IT-Concept for Repos MCC

Jens Richelsen (DUS - 9352)

Revision log

Date	*Name*	*Description*
06.12.2004	Richelsen	First Draft – Version 0.1
10.03.2006	Richelsen	Adding London Configuration
26.09.2006	Richelsen	Changes of Dus Config
04.01.2007	Richelsen	Changes to Asia client
07.03.2007	Richelsen	Changes of state rules
25.05.2007	Richelsen	Changes in turnover formula
24.10.2007	Richelsen	Changes of state rules Asia/London
10.09.2010	Richelsen	Review of location mapping/rules
30.12.2010	Richelsen	Review
18.07.2011	Richelsen	Marking unexpected locations
26.10.2011	Richelsen	Move of CBB books
21.11.2011	Richelsen	Change of Lon,EAA and HKG client
01.07.2012	Richelsen	Portigon
12.10.2012	Richelsen	Client selection criteria
23.08.2013	R. Steger	Removed EAA and Hong Kong clients
07.04.2014	Richelsen	Removed initial config-values that can be changed in DB
03.12.2014	R. Steger	Added PAG client

Table of contents

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 Repo-products from Summit.

2 Architecture

2.1 Data import

Is described in the document Repo_CR_20060224.doc.

2.1.1 Filter criteria

The extraction of the trades from Summit is defined by the Summit filter DUXX_MCC_REP.

2.1.2 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	PARIS trader location	Location name
Summit Repo London (SRL)	731	EAA London
	743	EAA Istanbul
	VBB_LON	Verbundbank London preparation
	VBB_807	Verbundbank Istanbul preparation
Summit Repo Asia (SRA) (no longer active)		
Summit Repo Hong Kong (SRH) (no longer active)		
Summit Repo EAA (SRE) (no longer active)		
Summit Repo PAG (SRG)	21	London
	38	Shanghai
	26	Hong Kong
	56	Singapore
	48	Madrid
	021	London
	026	Hong Kong
	038	Shanghai
	048	Madrid
	056	Singapore
	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	Instanbul
	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
	EAA2_HKG	EAA Hong Kong refill preparation
	EAA2_LON	EAA London refill preparation
	EAA2_807	EAA Istanbul refill preparation

2.1.3 Field reference

The semicolon separated ascii file contains a header records with the column headers.
The last line (indicated with 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 directly processed by the MGB tool.
A single record consists of the following fields:

Nr	field name	data type	definition	example
1	TradeId	char	TradeId	"0000071412"
2	Status	char	Trade status code	DONE, VER
3	TradeVersion	int	Current version number of the trade	
4	TradeType	char	Specifies the type of the instrument	REPO
5	RepoMarketType	char	market Type, the value "cSAVB_AUTO" indicates an auto-leihe trade	cSAVB_AUTO
6	OpenEndFlag	char	Open end	N
7	UndCategory	char	Category of the underlying	Supra
8	Instrument	char	Sumit name of the instrument	EIB2.625OCT07
9	ISIN	char	ISIN of the instrument	XS0189444564
10	Trader	char	Account of the trader	d055625
11	Book	char	Book ID	WL101
12	Counterparty	char	Counterparty name	BAYERLANMUE
13	CustomerType	char	Customer type	SPARKASSE
14	Start	date	Start time of the trade	20041105
15	End	date	Stop time of the trade	20041112
16	Days	int	number of days between start and end	7
17	TradeTime	date	Trade time	20041103 08:37:00
18	DoneDate	date	Date, when the trade is set to DONE	20041104
19	VerifyDate	date	Date, when the trade is set to VER	20041104
20	AmendDate	date	Date, when the trade was amended	20041104
21	NotionalAmount	float	Notional amount	5000000.0000
22	StartCash	float	Cashflow at the start date	-1812745.12
23	EndCash	float	Cashflow at the end date	2000087.50

24	Ccy	char	Currency	EUR
25	YieldCurve	char	Yield curve	EUR/GC
26	FXRate	float	Rate of currency against EUR	1.0000
27	RepoRate	float	Repo rate	1.90000
28	CurrMarketRate (YieldCurveRate)	float	Current market rate	1.03070
29	FundingSpread	float	Funding spread	1.2
30	MarketPriceUnderlying	float	EOD Market Price Underlying (clean price)	101.12
31	DealAccruedInterest	float	StartDirtyPrice - Startprice (clean Price)	1.256
32	StartPrice	float	clean price	99.45
33	BondAccruedInterest	float	Accrued Interest to Start Date related on Bonddefinion	1.32
34	NPV	float	NPV at EOD	-4587234

2.1.3.1 Special fields/calculations

Within the further processing some fields/value are calculated in a special manner. These are explained in this section.

If field ”undSubType” is either ”SECB” or "SECL", the allowed interval for the repo rate is calculated by defining the lower bound in base points (e.g. 3 PB) and the upper bound as a percentage (e.g. 90%) of the value in the “yieldCurveRate” (which is delivered from SUMMIT as the difference between the EUR-CASH and the GC/BBYFX curve). Based on this interval the following formula is calculating the reference market rate:

$$referenceRate = \frac{yieldCurveRate * relativeToleranceInPercent + absoluteToleranceInBP}{2}$$

The shown tolerance and difference values are refering to the Referencerate as well.

3 MCC logic Düsseldorf

In this chapter, the MCC logic is described.

3.1 General classification mechanism

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

3.1.1 Trade type category

Summit Trade	Trade type category
SECL	SECL
SECB	SECB
REPO	REPO
REV	REV

BSBK	BSBK
SBBK	SBBK
COLL	COLL
NOCL	NOCL
else	REPO

If the trade fulfills the condition is_automatic_wertpapier_anleihe (see below) the "Trade type catagory" is set to "AUTO". If the trade fulfills the condition is_bilaterale_wertpapier_anleihe (see below) the "Trade type catagory" is set to "BILATERAL".

3.1.2 Price thresholds

The tolerance depends on the category and is denoted in base points.

Category	Absolute tolerance in basepoints
REPO	See tolerance_absolute value in sql below
REV	See tolerance_absolute value in sql below
SECL	See tolerance_absolute value in sql below
SECB	See tolerance_absolute value in sql below
BSBK	See tolerance_absolute value in sql below
SBBK	See tolerance_absolute value in sql below
COLL	See tolerance_absolute value in sql below
NOCL	See tolerance_absolute value in sql below

For the following categories the calculation is different. The tolerance interval is defined by a lower bound in base points and an upper bound as a percentage of a reference value from the front office.

Category	Lower bound of tolerance interval in base points	Percentage used to calculate the upper bound of the tolerance interval
BILATERAL	See tolerance_absolute value in sql below	See tolerance_percent value in sql below
AUTO	See tolerance_absolute value in sql below	See tolerance_percent value in sql below

The current values can be extracted from the database by using the following sql:

```
    SELECT t05_instrument AS category,
         t11_tolerance_percent AS tolerance_percent,
         t11_tolerance_absolute AS tolerance_absolute
    FROM t05_instrument, t11_price_check_category
   WHERE     t05_instrument_type = 'pricecheck'
         AND t05_enabled = 'Y'
         AND fk_t05_t11_price_check = t11_id
```

```
        AND fk_t05_t09_mandant LIKE 'REP'
ORDER BY t05_instrument;
```

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.

status_name	check	Expression
New location	Y	!is_expected_location
Already checked an older version	N	!is_mcc_relevant_change
Open End Trade Terminated	N	is_open_end_terminated
Collateral	N	is_collateral
Small rate	Y	is_small_rate
Bloomberg check required	Y	is_sparkasse && (is_automatische_wertpapier_leihe || is_bilaterale_wertpapier_leihe)
Security Lending - Passed	N	is_security_lending && ratediff_less_bp_tolerance && !exceeds_max_pl_diff
Security Lending - Turnover	Y	is_security_lending && ratediff_less_bp_tolerance && exceeds_max_pl_diff
Security Lending - Bagatelle	N	is_security_lending && is_bagatelle && !is_open_end
Security Lending - Failed	Y	is_security_lending
Repo - Passed	N	ratediff_less_bp_tolerance && !exceeds_max_pl_diff
Repo - Turnover	Y	ratediff_less_bp_tolerance && exceeds_max_pl_diff
Repo - Bagatelle	N	is_bagatelle && !is_open_end
Repo - Failed	Y	

The following rules are executed after the Bloomberg data is received.

status_name	check	Expression
Bloomberg check required	Y	has_no_bloomberg_price

WP-Leihe - Passed	N	ratediff_less_bp_tolerance && is_underlying_passed && !exceeds_max_pl_diff
WP-Leihe - Turnover	Y	ratediff_less_bp_tolerance && is_underlying_passed && exceeds_max_pl_diff
WP-Leihe - Bagatelle	N	is_bagatelle && !is_open_end && is_underlying_passed
WP-Leihe - Failed underlying and rate	Y	!ratediff_less_bp_tolerance && !is_underlying_passed
WP-Leihe - Failed underlying	Y	!is_underlying_passed
WP-Leihe - Failed rate	Y	

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 hold in a list which is configured in the database in the table T42_MGB_CONFIGURATION and the key EXPECTED_LOCATIONS. The current values are 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.

3.2.3.2 Condition: is_mcc_relevant_change

The current trade is compared to previous ones. First the last 250 successfully loaded jobs are searched for previous trades. Then a field-comparison is made:

"bondAccruedInterest"
"currency"
"endCash"
"endDate"
"fundingSpread"
"instrument"
"npv"
"openEndFlag"
"repoRate"
"startCash"
"startDate"
"startPrice"
"underlyingValGroup"
"volume"

If one of the files has been changed, the condition returns true.

3.2.3.3 Condition: is_status_done

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

3.2.3.4 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)

3.2.3.5 Condition: is_status_canceled

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

3.2.3.6 Condition: is_small_rate

If the traded ”repoRate” is smaller than 0,01 the condition returns true.

3.2.3.7 Condition: is_sparkasse

If the filed ”customertype” equals “SPARKASSE” the condition returns true.

3.2.3.8 Condition: is_selo

The field ”undSubType” indicates this condition.
If the value is ”SECL”, it returns true.

3.2.3.9 Condition: is_sebo

The field ”undSubType” indicates this condition.
If the value is ”SECB”, it returns true.

3.2.3.10 Condition: is_security_lending

Either is_selo or is_sebo.

3.2.3.11 Condition: is_automatische_wertpapier_leihe

The field ” RepoMarketType” indicates this condition.
If the value is ”cSAVB_AUTO”, it returns true.

3.2.3.12 Condition: is_bilaterale_wertpapier_leihe

The field ” RepoMarketType” must not have the value ”cSAVB_AUTO” and the “TradeType” must either have the value “SECB” or “SECL” to return true.

3.2.3.13 Condition: is_collateral

The field ”undSubType” indicates this condition.
If the value is ”COLL”, it returns true.

3.2.3.14 Condition: is_bagatelle

To calculate the turnover in EUR the following formula is used:

$$turnover = startCash * fxRate * \frac{(repoRate - yieldCurveRate) * (startDate - endDate)}{100 * 360}$$

(The difference between startDate and endDate is set to the value of REPO_OPEN_END_DURATION in the table T42_MGB_CONFIGURATION for open end trades.)

If field ”undSubType” is either ”SECB” or "SECL", the following different formula is used:

$$turnover = fxRate * (\frac{referenceRate}{100 * repoRate} * endCash - endCash)$$

where the referenceRate is calculated as follows:

$$referenceRate = \frac{yieldCurveRate * relativeToleranceInPercent + absoluteToleranceInBP}{2}$$

(The difference between startDate and endDate is set to the value of REPO_OPEN_END_DURATION in the table T42_MGB_CONFIGURATION for open end trades.)

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

The threshold is configured in the database in the table T42_MGB_CONFIGURATION and a key that depends on a trade condition:

trade category	key
is_automatische_wertpapier_leihe	AWPL_BAGATELLE_LIMIT
is_bilatarate_wertpapier_leihe	BWPL_BAGATELLE_LIMIT
is_security_lending	SELO_BAGATELLE_LIMIT
else	REPO_BAGATELLE_LIMIT

The values are supposed to be in EUR.

3.2.3.15 Condition: is_underlying_passed

To check if the underlying is checked, first the Bloomberg price of the underlying or the underlying EOD price is taken as the reference market price and the following formula is evaluated:

$$reference\mathrm{Pr}ice = notional * \frac{(theoretica\,lAccruedIn\,terest + referenceMarket\mathrm{Pr}ice)}{100}$$

If the following relative difference is below a threshold, that depends on the classification above, the condition returns true.

$$diff = \left| \frac{(startValue - reference\mathrm{Pr}ice)}{reference\mathrm{Pr}ice} * 100 \right|$$

3.2.3.16 Condition: exceeds_max_pl_diff

To calculate the turnover in EUR the following formula is used:

$$turnover = startCash * fxRate * \frac{(repoRate - yieldCurveRate) * (startDate - endDate)}{100 * 360}$$

(The difference between startDate and endDate is set to the value of REPO_OPEN_END_DURATION in the table T42_MGB_CONFIGURATION for open end trades.)

If field ”undSubType” is either ”SECB” or "SECL", the following different formula is used:

$$turnover = fxRate * (\frac{referenceRate}{100 * repoRate} * endCash - endCash)$$

where the referenceRate is calculated as follows:

$$referenceRate = \frac{yieldCurveRate * relativeToleranceInPercent + absoluteToleranceInBP}{2}$$

(The difference between startDate and endDate is set to the value of REPO_OPEN_END_DURATION in the table T42_MGB_CONFIGURATION for open end trades.)

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

The threshold is configured in the database in the table T42_MGB_CONFIGURATION and the key REPO_TURNOVER_LIMIT. The values are supposed to be in EUR.

3.2.3.17 Condition: is_open_end

Returns true if the value of the field "openEndFlag" is "Y".

3.2.3.18 Condition: ratediff_less_bp_tolerance

The rate difference is calculated as follows:

$$rateDiff = (repoRate - yieldCurveRate)*100$$

If the rateDiff is smaller than the basepoint tollerance that is defined in the basepoint threshold table described above, the condition returns true.

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 GMT+1!

4 MCC logic London

In this chapter, the MCC logic is described.

4.1 General classification mechanism

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

4.1.1 Trade type category

Summit Trade	Trade type category
SECL	SECL
SECB	SECB
REPO	REPO
REV	REV
BSBK	BSBK
SBBK	SBBK
COLL	COLL
NOCL	NOCL
else	REPO

4.1.2 Price thresholds

The tolerance depends on the category and is denoted in base points.

Category	Absolute tolerance in basepoints
REPO	See tolerance_absolute value in sql below
REV	See tolerance_absolute value in sql below
SECL	See tolerance_absolute value in sql below
SECB	See tolerance_absolute value in sql below
BSBK	See tolerance_absolute value in sql below
SBBK	See tolerance_absolute value in sql below
COLL	See tolerance_absolute value in sql below
NOCL	See tolerance_absolute value in sql below

The current values can be extracted from the database by using the following sql:

```
    SELECT t05_instrument AS category,
         t11_tolerance_percent AS tolerance_percent,
         t11_tolerance_absolute AS tolerance_absolute
    FROM t05_instrument, t11_price_check_category
   WHERE     t05_instrument_type = 'pricecheck'
         AND t05_enabled = 'Y'
         AND fk_t05_t11_price_check = t11_id
         AND fk_t05_t09_mandant LIKE 'REL'
ORDER BY t05_instrument;
```

4.2 Trade status checks

4.2.1 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.

4.2.2 Definition of trade status rules

The following rules are used to calculate the status of the trade.

status_name	**check**	**Expression**
Already checked an older version	N	!is_mcc_relevant_change
Open End Trade Terminated	N	is_open_end_terminated
Collateral	N	is_collateral || is_collateral_portfolio
Small rate	Y	is_small_rate
Security Lending - Passed	N	is_security_lending && ratediff_less_bp_tolerance && !exceeds_max_pl_diff
Security Lending - Turnover	Y	is_security_lending && ratediff_less_bp_tolerance && exceeds_max_pl_diff
Security Lending - Bagatelle	N	is_security_lending && is_bagatelle && !is_open_end
Security Lending - Failed	Y	is_security_lending
Repo - Passed	N	ratediff_less_bp_tolerance && !exceeds_max_pl_diff
Repo - Turnover	Y	ratediff_less_bp_tolerance && exceeds_max_pl_diff
Repo - Bagatelle	N	is_bagatelle && !is_open_end
Repo - Failed	Y	

4.2.3 Definition of the conditions

All conditions behave like the ones in the Düsseldorf client above.