 31.12.9999  
    - HISTFLG = “U” (update)

The column “HISTFROM” is used for the time assignment in BOARD. When an order is first inserted in this table this column will be initialized with the order date.

**Example determing Order history:**

Date: 01.02.2014

FAKT\_OrderIntake

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **order no** | **order date** | **supplier** | **est deliverydate** | **value** | **db2** |
| 1 | 01.02.2014 | 15 | 15.05.2014 | 400 | 20 |
| 2 | 01.02.2014 | 66 | 18.08.2014 | 300 | 5 |
| 3 | 01.02.2014 | 77 | 20.12.2014 | 200 | 9 |

HIST\_OrderIntake

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **order no** | **order date** | **supplier** | **est deliverydate** | **value** | **db2** | **HISTFROM** | **HISTTO** | HISTFLG |
| 1 | 01.02.2014 | 15 | 15.05.2014 | 400 | 20 | 01.02.2014 | 31.12.9999 | I |
| 2 | 01.02.2014 | 66 | 18.08.2014 | 300 | 5 | 01.02.2014 | 31.12.9999 | I |
| 3 | 01.02.2014 | 77 | 20.12.2014 | 200 | 9 | 01.02.2014 | 31.12.9999 | I |

Date 02.02.2014

FAKT\_OrderIntake

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **order no** | **order date** | **supplier** | **est deliverydate** | **value** | **db2** |
| 1 | 01.02.2014 | 15 | 15.05.2014 | 400 | 20 |
| 2 | **05.02.2014** | 66 | 18.08.2014 | **200** | **4** |
| 3 | 01.02.2014 | **66** | 20.12.2014 | 200 | 9 |
| 4 | 02.02.2014 | 77 | 16.07.2014 | 800 | 8 |
| 5 | 02.02.2014 | 99 | 8.11.2014 | 888 | 99 |

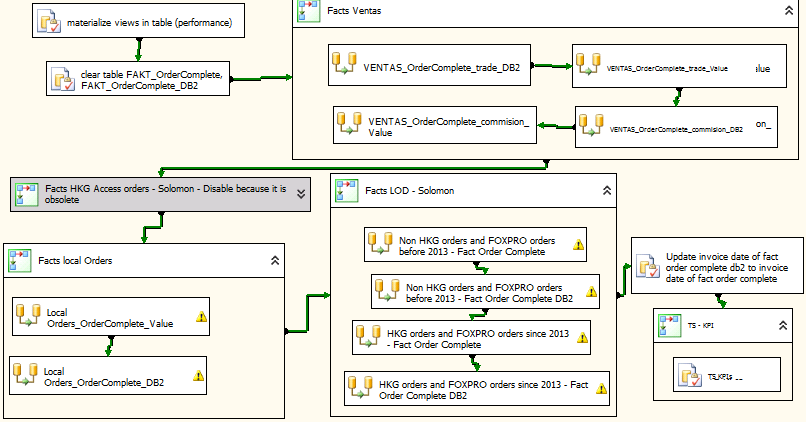
HIST\_OrderIntake

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **order no** | **order date** | **supplier** | **est deliverydate** | **value** | **db2** | **HISTFROM** | **HISTTO** | **HISTFLG** |
| 1 | 01.02.2014 | 15 | 15.05.2014 | 400 | 20 | 01.02.2014 | 31.12.9999 | I |
| 2 | 01.02.2014 | 66 | 18.08.2014 | 300 | 5 | 01.02.2014 | **01.02.2014** | I |
| 2 | 01.02.2014 | 66 | 18.08.2014 | **-300** | **-5** | **02.02.2014** | 31.12.9999 | **D** |
| 2 | **05.02.2014** | 66 | 18.08.2014 | 200 | 4 | **02.02.2014** | 31.12.9999 | **U** |
| 3 | 01.02.2014 | 77 | 20.12.2014 | 200 | 9 | 01.02.2014 | **01.02.2014** | I |
| 3 | 01.02.2014 | 77 | 20.12.2014 | -200 | -9 | **02.02.2014** | 31.12.9999 | D |
| 3 | 01.02.2014 | **66** | 20.12.2014 | 200 | 9 | **02.02.2014** | 31.12.9999 | I |
| 4 | 02.02.2014 | 77 | 16.07.2014 | 800 | 8 | 02.02.2014 | 31.12.9999 | I |
| 5 | 02.02.2014 | 99 | 8.11.2014 | 888 | 99 | 02.02.2014 | 31.12.9999 | I |

**Attention:**  
  
If an order is in the hist table and not in the actual table anymore, a “D” row is inserted but no “U” row. That means the order is eliminated, when you look at the kumulated data. That also means, when the query logic for the actual data is changed and some orders are filtered out (e.g. steel orders) these orders must be deleted from the hist table manually!

Order Complete

### The fact tables “Fakt\_OrderComplete” and “Fakt\_OrderComplete\_DB2” are populated by the SSIS package “Facttable\_OrderComplete.dtsx”.

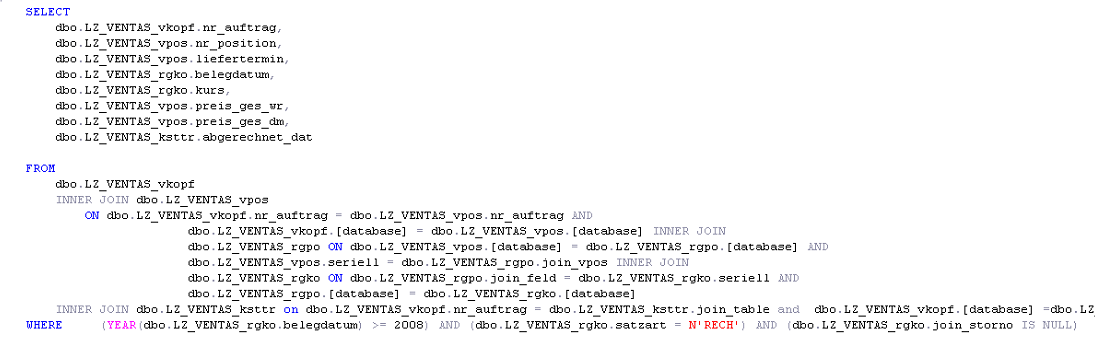


*The SSIS package “Facttable\_OrderComplete.dtsx”.* ***(Note: disable data flows are highlighted in “gray”)***

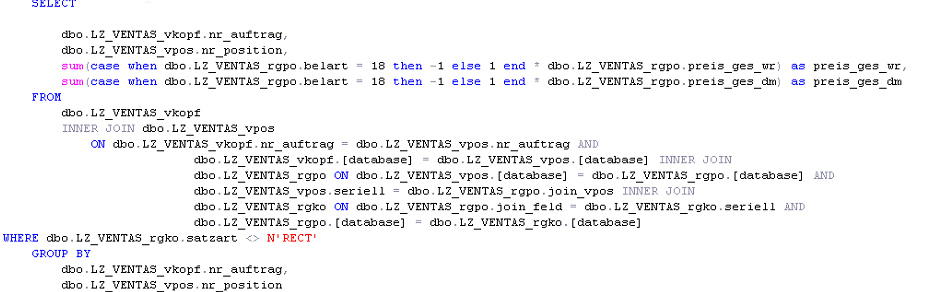
### View Order Complete for Trade orders in Ventas

To determin the Order Complete Trade value the view **“vTMP\_VENTAS\_OrderComplete\_Trade\_Value”** is used. The underlying SQL consists of two SELECT statements, which are joined together (inline view).

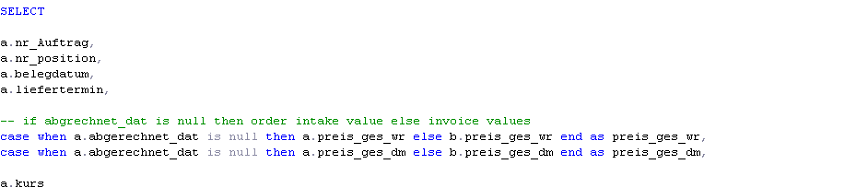
The first inline view gets the first invoice for getting the right order complete date (“Satzart” = RECH) and determins the Order Intake Value per order position.



The second view adds all others invoice value per order position where “Satzart” <> “RECT”. Invoice positions with type “18” are negated.



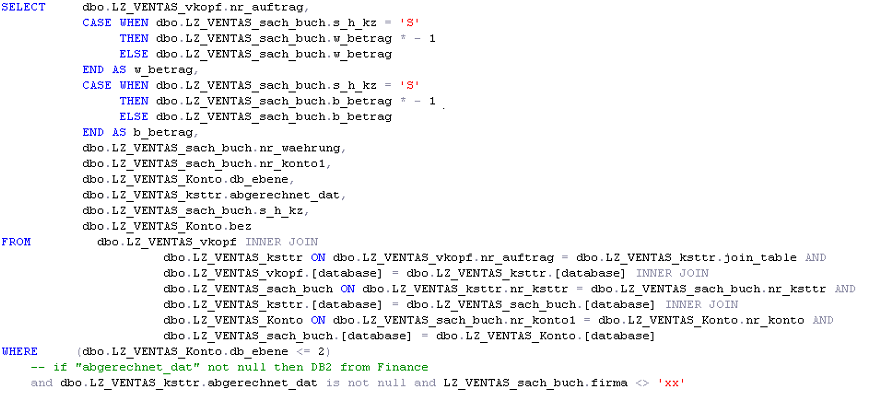
The resulting SELECT combines both views. If the Order is closed (“abgerechnet\_dat” has a value), all Invoice values are used for order complete value. Otherwise Order intake values are used.



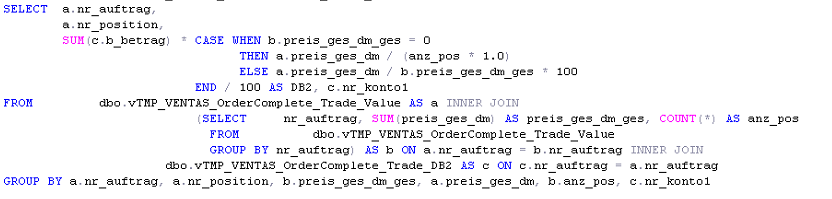
For the determination of Order complete Trade DB2 the two following views are used:

* vTMP\_VENTAS\_OrderComplete\_Trade\_DB2
* vTMP\_VENTAS\_OrderComplete\_Trade\_DB2pos

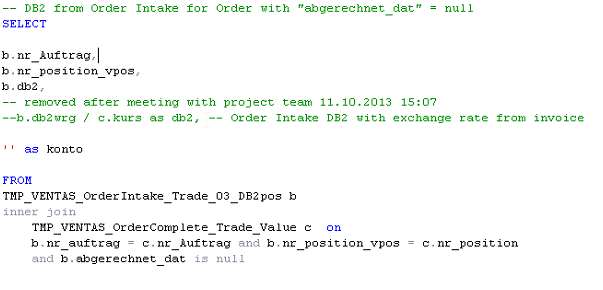
The first view gets all DB2 relevant finance bookings for all order. which are closed (“abgerechnet\_dat” has a value):



The second, resulting view consists of two SELECT statements which are combined (union all). The first SELECT statement determins the DB2 value from **vTMP\_VENTAS\_OrderComplete\_Trade\_DB2** per Order position on basis of their Trade value.



The second SELECT statement determins the DB2 order intake value for all orders which are not closed.



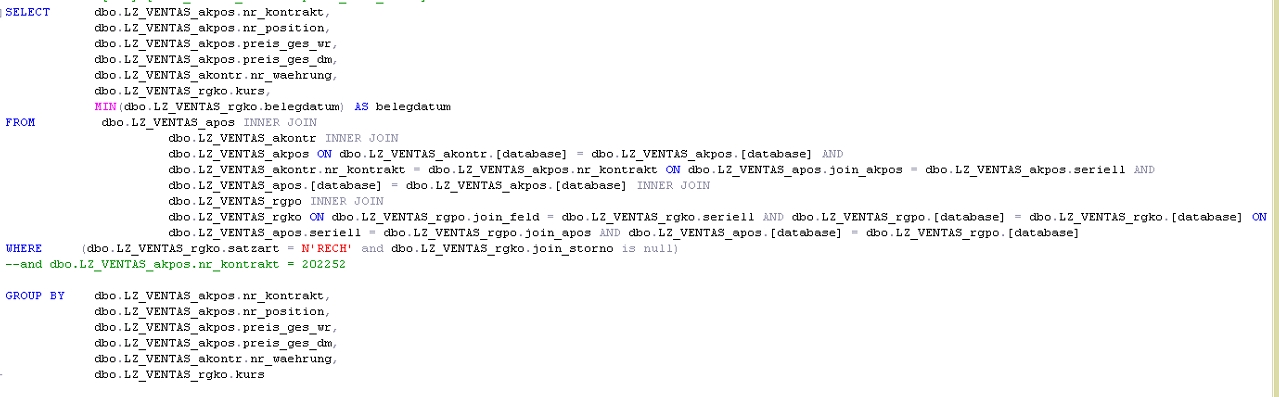
### View Order Complete for Commision from Ventas

To get the Order Complete Data for Commision Orders from Ventas we need the following Views:

* vTMP\_VENTAS\_OrderComplete\_Comm\_Value
* vTMP\_VENTAS\_OrderComplete\_Comm\_DB2
* vTMP\_VENTAS\_OrderComplete\_Comm\_DB2pos

In the first View **„vTMP\_VENTAS\_OrderComplete\_Comm\_Value“** we find the Order Complete values. In this View, all relevant Order Complete Commision tables (see Point 4.4.2) were joint together. The Invoice Date in this View is the minimal voucher date. To get the Order Complete Value we use the field “preis\_ges\_dm” from the table “LZ\_VENTAS\_akpos”. The Values in this View are filtered as followed:

* record type = RECH
* field Join\_storno is null (no canceled orders)

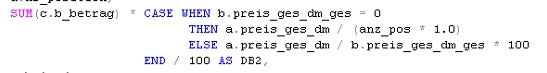


„select statement of View „vTMP\_VENTAS\_OrderComplete\_Comm\_Value“

To get the commission order DB2 values from Ventas we use the View **„vTMP\_VENTAS\_OrderComplete\_Comm\_DB2”.** In this view we join all Commision relevant DB2 tables to get the DB2 value on account level. The debit value is multiply by -1. The values are filtered as followed:

* DB\_level <= 2
* Company is not xx
* field „abgerechnet\_dat“ is not null (no canceled orders)

In the last View **„vTMP\_VENTAS\_OrderComplete\_Comm\_DB2pos”** we allocate the DB2 value from the View „vTMP\_VENTAS\_OrderComplete\_Comm\_DB2“ to the order positions. To get the percentage we use the propotion oft the Order Value.



„Calculation of DB2 per order position“

### View OrderComplete from HKG Access orders before 2013 – Solomon

**(For reference only, as HKG Access orders before 2013 are disconnected to Data warehouse)**

Following views are used to get the Order Complete Values from Solomon for Hong Kong Access orders before 2013:

* vTMP\_Solomon\_01\_GL\_localOrders
* vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType
* vTMP\_Solomon\_02\_OrderComplete\_value
* vTMP\_Solomon\_02\_OrderComplete\_DB2

The View **„vTMP\_Solomon\_01\_GL\_localOrders“** pulls respective DB2 relevant data for HKG Access orders by joining the main Solomon table (“LZ\_Solomon\_GLTran”) with following tables/views:

* „ADM\_SolomonDB2Account“/„ADM\_SolomonDB2Account\_SH“ : to get DB2 relevant data only. (linked by “Acct” field)
* vTMP\_HKGOrders\_OrderValue : to narrow down data for HKG Access orders only.

Due to the difference in DB2 relevant account codes and how to retrieve currency exchange rate, Solomon data is currently categorized into two groups - “Shanghai” or “non-Shanghai” based on database field. If database is “IndTechSha” or “PakTechSha” then that order comes from “Shanghai” database, otherwise the order comes from “non-Shanghai”. As a result, the view is the union of two separate queries.

The view **“vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType”** determines Order complete date of orders:

* The first “Sales” Booking (Account 4\*)

And the business Type:

* Account bookings from table ADM\_SolomonCommAccount/ ADM\_SolomonCommAccount\_SH are commission, otherwise trade.

The View **„vTMP\_Solomon\_02\_OrderComplete\_value“** shows how to calculate the Order Complete Value for Trade and Commision Orders. It is joined with view **“vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType”** to determine orders are trade or commission. For trade orders, the order complete value comes from Solomon, and it is retrieved from HKG Access source itself (vTMP\_HKGOrders\_OrderValue) if commission orders.

The last view in this section “**vTMP\_Solomon\_02\_OrderComplete\_DB2**” calculates sum of db2 values by each pair of an order and its respective accounts.

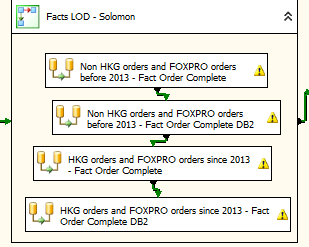
### View OrderComplete from HKG Access orders since 2013 onwards

Turnover information of these orders are directly retrieved from the view „vTMP\_HKGOrders\_DB2“, with „Shipment\_Date“ determines turnover status.

### Order Complete from LOD

LOD turnover information are retrieved from two sources (Solomon and LOD itself) as illustrated below:

* For newly registered HKG orders and historical FOXPRO orders since 2013 onwards (Order Intake!), the ETL process look up Solomon to get turnover information.
* For non-HKG orders and historical FOXPRO orders before 2013, turnover information come from LOD itself.



As aforementioned explanation, two first data flows on the top are used to populate turnover fact tables from non-HKG orders and FOXPRO orders before 2013, the queries for these two data flows just pull data from LOD landing zone tables/views.

Following views are used to get the Order Complete Values and turnover information from Solomon for newly registered HKG orders and FOXPRO orders since 2013 onwards:

* vTMP\_Solomon\_01\_GL\_New
* vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType\_New
* vTMP\_Solomon\_02\_OrderComplete\_value\_New
* vTMP\_Solomon\_02\_OrderComplete\_DB2\_New

The View **„vTMP\_Solomon\_01\_GL\_New“** pulls respective DB2 relevant data for LOD orders by joining the main Solomon table (“LZ\_Solomon\_GLTran”) with following tables/views:

* „ADM\_SolomonDB2Account“/„ADM\_SolomonDB2Account\_SH“ : to get DB2 relevant data only. (linked by “Acct” field)
* vLOD\_HKGOrderNo and vLOD\_FOXPROOrderNo: lookup Solomon to retrieve turnover figures is applied to Hong Kong orders in LOD only (including newly registered ones and old FOXPRO orders).

Similar to above, the view is the union of two separate queries due to classifying Solomon data into “Shanghai” and “non-Shanghai” sorts.

The view **“vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType\_New”** determines Order complete date of orders based on view **„vTMP\_Solomon\_01\_GL\_New“**:

* The first “Sales” Booking (Account 4\*)

And the business Type:

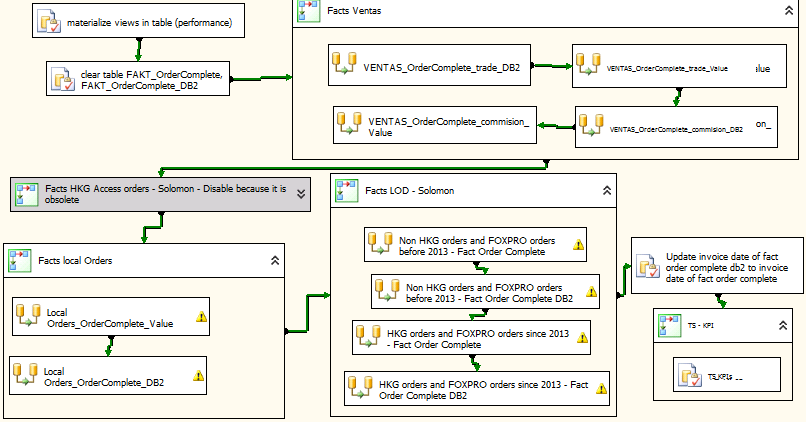
* Account bookings from table ADM\_SolomonCommAccount/ ADM\_SolomonCommAccount\_SH are commission, otherwise trade.

The View **„vTMP\_Solomon\_02\_OrderComplete\_value\_new“** shows how to calculate the Order Complete Value for Trade and Commision Orders. It is joined with view **“vTMP\_Solomon\_03\_OrderComplete\_Date\_BusinessType\_new”** to determine orders are trade or commission. For trade orders, the order complete value comes from Solomon, and it is retrieved from LOD source itself (pre-calculation figures) if commission orders.

The last view in this section “**vTMP\_Solomon\_02\_OrderComplete\_DB2\_new**” calculates sum of db2 values by each pair of an order and its respective accounts.

### Loading Factables “FAKT\_OrderComplete” and “FAKT\_OrderComplete\_DB2”

The Facttables „FAKT\_OrderComplete“ and „FAKT\_OrderComplete\_DB2“ are filled with the SSIS Package “FAKT\_OrderComplete.dtsx” (see Point 4.6.2):



*The SSIS package “Facttable\_OrderComplete.dtsx”.* ***(Note: disable data flows are highlighted in “gray”)***

To populate Fact tables

To get the Ventas OrderComplete data in our Facttable we join our Views (for Values „vTMP\_VENTAS\_OrderComplete\_Comm\_Value“ and for DB2 „vTMP\_VENTAS\_OrderComplete\_Comm\_DB2pos”) with the Facttable „FAKT\_OrderIntake“ so we can be sure that only Orders, that have an OrderIntake Value are used in our OrderComplete Facttable. Thereby we can get all additional informations (e.g. BU, Industry, Merchandise etc.) for our OrderComplete data.

The same logic is used for LOD/HKG Access orders to retrieve OrderComplete data.

Order Backlog

Order backlog actual (shipment)

The actual order backlog is calculated in the SQL View “vFAKT\_OrderBacklog\_Actual”. All orders from the fact table “FAKT\_OrderIntake” which are not completed (= not in fact table “FAKT\_OrderComplete”) will be selected as order back log

Order Backlog Stock per month

The order backlog stock value per month is calculated in the SQL-view “vFAKT\_OrderBacklog\_StockValue”. This view retriews the Order intake data from the “FAKT\_OrderIntake” and the invoice Date, if available, from the “FAKT\_OrderComplete”.

An order is in a selected month in backlog stock, when the following two condition are true:

1. Orderdate month <= selected month
2. Previous month of invoice date (=order complete date) >= selected month OR no

Example: Stock value February 2014

|  |  |  |  |
| --- | --- | --- | --- |
| **Order no** | **Order date** | **Invoice date** | **Stock value Feb/14?** |
| 1 | 01.01.2014 | - | **Yes**, orderdate month <= selected month and no invoice date |
| 2 | 01.01.2014 | 01.02.2014 | **No**, previous month of invoicedate = January, so < selected month  stock value only until january |
| 3 | 15.02.2014 | - | **Yes**, order date month <= selected monthand no invoice date |
| 4 | 15.02.2014 | 25.02.2014 | **No**, previous month of invoicedate = January, so < selected month  this order is never in backlog stock because it is process in february |
| 5 | 15.02.2014 | 15.03.2014 | **Yes**, orderdate month <= selected month and previous month of invoicedate >= selected month |
| 6 | 01.03.2014 | - | **No**, orderdate month > selected month |

Order Backlog Stock Historical per month

The order backlog per month with historical order intake data will be calculated like normale order backlog stock but with the historical fact table HIST\_OrderIntake als data source.

## TS KPIs (Technical Service KPIs)

Introduction

Due to the needs of analysis of technical services customers and technical service orders, additional KPIs are developed to provide an insight into these information.

Some terminologies:

* TS customer (technical service customer) is defined as a customer who buy spare parts or services from Rieckermann.
* TS order (technical service order) is defined as an order in which a customer buy spare parts or services from Rieckermann.
* JR machinery TS customer is defined as a TS customer who bought machinery/periphery before his first transaction buying spare parts/services.
* Non-JR Machinery TS customer is defined as a TS customer who didn’t buy any machinery/periphery before his first transaction buying spare parts/services.

Entities

* JR Customer Type

This entity indicates whether or not a new TS customer is JR machinery or non-JR machinery.

* Service Type

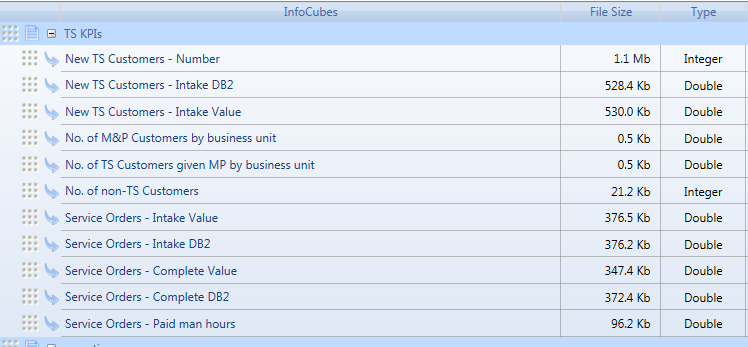
This entity indicates whether a service order is type of external or internal.

* Chargeable

This entity indicates whether a service order is chargeable or non-charegable.

Cubes

Additional cubes are developed to reflect TS-KPIs.



* New TS Customers – Number

The cube stores list of new TS customers by month, business unit and JR Customer Type.

Comprises of following entities: Month, Business Unit, Customer and JR Customer Type.

* New TS Customers – Intake DB2

The cube stores total intake db2 value of new TS customers, grouping by month, business unit, industry and JR Customer Type

Comprises of following entities: Month, Business Unit, Industry and JR Customer Type.

* New TS Customers – Intake Value

The cube stores total order intake value of new TS customers, grouping by month, business unit, industry and JR Customer Type

Comprises of following entities: Month, Business Unit, Industry and JR Customer Type.

* No. of M&P Customers by business unit

The cube contains total number of customers who buy machinery or periphery per business unit.

Involved entity: Business Unit.

* No. of TS Customers given MP by business unit

The cube contains total number of customers who buy both machinery/periphery and spare parts/service per business unit.

Involved entity: Business Unit.

* No. of non-TS Customers

This cube stores list of customers who buy machinery/periphery without any spare parts/service, per business unit.

Involved entities: Business Unit, Customer.

* Service Orders – Intake Value

This cube stores total order intake value of technical service orders, grouping by month, business unit, industry, service type and chargeable.

Involved entities: Month, Business Unit, Industry, Service Type, Chargeable.

* Service Orders – Intake DB2

This cube stores total intake db2 value of technical service orders, grouping by month, business unit, industry, service type and chargeable.

Involved entities: Month, Business Unit, Industry, Service Type, Chargeable.

* Service Orders – Complete Value

This cube stores total order complete value of technical service orders, grouping by month, business unit, industry, service type and chargeable.

Involved entities: Month, Business Unit, Industry, Service Type, Chargeable.

* Service Ordres – Complete DB2

This cube stores total order complete DB2 value of technical service orders, grouping by month, business unit, industry, service type and chargeable.

Involved entities: Month, Business Unit, Industry, Service Type, Chargeable.

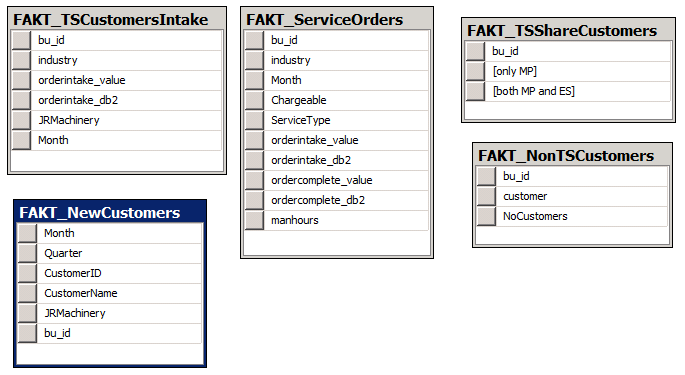
* Service Orders – Paid man hours

This cube stores total man hours (post-calculated) of technical service orders, grouping by month, business unit, industry, service type and chargeable.

Involved entities: Month, Business Unit, Industry, Service Type, Chargeable.

Data warehouse and ETL process

Extra fact tables are created to populate respective cubes in BOARD for TS-KPIs as the picture below.



Tables diagram for TS-KPIs fact tables.

* FAKT\_NewCustomers

This table stores list of new TS customers, including information like: JR Machinery or not, business unit, time when a customer become new TS customer.

Populate cubes:

* New TS Custmers - Number
* FAKT\_TSCustomersIntake

This table includes total of order intake value and order intake db2 of all new TS customers, group by business unit, industry, month and JR customer type. Order intake value and order intake db2 are summed up.

Populate cubes:

* New TS Customers – Intake DB2
* New TS Customers – Intake Value
* FAKT\_TSShareCustomers

This table counts the number of customers who buy both machinery/periphery and spare parts/service, and the number of customers who buy machinery/periphery without any spare parts/service, per business unit.

Populate cubes:

* No. of M&P Customers by business unit
* No. of TS Customers given MP by business unit
* FAKT\_NonTSCustomers

This table stores list of specific customers who buy only machinery/periphery without any spare parts/service.

Populate cubes:

* No. of non-TS customers
* FAKT\_ServiceOrders

This table stores turnover information of technical service orders, including service type, chargeable and man hours information.

SQL – Script to populate these fact tables in DWH

-- retrieve list of new TS customers (both JR and non-JR machinery) per month, per BU

truncate table dbo.FAKT\_NewCustomers

insert into dbo.FAKT\_NewCustomers(Month, Quarter, CustomerID, CustomerName, JRMachinery, bu\_id)

select distinct newTSCust.Month, newTSCust.Quarter, newTSCust.customer, null

, case when fc.orderno is not null then 1 else 0 end as JRMachinery

, newTSCust.bu\_id

from

(

select distinct LEFT(CONVERT(VARCHAR(8), min(f.Orderdate), 112), 6) as Month

, + cast(YEAR(MIN(f.orderdate)) as varchar(4)) + '0' + cast(DATEPART(QQ, min(f.Orderdate)) as varchar(1)) as Quarter

, f.customer, min(f.orderdate) as OrderDate

, f.bu\_id

from FAKT\_OrderIntake f

where f.merchandise in ('E', 'S') -- only count technical services customer: spare parts or services

and not exists ( select 1 from FAKT\_OrderIntake f1

where f.customer = f1.customer and f1.merchandise in ('E', 'S')

and f.bu\_id = f1.bu\_id

and -- make sure this TS customer did not buy services before

DATEDIFF(SECOND, f1.orderdate, f.orderdate) > 0

)

group by f.customer, f.bu\_id

) newTSCust

left join FAKT\_OrderIntake fc on fc.customer = newTSCust.customer

and fc.bu\_id = newTSCust.bu\_id

and fc.merchandise in ('M', 'P')

and newTSCust.orderdate >= fc.orderdate

-- Intake figures of new TS customers by business unit, industry

truncate table FAKT\_TSCustomersIntake

insert into FAKT\_TSCustomersIntake(bu\_id, industry, Month, JRMachinery, orderintake\_db2

, orderintake\_value)

select fn.bu\_id,industry, MONTH, JRMachinery, SUM(orderintake\_db2), SUM(orderintake\_value)

from FAKT\_NewCustomers fn join FAKT\_OrderIntake fo on fn.CustomerID = fo.customer

where YEAR(fo.orderdate) = fn.Month/100 and MONTH(fo.orderdate) = fn.Month%100

group by fn.bu\_id,industry, MONTH, JRMachinery

-- calculate Share of TS customers within existing machine customers per period

truncate table FAKT\_TSShareCustomers

insert into FAKT\_TSShareCustomers(bu\_id, [both MP and ES], [only MP])

select distinct bu.bu\_id, isnull(both.NoCustomer, 0) as [both MP and ES]

, isnull(one.NoCustomer, 0) as [only MP]

from

(

select count(distinct customer) as NoCustomer, bu\_id

from FAKT\_OrderIntake fs

where merchandise in ('M', 'P')

group by bu\_id

) one left join

(

select count(distinct customer) as NoCustomer, bu\_id

from FAKT\_OrderIntake fs

where merchandise in ('M', 'P')

and exists (select 1 from FAKT\_OrderIntake fm

where fs.customer = fm.customer and fm.merchandise in ('E', 'S')

)

group by bu\_id

) both on one.bu\_id = both.bu\_id

join tmp\_bu bu on one.bu\_id = bu.bu\_id

--- populate list of non-TS customers given by MP by business unit

truncate table FAKT\_NonTSCustomers

insert into FAKT\_NonTSCustomers(bu\_id, customer, NoCustomers)

select distinct bu\_id, customer, 1

from FAKT\_OrderIntake fs

where merchandise in ('M', 'P')

and not exists (select 1 from FAKT\_OrderIntake fm

where fs.customer = fm.customer and fm.merchandise in ('E', 'S')

)

--- populate FAKT\_ServiceOrders

truncate table FAKT\_ServiceOrders

insert into FAKT\_ServiceOrders(bu\_id, industry, month, chargeable, servicetype,

[orderintake\_value],

[orderintake\_db2],

ordercomplete\_value,

ordercomplete\_db2, manhours)

select f.bu\_id, f.industry, LEFT(CONVERT(VARCHAR(8), f.Orderdate, 112), 6)

, ISNULL(l.Chareable,0)

, case when l.ServiceType = '' then NULL else l.ServiceType end as ServiceType

, f.orderintake\_value, f.orderintake\_db2

, fc.ordercomplete\_value, fdb.ordercomplete\_db2

, l.ManHoursPost

from FAKT\_OrderIntake f

join FAKT\_OrderComplete fc on f.orderno = fc.orderno

join FAKT\_OrderComplete\_DB2 fdb on f.orderno = fdb.orderno

join LZ\_LOD\_OrderReg o on f.orderno = o.GroupNo

join LZ\_LOD\_OrderRegService l on l.OrderNo = o.OrderNo

where f.Merchandise in ('E', 'S')

1. Loading data in Board

For uploading the data in Board we load all data from the following Facttables in Board:

* FAKT\_OrderIntake
* FAKT\_OrderComplete
* FAKT\_OrderComplete\_DB2
* HIST\_OrderIntake

When we load the data in BOARD, we make no transformations. The data in Board are like the data in the fact tables.

To refresh all Cubes and Entitiys we use the Procedure “Load all” which you can find under the Database procedures.

1. Automated data load prozesses

Load Capsule in BOARD

To execute the BOARD procedures for loading the data from staging, the following capsule is used:

**Capsules\LF\DataReader.cpsx**

The capsule includes two screens with acticated screen triggers, for the import of the two databases:

[LOAD] RIS 🡺 load RIS Database

[LOAD] Sales 🡺 load Sales Database

linkFISH ETL Starter

The linkFISH ETL Startet is a small Tool which executes SSIS Packages and BOARD procedures (per screen trigger). After the execution of each process a status e-mail is send to the recipients defined in the configuration.

The tool is located in “D:\LF ETL Starter” and consists of two exe-Files:

LF\_BOARD\_Starter.exe 🡺 start BOARD procedures

LF\_SSIS\_Starter.exe 🡺 start SSIS Package

The configuration of the data load process is defined in tables in a SQL Server Database. Per exe-File the connection string for the database must be stored in an XML file with the extension “config”

**Example “LF\_SSIS\_Starter.exe.config”:**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<configSections>

</configSections>

<connectionStrings>

<add name="LF\_ETL\_Starter\_Lib.My.MySettings.LF\_ETL\_StarterConnectionString"

**connectionString="Data Source=localhost;Initial Catalog=Staging;Integrated Security=True"**

providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

The configuration consists of the four SQL Server tables:

**LFREPO\_Konfiguration**

This table contains the main configuration (e.g. BOARD directory, e-mail server, BOARD client path…):





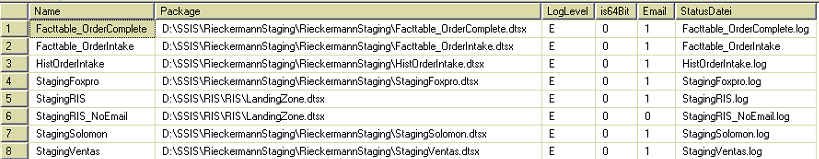
**LFREPO\_BOARD\_Prozedur**

The capsule and screens for loading the data is located in this table:



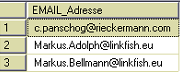
**LFREPO\_SSIS**

This table contains all SSIS Packages for loading the staging area:



**LFREPO\_Email\_Empfaenger**

The recipients for the status e-mails are stored in this table



Scheduled Tasks

There are two scheduled Tasks (Windows Task Scheduler) which triggers the two data processes:

|  |  |  |
| --- | --- | --- |
| **Name** | **Trigger** | **File to execute** |
| BOARD RIS | 20:00 every day except Saturday | D:\LF ETL Starter\RIS.cmd |
| BOARD Sales | 20:00 every day except Saturday | D:\LF ETL Starter\Sales.cmd |

The cmd files executes the ETL Starter with the wanted load process:

**RIS.cmd**

set BOARDSTARTER="D:\LF ETL Starter\LF\_BOARD\_Starter.exe"

set SSISSTARTER="D:\LF ETL Starter\LF\_SSIS\_Starter.exe"

%SSISSTARTER% StagingRIS

%BOARDSTARTER% RIS

**Sales.cmd**

set BOARDSTARTER="D:\LF ETL Starter\LF\_BOARD\_Starter.exe"

set SSISSTARTER="D:\LF ETL Starter\LF\_SSIS\_Starter.exe"

%SSISSTARTER% StagingFoxpro

%SSISSTARTER% StagingVentas

%SSISSTARTER% StagingSolomon

%SSISSTARTER% Facttable\_OrderIntake

%SSISSTARTER% Facttable\_OrderComplete

%SSISSTARTER% HistOrderIntake

%BOARDSTARTER% Sales

1. Multidimensional Database (BOARD)

The following cubes are needed for reporting and analysis:

* order intake value actual
* order intake DB2 actual
* turnover value actual
* turnover DB2 actual
* backlog value (stock value)
* backlog DB2 (stock value)
* backlog value (delivery date)
* backlog DB2 (delivery date)
* order intake value plan
* order intake DB2 plan

For every cube the following dimensions are available:

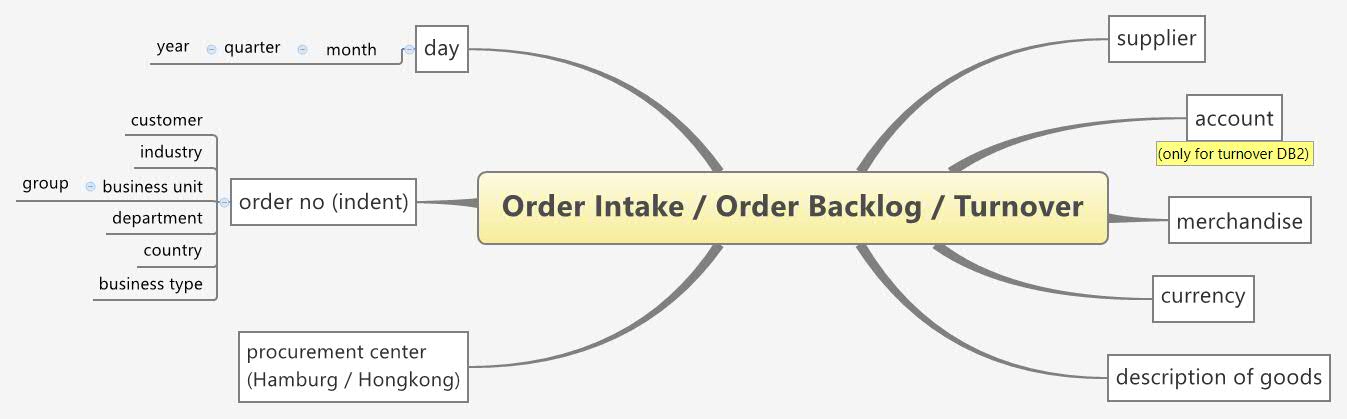


Figure 9: Data model of multidimensional database in BOARD

1. Authorization Concept

User license profiles of BOARD - an overview

Four different types of licenses exist, each providing a different set of functionality:

* + - Developer : this is the highest level license, it allows you to create and edit Board databases a capsules without restrictions. It is intended for Board administrators, developers and power users.
    - Power User : this license provides unrestricted access to Capsules both for navigation and for development purposes but does not allow to access any database design feature. This license is for power users that do not need to modify the design of Board databases.
    - LitePlus : this license provides access to the Capsule environment and all navigation features but does not allow to design or modify a Capsule; access to the Capsule *design* mode is prohibited. It is intended for users that only need to open existing Capsules, including budgeting and simulation applications, that have been designed for them by a user having a Developer or Power User license.
    - Lite : this license provides the same functions as the LitePlus license with the exception that data-entry is prohibited. It is essentially a ”read-only” license suitable for users that need to navigate existing Capsules with reports, charts or any other data-presentation object but who can't enter data.

Each of the above user licenses types can be upgraded with the MS-Office Add-in license which allows to use the Board Add-in for Microsoft Office 2007 or 2010.

Rieckermann owns licenses of all above named license profiles.

Authorization Concepts within capsules and databases

Three different authorization concepts are available within customer specific BOARD capsules and databases:

1. Access rights for Portal Areas (a set of reports)
2. Access rights for individual reports or data entry masks (single screens)
3. Access rights for individual database entities (e.g. business unit ‘china machinery’)

All three authorization concepts will be implemented within the Rieckermann BOARD solution. The administration of the authorization rights is only possible for BOARD users with the license type ‘Developer’. Currently (8.8.2013) Rieckermann owns one developer license. Controlling in Hamburg is responsible for the administration of the authorization concept and maintenance and obtains the ‘Developer’-License.

How to create and maintain user profiles

Chapter 5 ‘User accounts and security’ on <http://help.board.com/> describes how to create and maintain user profiles.

Current Authorization Matrix

Next to the user setting in BOARD the below illustrated authorization matrix is established to manage the authorization rights. Please see attachment 6.4 for details of the authorization matrix.



Figure 10: Authorization matrix as of August 13th 2013

Next to the above illustrated authorization matrix in Excel, the project team will implement an authorization report in BOARD to track which user obtains which rights for screens and portal areas (user in rows, screens in columns).

1. Attachments

Detailed specification of Reports and Data entry masks

Report and budget specification Rieckermann.xls

Detailed specification of portal layout and navigation

Navigation\_structure BOARD.xls

Detailed specification of authorizations per BOARD user

Rieckermann Berechtigungsmatrix BOARD.xls

List of Definitions

Defintions (1) PST 2013-08-01.xls