

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 1 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------



## IT-Concept for Bond MCC

Jens Richelsen (DUS - 9352)

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 1	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 2 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

## Revision log

Date	Name	Description
06.12.2004	Richelsen	First Draft – Version 0.1
02.05.2005	Richelsen	Additional trade conditions – Version 1.0
29.11.2005	Richelsen	Additional trade conditions – Version 1.1
15.12.2006	Richelsen	Additional trade conditions – Version 1.2
18.04.2007	Richelsen	Additional mandant – Version 1.3
23.05.2007	Richelsen	Additional trade conditions – Version 1.4
24.10.2007	Richelsen	Additional trade conditions – Version 1.5
11.03.2010	Richelsen	New Front OfficeLoanIQ– Version 1.6
10.09.2010	Richelsen	Review of location filter – Version 1.7
30.12.2010	Richelsen	Review – Version 1.8
11.03.2011	Richelsen	Review – Version 1.9
04.07.2011	Richelsen	Rule to check locations
26.10.2011	Richelsen	Move of Covered Bond Bank
11.04.2012	Richelsen	Review
01.07.2012	Richelsen	Portigon
12.10.2012	Richelsen	Client selection criteria
23.08.2013	R. Steger	Removed EAA and Hong Kong clients
14.04.2014	R. Steger	Removed London client Removed initial config values
18.06.2014	R. Steger	Detailed “min”/”max” behavior for price tolerance check (section 3.2.3.27)
02.12.2014	R. Steger	Added PAG client

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 2	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 3 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

## Table of contents

1	Introduction .....	4
2	Architecture .....	4
2.1	DATA IMPORT .....	4
2.1.1	Summit.....	4
2.1.2	LoanIQ.....	7
3	MCC logic .....	8
3.1	GENERAL CLASSIFICATION MECHANISM .....	8
3.1.1	Currency category .....	9
3.1.2	Trade type category .....	9
3.1.3	Price thresholds .....	9
3.2	TRADE STATUS CHECKS .....	10
3.2.1	General .....	10
3.2.2	Definition of trade status rules .....	10
3.2.3	Definition of the conditions .....	11
3.3	PRICE REQUEST.....	15
3.3.1	Building a Bloomberg request .....	15
3.3.2	Data retrieval from Bloomberg.....	15

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 3	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 4 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

## 1 Introduction

The market conformity check (MCC) on a daily basis is a central requirement of the MaH's. This document covers details of the extraction and the processing of Bond-products from Summit.

## 2 Architecture

### 2.1 Data import

#### 2.1.1 *Summit*

##### 2.1.1.1 Selection criteria

The extraction of the trades from Summit is defined by the Summit filter DUTC\_MCC\_BO.

##### 2.1.1.2 Client selection criteria

The further processing of the trade depends on the trader location. Using the trader location of book of the trade is looked up in PARIS.

Client (mandant name)	PARIS trader location	Location name
Summit Bond London (SML) (no longer active)		
Summit Bond Luxemburg (SMX) (no longer active)		
Summit Bond Asia (SMA) (no longer active)		
Summit Bond Hong Kong (SMH) (no longer active)		
Summit Bond EAA (SME) (no longer active)		
Summit Bond PAG (SMG)	21	London
	38	Shanghai
	26	Hong Kong
	56	Singapore
	48	Madrid
	021	London
	026	Hong Kong
	038	Shanghai
	048	Madrid
	056	Singapore

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 4	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 5 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

	001	Duesseldorf
	022	New York
	023	Tokyo
	024	Tokyo
	027	Tokyo
	041	Sydney
	081	Luxemburg
	082	Singapore
	084	London
	087	Singapore
	406	London
	454	Johannesburg
	617	Luxemburg
	804	Dublin (WLB Ireland)
	806	Budapest
	807	Istanbul
	812	Madrid
	817	Moscow
	818	Warsaw
	828	Milan
	831	Milan
	833	Paris
	842	London
	861	New York
	862	Tokyo
	866	London
	869	Toronto
	871	London
	874	New York
	876	Sao Paulo
	878	New York
	888	London
	898	Singapore
	899	Singapore
	999	
	VBB	Verbundbank preparation
	xxx	
	Tokyo	Tokyo
	[n/a]	
	Sydney	Sydney
	Hongkong	Hong Kong
	Hong Kong	Hong Kong
	Singapore	Singapore
	EAA2	EAA Düsseldorf refill preparation
	EAA2_NYC	New York

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 5	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 6 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

	EAA2_HKG	EAA Hong Kong refill preparation
	EAA2_LON	EAA London refill preparation
	EAA2_807	EAA Istanbul refill preparation
Summit Bond Düsseldorf (SMB)	Any other	

## 2.1.1.3 Field reference

The data is extracted from Summit at 23:00 into a semicolon separated ascii file. It contains a header records with the column headers.

The last line (indicated by a #) contains as checksum the number of business records, the start and the stop time of the business interval extracted and finally the extraction timestamp.

Example:

```
#;435;25.03.2004 08:08;25.03.2004 20:16;25.03.2004 22:44
```

The file is transferred to the Department-Server via Connect:Direct and is stored at \\NTSAPPLI0010504\Daten\TradeControl\Summit\mgb\_bond\_YYYYMMDD.txt.

A single record consists of the following fields:

Nr	field name	data type	definition	example
1	Tradeld	char	Tradeld	"0000071412"
2	TradeVersion	int	Current version number of the trade	
3	Status	char	Trade status code	DONE, VER
4	BBG_Id	char	Bloomberg Identifier the trade	PP5J2PW83
5	ISIN	char	ISIN of instrument	
6	Ccy	char	Currency	USD
7	FXRate	float	Exchange rate at current currency	1.227.900
8	TradeType	char	Specifies the type of the instrument	BOND, BONDOP, LOPT, FUTURE
9	Subtype	char	Subtype of bond definition	DOM, BUND
10	Category	char	Sec classification WMBINT/CATEGORY	
11	Start day	date	Trade start day	20.08.03
12	Expire day	date	Trade expiry/settle day	28.12.07
13	InstrStartDay	date	Instrument start day	
14	InstrExpireDay	date	Instrument expiry day	
15	Issuer	char		~Europ. Hypbk. Lux.
16	Book	char	Book of the trade	UB250
17	Nominal	float	The nominal amount of the trade	54000
18	Cpty	char	Counterpart of the trade	Iserlohn ssk
19	Cpty Ref	char	Counterparty reference	f+c
20	Description	char	Description	
21	Alias	char	Is the identification number of the counterpart (needs to be delivered	SA395

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 6	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 7 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

			only if we have it)	
22	Trader	char	Identification of trader who make the trade	SCHNEIDERS
23	Trade Date	date	TradeEntryDate from envelope screen	31.12.03
24	Trade Time	time	Input time from envelope screen	20:00:00
25	Value Date	date	This is the date when the trade premium is to be paid or received	31.12.03
26	Update Time	datetime	Amend datetime from back screen	11.03.2004 10:36:37
27	Trade Price	float	Price at which the trade has been done	102.25
28	Market Price	float	Reference price from MTM application. Regardless of the quote method setting the price shall be retrieved from the market sheet	102.25
29	Theor Price	float	Theoretical price of trade	102.25
30	Updated by	char	User name from amend field in back screen	SCHNEIDERS
31	Struct	char	Optional field that describes the structure type of the security classification	CDN
32	Company	char	Company the trade was traded for	WESTLB
33	discount	char	Shows the trade's fee. Relevant for trades shown in the S-MGB	0.5
34	CustomerType	char	Trades with SPARKASSE can be identified	SPARKASSE
35	LegalName	char	Legal name of the counterpart	SPARKASSE XYZ
36	ExtNote1	char	Comment field	
37	ExtNote2	char	Comment field	
38	Trade Yield	float	Yield related to the traded price	6.56
39	WLB Yield	float	Yield related to the theoretical price	6.34
40	Theo Yield	float		6.34
41	WLB Price	float		100.12
42	Market Yield	float		6.34
43	Marketer	char		SCHNEIDR

## 2.1.2 LoanIQ

### 2.1.2.1 Client selection criteria

The LoanIQ trades are only processed in the London client.

### 2.1.2.2 Field reference

The data is extracted from LoanIQ at 01:00 into a semicolon separated ascii file. It contains a header records with the column headers.

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 7	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 8 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

The last line (indicated by a #) contains as checksum the number of business records, the start and the stop time of the business interval extracted and finally the extraction timestamp.

Example:

```
#;435;25.03.2004 08:08;25.03.2004 20:16;25.03.2004 22:44
```

A single record consists of the following fields:

Nr	field name	data type	definition	example
1	Trade ID	char	TradeId	7A8YNYDL
2	Branch	char	Part of the paris book name. The 3 trailing numbers and the number of the risk book is combined to the paris book id: e.g. 368510_866	00866
3	Risk Book	char	LoanIQ book. Part of the paris book name	368510 EM Loan Trading - EMLT
4	Deal Name	char	Name of the deal	SUEZ EUR4.5BN 10MAY05 (LT)
5	Facility Number	char	LoanIQ number of the instrument	20526
6	Facility Name	char	LoanIQ name of the instrument	REVOLVING CREDIT FACILITY
7	Buy/Sell	char	Buy/Sell flag	SELL
8	Amount	Float		20000000.00
9	Currency	Char		EUR
10	Price	Float		100.0
11	Trade Date	Date		20100305
12	Counterparty	Char		SACHSEN BANK,LEIPZIG
13	Category	Char		Loan
14	Maturity	date		20120510
15	Trader	Char		

## 3 MCC logic

In this chapter, the MCC logic is described, which is used for the data from Summit.

The LoanIQ data has no further processing or logic. All LoanIQ trades need to be checked manually.

### 3.1 General classification mechanism

A trade falls in two different categories to examine the combined tolerance class. This class defines the tolerance threshold for the trade price compared to the market price.

Structured products are treated different. If the field "struct" is filled, it is taken as category name.

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 8	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------



# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 9 of 15	Version Date 2014-06-18
---------	--	--	--------------	-----------------	----------------------------

A list of general exceptional struct-names can be defined in the database in the table T42\_MGB\_CONFIGURATION and the key WESTLB\_NON\_STRUCTURE\_TOLERANCE\_CATEGORY. They will be treated as normal trades as described below.

Additionally a list with exceptional struct-names for instruments that start with “WLB” or “WESTLB” can be defined. They will be treated as normal trades as described below.

## 3.1.1 Currency category

A currency can be mapped to a category, to reduce the number of threshold-combinations. The current mapping can be found in the table T52\_CURRNCY\_CATEGORY\_MAPPING. If no mapping is defined the category DEFAULT is chosen.

The following SQL extracts the mapping for a given client (replace ‘???’ with the mandant name from above):

```
SELECT t52_currency, fk_t52_t51_category
FROM t52_currncy_category_mapping
WHERE fk_t52_t09_mandant = '???';
```

## 3.1.2 Trade type category

The summit trades type can be mapped to a category, to reduce the number of threshold-combinations. The current mapping can be found in the table T54\_TRADETYPE\_CATEGORY\_MAPPING. If no mapping is defined the category DEFAULT is chosen.

The following SQL extracts the mapping for a given client client (replace ‘???’ with the mandant name from above):

```
SELECT t54_tradetype, fk_t54_t53_category
FROM t54_tradetype_category_mapping
WHERE fk_t54_t09_mandant = '???';
```

## 3.1.3 Price thresholds

The concatenation of trade type category and the currency category builds the combined category that can be mapped to a price tolerance threshold. If the field “Struct” is filled, it is taken as the combined category.

The following SQL extracts the mapping for a given client client (replace ‘???’ with the mandant name from above):

```
SELECT t05_instrument AS combined_category, t11_name AS price_tolerance
FROM t05_instrument, t11_price_check_category
WHERE t05_instrument_type = 'pricecheck'
AND fk_t05_t11_price_check = t11_id
AND fk_t05_t09_mandant = '???';
```

The following SQL shows the complete list of combinations for a given client client (replace ‘???’ with the mandant name from above):

```
WITH combined_mapping AS
  (SELECT t52_currency AS currency, t54_tradetype AS tradetype,
    fk_t54_t53_category || '-' || fk_t52_t51_category AS
combined_category,
    fk_t54_t09_mandant AS mandant
  FROM t52_currncy_category_mapping JOIN t54_tradetype_category_mapping
```

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 9	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	-----------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 10 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

```

        ON fk_t52_t09_mandant = fk_t54_t09_mandant)
SELECT    currency, tradetype, t05_instrument AS combined_category,
        t11_name AS price_tolerance
FROM      t05_instrument i JOIN t11_price_check_category
        ON fk_t05_t11_price_check = t11_id
        LEFT OUTER JOIN combined_mapping cm
        ON mandant = fk_t05_t09_mandant
        AND t05_instrument = cm.combined_category
WHERE     t05_instrument_type = 'pricecheck' AND fk_t05_t09_mandant = '???'
ORDER BY  currency, tradetype;

```

## 3.2 Trade status checks

### 3.2.1 General

To calculate the status of a trade a set of expressions is evaluated. If the expression returns true or if the expression is empty, the evaluation is stopped and the related state is taken as the resulting state.

An expression consists of expressions combined with the logical and-operator (&&) or the or-operator (||). An expression can also be negated with the not-operator (!) and finally braces can be used to group expressions. A nuclear expression (a condition) is then evaluated by executing a java-function of the trade.

An example:

status_name	expression
no_check	product_not_mcc_relevant    is_storno
internal_deal	is_internal
high_low_check	is_net_trade
historical_check	

In the example the first expression "product\_not\_mcc\_relevant" is extracted and the corresponding java-method is looked up in a special configuration table and executed on the trade-object. Then the next condition "is\_storno" is checked against the trade-object. The result is logically combined with the or-operator. Assuming the result is true, the final state would be "no\_check". Again assuming the first three expression return false, the forth state "historical\_check" would become the final result.

### 3.2.2 Definition of trade status rules

The following rules are used to calculate the status of the trade. The last 10 rules are executed after the Bloomberg data is received.

status_name	expression
New location	!is_expected_location
FO-Confirmed ('DONE')	is_status_done && is_current_version
Informational ('DONE')	is_status_done
No check	is_exchange_traded
Storno	is_status_canceled
Already checked an older version	!is_mcc_relevant_change
MRM Equity Product	is_equity_product
Initial Public Offering (no check)	(is_ipo    is_ipo_books) && (is_sell    (is_buy && is_price_hundred))
Initial Public Offering (check)	is_ipo    is_ipo_books
Bloomberg check required (Back to back)	is_back_to_back
Bloomberg check required (H/L)	!is_book_with_no_bloomberg_prices &&

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 10	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 11 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

	!is_category_with_no_bloomberg_prices && has_bloomberg_request_string && is_category_with_high_low_bloomberg_prices
Bloomberg check required	!is_book_with_no_bloomberg_prices && !is_category_with_no_bloomberg_prices && has_bloomberg_request_string
No reference price	has_no_theoretical_price
MCC check failed	!is_theoretical_ok && !is_theoretical_bagatelle
Bagatelle (check required)	!is_theoretical_ok && is_theoretical_bagatelle && is_late_deal && has_predecessor && !is_rebooking_book
Bagatelle	!is_theoretical_ok && is_theoretical_bagatelle
Theoretical price OK (check required)	is_late_deal && has_predecessor && !is_rebooking_book
Theoretical price OK	
Back to back (Durchhandel)	is_back_to_back && is_back_to_back_check
Back to back (Durchhandel)	is_back_to_back
No reference price	has_no_price && has_no_theoretical_price
MCC check failed	has_no_price && !is_theoretical_ok && !is_theoretical_bagatelle
Bagatelle (check required)	has_no_price && !is_theoretical_ok && is_theoretical_bagatelle && is_late_deal && has_predecessor && !is_rebooking_book
Bagatelle	has_no_price && !is_theoretical_ok && is_theoretical_bagatelle
Theoretical price OK (check required)	has_no_price && is_late_deal && has_predecessor && !is_rebooking_book
Theoretical price OK	has_no_price
Out of price	!is_bloomberg_ok && !is_bloomberg_bagatelle
Bagatelle (check required)	!is_bloomberg_ok && is_bloomberg_bagatelle && is_late_deal && has_predecessor && !is_rebooking_book
Bagatelle	!is_bloomberg_ok && is_bloomberg_bagatelle
Bloomberg OK (check required)	is_bloomberg_ok && is_late_deal && has_predecessor && !is_rebooking_book
Bloomberg OK	

**Remark:** If a price has been received from Bloomberg and successfully saved, the status of the request id set to "OK\_PRICE\_UNVALIDATED". Only if the trade status has been calculated by analyzing the price, the request status is set to "OK". If no price was found, the request is transiting from "ERROR" to "OK\_NO\_PRICE".

## 3.2.3 Definition of the conditions

### 3.2.3.1 Condition: is\_expected\_location

It checks if the trader location, that is mapped to the book is held in a list which is configured in the database in the table T42\_MGB\_CONFIGURATION and the key EXPECTED\_LOCATIONS. This list should be in sync with the values documented in section 'Client selection criteria'. If the value is in the list, the condition returns true. (The condition is exclusively used for the Düsseldorf client.)

New locations should be checked regarding their client mapping and their report location:

1. If they should be mapped to a different client, a new MGB release is needed that implements the change in the converter stage.
2. If they should be mapped to a report location, they need to be added to the T120\_REPORT\_CONFIGURATION table.

Both changes involve the IT support team.

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 11	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 12 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

## 3.2.3.2 Condition: is\_status\_done

The field "status" indicates this condition.  
If the value is "DONE", it returns true.

## 3.2.3.3 Condition: is\_current\_version

If trades in summit are saved the field version is increased and the tradeID stays unchanged.  
The record with the highest version per trade and imported file is flagged as current version.  
(Remark: this condition is unnecessary, since the extraction only delivers current versions)

## 3.2.3.4 Condition: is\_exchange\_traded

The field "alias" indicates this condition.  
If the value starts with "HD", it returns true.

## 3.2.3.5 Condition: is\_status\_canceled

The field "status" indicates this condition.  
If the value is "CANC", it returns true.

## 3.2.3.6 Condition: is\_mcc\_relevant\_change

It looks for the first trade with the same tradeID and the state "VER" within the previous 200 successfully loaded jobs in the tool. These trades are linked together, and are compared field-wise.

If such a trade exists and none of the following fields

"sourceSystemInstrument", "tradeType", "subType", "category", "currency",  
"startDay", "expireDay", "tradePrice", "tradeDate"

are different, the condition returns false.

## 3.2.3.7 Condition: is\_equity\_product

It checks if the book name is hold in a list which is configured in the database in the table T42\_MGB\_CONFIGURATION and the key EQUITY\_PRODUCTS\_BOOKS.  
If the value is in the list, the condition returns true.

## 3.2.3.8 Condition: is\_ipo

If the book name of a trade starts with 'NE' the condition returns true.

## 3.2.3.9 Condition: is\_ipo\_books

It checks if the book name is hold in a list IPO- books. The list of IPO-books is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_SUMMIT\_IPO\_BOOKS.  
If the value is in the list, the condition returns true.

## 3.2.3.10 Condition: is\_buy

Returns true if the volume is positive.

## 3.2.3.11 Condition: is\_sell

Returns true if the volume is not positive.

## 3.2.3.12 Condition: is\_price\_hundred

Returns true if the price is hundered (with an epsilon of 1E-05).

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 12	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 13 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

## 3.2.3.13 Condition: is\_back\_to\_back

Goal is to recognize back to back trades ("Durchhandelsgeschäfte"). A set of trades with the same book and the same instrument and a volume that can be added to zero, will not result in a position and is identified as back\_to\_back. Additionally the trade prices differences must stay inside a defined range and only special books are regarded.

The list of books is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOOKS\_WITH\_BACK\_TO\_BACK\_TRADES.

The maximum spread between the maximum and the minimum price of the back\_to\_back trades is configured in the database in the table T42\_MGB\_CONFIGURATION and the key MAX\_BACK\_TO\_BACK\_SPREAD.

## 3.2.3.14 Condition: is\_back\_to\_back\_check

Based on the condition is\_back\_to\_back. If that condition is true, it returns true with a probability of a given percentage.

The percentage is configured in the database in the table T42\_MGB\_CONFIGURATION and the key SAMPLE\_BACK\_TO\_BACK\_PERCENTAGE.

## 3.2.3.15 Condition: is\_category\_with\_no\_bloomberg\_prices

The field "category" indicates this condition. A list of categories, where no price is expected is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_SUMMIT\_NON\_BLOOMBERG\_CATEGORIES.

If the value is not in the list, the condition returns true.

## 3.2.3.16 Condition: is\_book\_with\_no\_bloomberg\_prices

The field "book" indicates this condition. A list of books, where no price is expected is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_SUMMIT\_NON\_BLOOMBERG\_BOOKS.

If the value is not in the list, the condition returns true.

## 3.2.3.17 Condition: is\_category\_with\_bloomberg\_high\_low\_prices

The calculated field of the combined category, that was described above indicates this condition. A list of combined categories, where high-low prices are more reasonable is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_HIGH\_LOW\_BLOOMBERG\_CATEGORIES.

If the value is not in the list, the condition returns true.

## 3.2.3.18 Condition: has\_bloomberg\_request\_string

Checks if a request string for Bloomberg has been build.

## 3.2.3.19 Condition: is\_theoretical\_ok

The theoretical price delivered by Summit is based on data which has been reviewed by the MTM-process on the previous day.

Taking this theoretical price, an interval is built by adding/subtracting a percentage that is defined in the price threshold table described above.

If the traded price is inside this interval, the condition returns true.

## 3.2.3.20 Condition: is\_theoretical\_bagatelle

To calculate the turnover the following formula is used:

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 13	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 14 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

$$\text{turnover} = \frac{(\text{marketPrice} - \text{tradePrice}) * \text{volume}}{100}$$

If the absolute turnover is smaller than a threshold, the condition returns true.

The threshold is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_BAGATELLE\_LIMIT.

The values are supposed to be in EUR.

## 3.2.3.21 Condition: is\_late\_deal

If the difference between update time and trade time is greater than 24 hours, it returns true.

## 3.2.3.22 Condition: has\_predecessor

It looks for the first trade with the same tradeID and the state "VER" within the previous 25 successfully loaded jobs in the tool.

If such a trade exists, the condition returns true.

## 3.2.3.23 Condition: is\_rebooking\_book

It checks if the book name is hold in a list which is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_REBOOKING\_BOOKS.

If the value is in the list, the condition returns true.

## 3.2.3.24 Condition: has\_no\_historic\_price

Returns true if the historic price is zero.

## 3.2.3.25 Condition: has\_no\_theoretical\_price

Returns true if the theoretical price is zero.

## 3.2.3.26 Condition: has\_no\_price

Returns true if no price could be found in Bloomberg.

## 3.2.3.27 Condition: is\_bloomberg\_ok

The theoretical price delivered by Summit is based on data which has been reviewed by the MTM-process on the previous day.

Taking the market price requested from Bloomberg, an interval is built by adding/subtracting a percentage that is defined in the price threshold table described above. (The percentage is subtracted from Bloomberg's "min" price, and added to the "max" price to form the interval.)

If the traded price is inside this interval, the condition returns true.

## 3.2.3.28 Condition: is\_bloomberg\_bagatelle

To calculate the turnover the following formula is used:

$$\text{turnover} = \frac{(\text{MID}(\text{marketPrice}) - \text{tradePrice}) * \text{volume}}{100}$$

If the marketPrice is an interval price like bid/ask or high/low, the average price is taken.

If the absolute turnover is smaller than a threshold, the condition returns true.

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 14	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------

# Portigon Documentation

Project	Document Title Bond Market Conformity –IT Concept	Filename IT_Concept_Bond_MCC_V2.5 .doc	Section 1	Page 15 of 15	Version Date 2014-06-18
---------	--	--	--------------	------------------	----------------------------

The threshold is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_BAGATELLE\_LIMIT.

The values are supposed to be in EUR.

## 3.3 Price request

### 3.3.1 Building a Bloomberg request

The Request string is in general constructed by combining the ISIN and a pricing source. The syntax is:

<name><@source> <yellow key> [type]

The name is an ISIN, the yellow key is "Corp" for bond price requests and the type describes the name e.g. "ISIN". Important is the source e.g. FRTB for Fortis Bank.

Since not all sources deliver prices from Bloomberg and only one source can be requested per time the following algorithm was implemented:

The Bloomberg requester is iterating over a list of predefined sources. This list is configured in the database in the table T42\_MGB\_CONFIGURATION and the key BOND\_BLOOMBERG\_DEFAULT\_SOURCES.

The interval where Bloomberg looks for prices is widened up to an high-low request (to be configured in client.properties).

If a price is found, the successful source is moved to the beginning of the list and this list is saved with the requested instrument to speedup future requests.

### 3.3.2 Data retrieval from Bloomberg

The Bloomberg data will be received from the ActiveX interface of the Bloomberg terminal.

The license model requires that the Bloomberg data will not leave the terminal-PC. To ensure this, the data is saved in a flat file on the local terminal and an anonymous reference is saved in the server database.

The request-string together with all relevant price-check data (trade price, trade time, check type, check thresholds, etc) is sent to the client application. The request-string is passed to Bloomberg. The process is synchronously waiting for the response data.

The price is saved in a local file (to be configured by the user) together with an unique id, which is also saved in the server database to link the price to the corresponding trade.

The client performs the price-check and saves the result, but not the Bloomberg price, in the server database.

So the price data is only displayed and saved on the client-PC and the data is useless for anyone else, since no trade information is save on that client. Only if you have access to the database and your own profile (local price file) is configured correctly, you can read the prices you received and the connected trades.

Bloomberg time is London time, which is in general CET-1!

Published by RMS&C	This substitutes	Project-No.	Author	Section	Page 15	Date 03.12.14
-----------------------	------------------	-------------	--------	---------	------------	------------------