

MultiCash[®] / MultiCash Transfer[®]

SEPA Payments (SPA) 3.20

Technical information

August 2017

Omikron Systemhaus GmbH & Co. KG
Von-Hünefeld-Str. 55
D-50829 Cologne (Germany)

Tel.: +49 (0)221 -59 56 99 -0
Fax: +49 (0)221 -59 56 99 -7

info@omikron.de
www.omikron.de

Table of Contents:

1	INTRODUCTION	4
1.1	General Information	4
1.2	General Processing Principles	5
1.2.1	Format Sub-System.....	5
1.2.2	Payment Order Module	5
1.2.3	General.....	5
2	SUPPORTED FORMATS	6
2.1	Format Family UNIFI / EPC.....	6
2.2	Format Family Austria	6
2.3	Format Family Germany	7
2.4	Format Family MultiCash	8
3	SUPPORTED SESSION TYPES	9
3.1	Upload Session Types.....	9
3.2	Download Session Types.....	10
4	CONFIGURATION OPTIONS ON CUSTOMER SIDE (CUSTOMER MODULE)	11
4.1	General configuration using the SEPA.INI	11
4.1.1	Section [COMMS_*] - Customizing available file creation procedures.....	11
4.1.2	Section [StartdatumVn.n] - Start dates for new rulebooks.....	13
4.1.3	Section [START_COR1] - Start date for COR1 direct debits (before RB9).....	14
4.1.4	Section [DD_TIME_RANGES] - Submission deadlines for SEPA direct debits ..	14
4.1.5	Section [DEL_LEADING_ZERO] - Handling of leading zeroes (CZ, SK only)	16
4.1.6	Section [TRACE] - Tracing	16
4.1.7	Section [PurposeCode] - Special purpose code for capital building benefits	17
4.1.8	Section [SessiontypCheck] – Check for assumed file content	18
4.1.9	Section [MandatePlus] – Disable ILM alerts (SPM module).....	18
4.1.10	Section [MVPlus] – Send BCC (SPM module)	19
4.1.11	Section [COMMON] - No display of large files (SEPA check program SPC)	19
4.1.12	Section [COMMON] - No AWW processing of large files (AWV declaration program CBD)	19
4.1.13	Section [COMMON] - Non-Liability-Clause on IZV/AZV data import.....	20
4.1.14	Section [COMMON] - BIC determination from IBAN ("IBAN only")	21
4.1.15	Section [COMMON] - Controlling the file creation	21
4.1.16	Section [COMMON] - Expiry of mandates not used.....	22
4.1.17	Section [COMMON] - Activate EPC character set restriction	22
4.1.18	Section [COMMON] - Use alternative scheme (Austria)	23
4.1.19	Section [COMMON] - Performance optimization for checking and preparation for the display	23
4.1.20	Section [COMMON] - Mark unconfirmed records from IBAN conversion.....	24
4.1.21	Section [COMMON] - Differentiation between "hard technical" and "soft functional" errors	24

4.1.22	Section [COMMON] – Log level for e-mail sending (SPM module)	25
4.1.23	Section [COMMON] – Check urgent transfers for presence of the BIC	27
4.1.24	Section [COMMON] – Only one batch per file	27
4.1.25	Section [COMMON] – Activate format variant for RB4 additionally	28
4.1.26	Section [COMMON] – Add client ID to E2E reference	28
4.1.27	Section [COMMON] – Control of the checks for the money transfer regulation ..	29
4.1.28	Mapping tables for reject information	29
4.2	Configuration files for code lists SPACD*.INI.....	31
4.2.1	Naming conventions	31
4.2.2	Structure	32
4.2.3	Example	36
4.3	Configuration files for code list texts SPACDTEXT*.INI	37
4.3.1	Naming conventions	37
4.3.2	Structure	38
4.3.3	Example	40
4.3.4	Concrete example: Special keywords for Germany	41
4.3.5	Concrete example: Special keywords for Austria	41
4.4	Configuration files for mandate defaults SPAMD*.INI	43
4.4.1	Naming conventions	43
4.4.2	Structure	43
4.4.3	Example	45
4.5	Rules for the detection of R transactions using INFO-MAPPING.INI	46
4.6	Deactivation of mandates using MANDATE-STOP.INI.....	47
5	CONFIGURATION OPTIONS ON BANK SIDE (OMIKRON BANK SERVER)	48
5.1	Configuration via SEPA.INI	48
5.1.1	Section [COMMS_ALLOWED] – Allowed files	48
5.1.2	Section [DEL_LEADING_ZERO] - Handling of leading zeroes (CZ, SK only)	48
5.1.3	Section [TRACE] - Tracing	49
5.1.4	Section [SessiontypCheck] – Check for assumed file content	49
5.1.5	Section [COMMON] - New format sub-system under bank server 3.21	50
5.1.6	Section [COMMON] - Switch off IBAN and CIN check	50
5.1.7	Section [COMMON] - Limit acceptance to the last 2 scheme versions	50
5.1.8	Section [COMMON] – Control of the checks for the money transfer regulation ..	51
5.1.9	Section [DD_TIME_RANGES] - Submission deadlines for direct debits / Check	51

1 Introduction

1.1 General Information

SEPA must be regarded as a general revolution in payment processing as it is a common initiative in all EURO countries making use of the global **UNIFI** scheme (**UNI**versal **F**inancial **I**ndustry message scheme) based on the definition **ISO 20022**.

Although this is a general specification, the national banking communities tend to use this common framework with their local flavour in order to be free in their specification procedures during the necessary process of European standardization.

The result of this process is a “jungle” of

- format families
- format versions
- old and new services
- migration problems
- special requirement in some countries or even for some banks

Omikron tries to cover all that with their MultiCash product families and intends to hide the complexity of this SEPA introduction process for the user. The many evolution cycles of the rulebooks defined by the different specification groups lead to frequent changes in the involved software components.

From release 3.20.030 in the MultiCash SEPA module no detailed description of the different formats is included anymore, because this information is reliably available on the public web sites of ISO, EPC and the national banking organisations.

This document lists all the supported format families and gives an overview over the “hidden” configuration options which should be used only in special cases.

It is addressed to

- all users of the MultiCash SEPA module
- the support teams of your Omikron software partner or bank
- Omikron support teams
- Omikron partners

1.2 General Processing Principles

1.2.1 Format Sub-System

1. The standard transfer processing for payment orders from third party applications executes format checks on the files but never changes them.
2. The format checks are executed by format sub-systems which identify the applying rules by the namespace in the file header.
3. Files with unknown namespace are refused.
4. The format checks are executed on two layers:
 - a) XML schema: All formal rules defined in linked schema:
 - file structure
 - mandatory fields
 - field content (as far as defined in the schema)
 - b) Application layer:
General:
 - check digits of IBAN (configurable, SEPA.INI)
 - check digits of Creditor IDSpecial rules according to the rulebook of the format "owner":
 - additional mandatory fields
 - character set restrictions (configurable, SEPA.INI)
 - content restrictions like length of fields or number of repetition
 - special field content (configurable, several INI files)
5. The format sub-system supports all older format versions (parameter, SEPA.INI: accept only current and last version).

1.2.2 Payment Order Module

1. The payment order module supports the format family which is defined in the bank table.
2. The payment order module switches automatically to a newer version when the defined start date has come (configurable, SEPA.INI)
3. Incoming payment status reports (pain.002) are matched to the payment history if activated. Here all listed old and new formats are supported in parallel. However, for processing of ZIP-Containers the add-on module "XML Account Information" is necessary.

1.2.3 General

The processing can be changed to a certain extent with configuration file SEPA.INI which is described below.

2 Supported Formats

Message versions introduced with RB 9.0 are marked **red**.

2.1 Format Family UNIFI / EPC

Version	Namespace	Message	From Release
RB 3.2 (UNIFI 2006)	urn:iso:std:iso:20022:tech:xsd:pain.001.001.02	pain.001	3.20.012
	urn:iso:std:iso:20022:tech:xsd:pain.008.001.01	pain.008	
	urn:iso:std:iso:20022:tech:xsd:pain.002.001.01	pain.002	3.20.012
RB 4.0 RB 5.0 (UNIFI 2009)	urn:iso:std:iso:20022:tech:xsd:pain.001.001.03	pain.001	3.20.019
	urn:iso:std:iso:20022:tech:xsd:pain.008.001.02	pain.008	
	urn:iso:std:iso:20022:tech:xsd:pain.002.001.02	pain.002	3.20.030
RB 6.0/7.0 (UNIFI 2009)	urn:iso:std:iso:20022:tech:xsd:pain.007.001.02	pain.007	3.20.027 (add-on module)
	Like RB 4.0	pain.001 pain.008	SPA 3.20.030: (DD COR1, IBANOnly)
	urn:iso:std:iso:20022:tech:xsd:pain.002.001.03	pain.002	SPA 3.20.027
RB 9.0 (UNIFI 2009)	urn:iso:std:iso:20022:tech:xsd:pain.002.001.04	pain.002	SPA 3.20.030
	Like RB 4.0	pain.001 pain.008	Transfers unchanged compared with previous version Direct debits according to RB 9.0/7.0 starting from Release 3.20.039
UNIFI 2012	urn:iso:std:iso:20022:tech:xsd:pain.001.001.04	pain.001 pain.007 pain.008	Zunächst kein Bedarf, da vom EPC nicht verwendet

2.2 Format Family Austria

Version	Namespace	Message	From Release
RB 2.3	APC:STUZZA:payments:ISO:pain:001:001:02:austrian:001 paxx.container.003.xsd	pain.001 Container	3.20.001 (not recommended any more, still supported)
RB 3.2	APC:STUZZA:payments:ISO:pain:001:001:02:austrian:002 APC:STUZZA:payments:ISO:pain:008:001:01:austrian:002	pain.001 pain.008	3.20.012
RB 4.0	ISO:pain.001.001.03:APC:STUZZA:payments:001 ISO:pain.008.001.02:APC:STUZZA:payments:001	pain.001 pain.008	3.20.023

Version	Namespace	Message	From Release
RB 5.0	ISO:pain.001.001.03:APC:STUZZA:payments:002 ISO:pain.008.001.02:APC:STUZZA:payments:002	pain.001 pain.008	3.20.030
RB 6.0/ RB 7.0	ISO:pain.001.001.03:APC:STUZZA:payments:003 ISO:pain.008.001.02:APC:STUZZA:payments:003 ISO:pain.002.001.03:APC:STUZZA:payments:003	pain.001 pain.008 pain.002	3.20.030
	ISO:pain.001.001.03:APC:STUZZA:payments:003:N	pain.001N	Non-SEPA payments: Not supported by SEPA module

Remark: According to the STUZZA/APC definitions by default the payment order module creates messages in UNIFI format. This can be changed if necessary (see chapter 4.1.14)

2.3 Format Family Germany

Version	Namespace	Message	From Release
2.2/2.3 ZKA (EPC 2.3)	urn:sepade:xsd:pain.001.001.02 urn:sepade:xsd:pain.008.001.01	pain.001 pain.008	3.20.001 (session type CCM, not recommended any more, still supported)
2.2/2.3 ZKA Container (EPC 2.3)	urn:sepade:xsd:pain.001.001.02.con urn:sepade:xsd:pain.008.001.01.con	pain.001 pain.008	3.20.001
2.2/2.3 Ext.Grp. (EPC 2.3)	urn:sepade:xsd:pain.001.001.02.grp urn:sepade:xsd:pain.008.001.01.grp	pain.001 pain.008	3.20.001
2.4 (EPC 3.2)	urn:swift:xsd:\$pain.001.002.02 urn:swift:xsd:\$pain.008.002.01 urn:swift:xsd:\$pain.002.002.02	pain.001 pain.008 pain.002	3.20.012
2.5/2.6 (EPC 4.0/2.0)	urn:iso:std:iso:20022:tech:xsd:pain.001.002.03 urn:iso:std:iso:20022:tech:xsd:pain.008.002.02 urn:iso:std:iso:20022:tech:xsd:pain.002.002.03	pain.001 pain.008 pain.002	3.20.019
2.7/2.8/2.9 (EPC 7.0/5.0)	urn:iso:std:iso:20022:tech:xsd:pain.001.003.03 urn:iso:std:iso:20022:tech:xsd:pain.008.003.02 urn:iso:std:iso:20022:tech:xsd:pain.002.003.03	pain.001 pain.008 pain.002	3.20.030
3.0 (EPC SCT 8.0 SDD/B2B 9.0/7.0, = ISO/UNIFI 2009)	urn:iso:std:iso:20022:tech:xsd:pain.001.001.03 urn:iso:std:iso:20022:tech:xsd:pain.008.001.02 urn:iso:std:iso:20022:tech:xsd:pain.002.001.02	pain.001 pain.008 pain.002	starting from Release 3.20.039

2.4 Format Family MultiCash

For some countries in Central Europe with active MultiCash user groups Omikron has created own schema definitions in order to support special requirements of the national banking groups. This initiative started with reduced and adopted subsets of the EPC schemes as the priority was to have an easy-to-use solution.

With increasing use of the SEPA solution with many changes in the rule books we see that many specific schema definitions cannot be supported with reasonable commercial impact. Therefore, beginning with rulebook 6.0, these messages are based on the UNIFI / EPC schemes.

Special local rules are still supported in the payment order module according to the format family linked to the ordering party bank. But the exported payment orders are always compatible to the UNIFI format family. For third party files processed by the MultiCash client and bank applications no special checks are executed as all files are based on the standard ISO 20022 namespace.

Type	Owner	Identification xxx
General	Omikron	Without ID
National	Banks of the following countries: Poland Czech Republic Hungary Slovak Republic	pl cz hu sk
Bank	Unicredit Bank	cee

Version	Namespace	Message	From Release
RB 2.3	urn:multicash:xxx:sepa:pain.001.001.02	pain.001 pain.008	3.20.012
RB 3.2	urn:multicash:xxx:sepa:pain.001.002.02 urn:multicash:xxx:sepa:pain.008.002.01	pain.001 pain.008	3.20.012
RB 4.0/ RB 5.0	urn:multicash:xxx:sepa:pain.001.002.03 urn:multicash:xxx:sepa:pain.008.002.02	pain.001 pain.008	3.20.019
RB 6.0/ RB 7.0/ RB8.0 RB 9.0	urn:iso:std:iso:20022:tech:xsd:pain.001.001.03 urn:iso:std:iso:20022:tech:xsd:pain.008.001.02	pain.001 pain.008	3.20.030 starting from Release 3.20.039

3 Supported Session Types

3.1 Upload Session Types

Session type	Description/Remarks	Message	Format Sub-System
CCM	SEPA Credit Transfer Initiation ZKA 2.3 (old, to be supported until 01.11.2010)	pain.001	CCM
CDM	SEPA Direct Debit Initiation ZKA 2.3 (old, to be supported until 01.11.2010)	pain.008	CCM
CCT	SEPA Credit Transfer Initiation	pain.001	CCM
CCU	SEPA Urgent Payments	pain.001	CCM
CDD	SEPA Direct Debit Initiation	pain.008	CCM
CD1	SEPA Direct Debit Initiation COR1	pain.008	CCM
CDB	SEPA Business Direct Debit	pain.008	CCM
CCC	SEPA Credit Transfer Initiation Container (German container format)	pain.001	CCM
CDC	SEPA Direct Debit Initiation Container (German XML container format upload)	pain.008	CCM
C1C	SEPA Direct Debit COR1 Container	pain.008	CCM
C2C	SEPA B2B Direct Debit Container (German XML container format upload)	pain.008	CCM
CPR	SEPA Payment reversals	pain.007	CPR ¹⁾
German XML container format upload by service providers ²⁾			
CCS	SEPA Credit Transfer by Servicer	pain.001	CCM
CCX	SEPA Credit Transfer by Servicer (VEU) (optional for distributed signature)	pain.001	CCM
CDS	SEPA Direct Debit by Servicer (VEU)	pain.008	CCM
CDX	SEPA Direct Debit by Servicer (optional for distributed signature)	pain.008	CCM
C1S	SEPA Direct Debit COR1 by Servicer	pain.008	CCM
C1X	SEPA Direct Debit COR1 by Servicer (VEU) (optional for distributed signature)	pain.008	CCM
C2S	SEPA B2B Direct Debit by Servicer	pain.008	CCM
C2X	SEPA B2B Direct Debit by Servicer (VEU) (optional for distributed signature)	pain.008	CCM

¹⁾ Add-on modules

²⁾ Supported by format sub-system only, not by payment order module

3.2 Download Session Types

Session type	Description/Remarks	Message
CRJ	SEPA Status Report Credit ZKA 2.3 (old, to be supported until 01.11.2010)	pain.002
CDR	SEPA Status Report Direct Debit ZKA 2.3 (old, to be supported until 01.11.2010)	pain.002
CRC	SEPA Status Report Credit Container (German XML container format download)	pain.002
CBC	SEPA Status Report Direct Debit Container (German XML container format download)	pain.002
CRZ ¹⁾	Payment Status Report for Credit Transfer (ZIP container)	pain.002
CDZ ¹⁾	Payment Status Report for Direct Debit (ZIP container)	pain.002
ACK ²⁾	EBICS Acknowledgment message France (old, for EBICS version 2.4, H003)	pain.002 (propr.)
HAC ²⁾	EBICS Acknowledgment message (from EBICS version 2.5, H004)	pain.002 (propr.)

¹⁾ Add-on module "XML Account Information" required

²⁾ These session types are part of the EBICS specification, not of the SEPA concept. However, they are proprietary definitions of the pain.002 format and therefore sometimes cause confusion. You should know that they are used as technical confirmation messages (ACK/NACK) for all types of data on transport layer only. They should not be mixed up with pain.002 messages used on SEPA layer.

4 Configuration options on customer side (customer module)

4.1 General configuration using the SEPA.INI

The configuration of the SEPA payments module is mainly effected using the SEPA.INI file in the ..\DAT directory of the customer module.

The given **default** values are effective if the corresponding parameter is missing or the SEPA.INI is completely missing.

4.1.1 Section [COMMS_*] - Customizing available file creation procedures

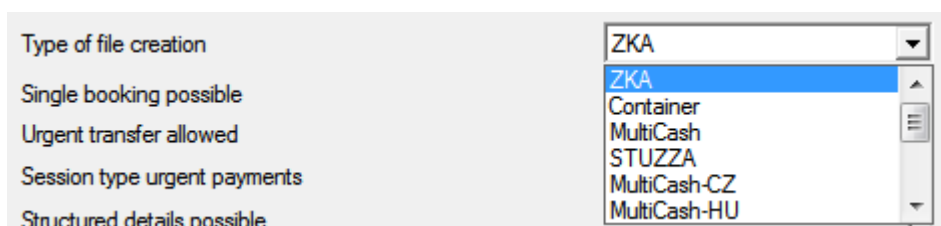
Using the sections starting with [COMMS_...] or [COMMSnn_...] in the SEPA.INI configuration file, which is located in the ..\DAT directory of the client system, you can specify type and number of the file creation procedures available in the Banks reference table.

Section [COMMS_DEFAULT] or [COMMSnn_DEFAULT]:

The default entry, which provides all procedures available in the client system of a defined version (nn represents the version of the corresponding EPC rulebook, e.g. 70 stands for the SEPA rulebook version 7.0), is as follows:

```
[COMMS70_DEFAULT]
NUMBERS=12
COMMS1=ZKA
COMMS2=Container
COMMS3=MultiCash
COMMS4=STUZZA
COMMS5=MultiCash-CZ
COMMS6=MultiCash-HU
COMMS7=UNIFI
COMMS8=MultiCash-CEE
COMMS9=MultiCash-PL
COMMS10=MultiCash-SK
COMMS11=UNIFI 2009
COMMS12=SEPA CH
COMMS13=SEPA IT
```

The options are offered in a list box on the client system:



For the entries in the SEPA.INI the following syntax applies:

After the section label [COMMS_DEFAULT] first the number <n> of allowed procedures follows:

NUMBERS=<n>

Subsequently, the list of the procedures follows:

COMMS<1>=<Name of the procedure>

...

COMMS<n>=< Name of the procedure >

Please note:

Beside the number also the sequence of the procedures can be changed, however not the names of the procedures. They have to be used exactly as named above.

The sections described in the following are optional. They can be used to limit the available types of file creation country-specifically and/or bank group and country-specifically.

Section [COMMS_<Country code>] or [COMMSnn_<Country code>]:

This section can occur several times. Using this the number of available procedures can be limited country-specifically. On entering a BIC in the Banks reference table containing a country code defined in this section of the SEPA.INI the default options are limited accordingly.

```
[COMMS70_DE]
NUMBERS=2
COMMS1=Container
COMMS2=ZKA
```

E.g. the above section inserted additionally effects that for German SWIFT addresses (country code DE) only two procedures are available:

```
[COMMS70_AT]
NUMBERS=1
COMMS1=STUZZA
```

The above section e.g. effects that for Austrian SWIFT addresses (country code AT) only one procedure is available.

Section [COMMS_<Bank code><Country code>] or [COMMSnn_<Bank code><Country code>]:

This section also can occur several times. Using this the number of available procedures can be limited bank group and country-specifically. If the SWIFT address entered in the Banks reference table matches a combination of bank code (digits 1-4 of the BIC) and country code (digits 5+6 of the BIC) specified in the SEPA.INI, the default options are limited accordingly.

```
[COMMS70_GEBAA]  
NUMBERS=3  
COMMS1=STUZZA  
COMMS2=UNIFI  
COMMS3=UNIFI 2009
```

The above section inserted additionally effects that e.g. for SWIFT addresses of this bank three options are available.

The SWIFT address check always starts with the special one and ends with the general one, i.e. first it is checked whether a restriction concerning bank group+country applies. If that is not the case, it is checked whether a country-specific restriction exists. Even if this is not the case, the options are provided as defined in the [COMMS_DEFAULT] or [COMMSnn_DEFAULT] section.

With entries in this section also the order of the offered procedures can be controlled.

Another bank maybe requires, that the available procedures should appear in a different order. This can be effected by the following entry:

```
[COMMS_BANKDE]  
NUMBERS=2  
COMMS1=Container  
COMMS2=ZKA
```

On entering the BIC of this bank the ZKA procedure will be provided on the second place.

4.1.2 Section [StartdatumVn.n] - Start dates for new rulebooks

Using entries in these sections, start dates can be defined from those files according to new rules are to be created. Starting from these dates, new payments entry dialogs may appear automatically in accordance with the new version and appropriate files are created.

The date can be controlled like the following (the given examples are also the **default values** in the payments module and in the log DLL):

```
[StartdatumV3.2]  
STUZZACT=1.03.09  
STUZZADD=1.11.09  
DE2CT=1.11.09  
DE2DD=1.11.09  
  
[StartdatumV4.0]  
DE3CT=1.11.10  
DE3DD=1.11.10  
STUZZA3CT=1.11.12  
STUZZA3DD=1.11.12
```

```
[StartdatumV6.0]
STUZZA5CT=1.04.13
STUZZA5DD=1.04.13
```

```
[StartdatumV7.0]
DE4CT=1.11.13
DE4DD=1.11.13
```

```
[StartdatumV9.0]
DE9CT=20.11.16
DE9DD=20.11.16
```

4.1.3 Section [START_COR1] - Start date for COR1 direct debits (before RB9)

This parameter is only interpreted for old SDD procedures previous to RB9!

Use this parameter to define country-specifically the starting point for so-called COR1 direct debits, i.e. Core direct debits with a lead time shortened to one day:

```
[START_COR1]
DE=4.11.13
AT=8.04.13
```

The **default value** in the payments module is in both cases the "1.02.14".

4.1.4 Section [DD_TIME_RANGES] - Submission deadlines for SEPA direct debits

The entries in this section control the submission deadlines for SEPA direct debits (DD).

In order to ensure the submission in time the following lead time values are defined as default. Examples (the values given for CORE1, CORE2, COR1 and B2B are also the **default values** in the payments module and in the log DLL):

```
[DD_TIME_RANGES]
CORE1=7
CORE2=4
COR1=3
B2B=3
ChkDDTimeRanges=0
CutOffTime=15:00
ReqdColltnDtModification=0
```

Here, after the conversion date to RB 9 (20.11.1016) the entries for the old CORE rules (CORE1,

CORE2) can be set to the same value as for COR1, if still old formats should be supported. For standard direct debit messages according to RB 9 only the keyword COR1 is interpreted.

[DD_TIME_RANGES]

CORE1=3

CORE2=3

COR1=3

B2B=3

The further parameters:

The control of the submission times is activated by setting the parameter

ChkDDTimeRanges=1

(Deactivate: set to "0").

The **default value** in the log DLL is "0" (i.e. the deadline check is deactivated by default starting from release 3.20.031), since most of the banks also accept SEPA direct debits, if the submission deadline has been exceeded.

Since the submission of SEPA direct debits in time is sometimes very important, on determining the lead time the current time is to be considered. If the determination is carried out after cut-off time, the count of the days for the lead time starts at the following day.

The cut-off time to be considered is given behind the following parameter. Example:

CutOffTime=15:00

Default cut-off time is 15:00 o'clock.

The **default value** in the payments module and in the log DLL is "12:00".

If the due date should be modifiable, this can be made by setting the parameter

ReqdColltnDtModification=1

enabled (only in the context of a special solution, please do not modify!).

The **default value** in the log DLL is "0".

4.1.5 Section [DEL_LEADING_ZERO] - Handling of leading zeroes (CZ, SK only)

Using this entry leading zeroes can be removed from bank codes extracted from IBAN respectively (relevant for CZ and SK):

```
[DEL_LEADING_ZERO]  
DELETE=1
```

effects, that leading zeroes before the bank codes are removed.

```
[DEL_LEADING_ZERO]  
DELETE=0
```

effects, that leading zeroes before the bank codes remain.

The **default value** in the log DLL is "0".

4.1.6 Section [TRACE] - Tracing

Using the following entry, problems in the context of XML (so-called exceptions) can be traced.

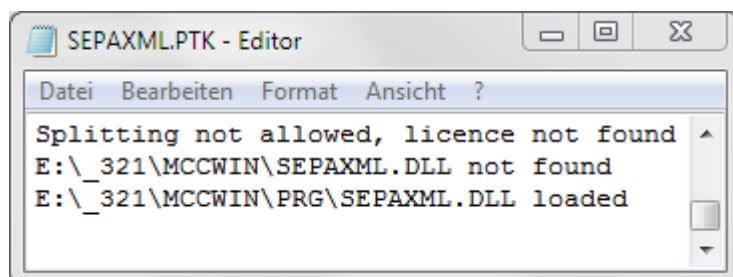
```
[TRACE]  
SEPAXML=0
```

means, that the trace file is not written (**default value** in the log DLL).

```
[TRACE]  
SEPAXML=1
```

effects, that the trace file (SEPAXML.PTK) is written.

Example:



Using the following entry, an error log can be generated.

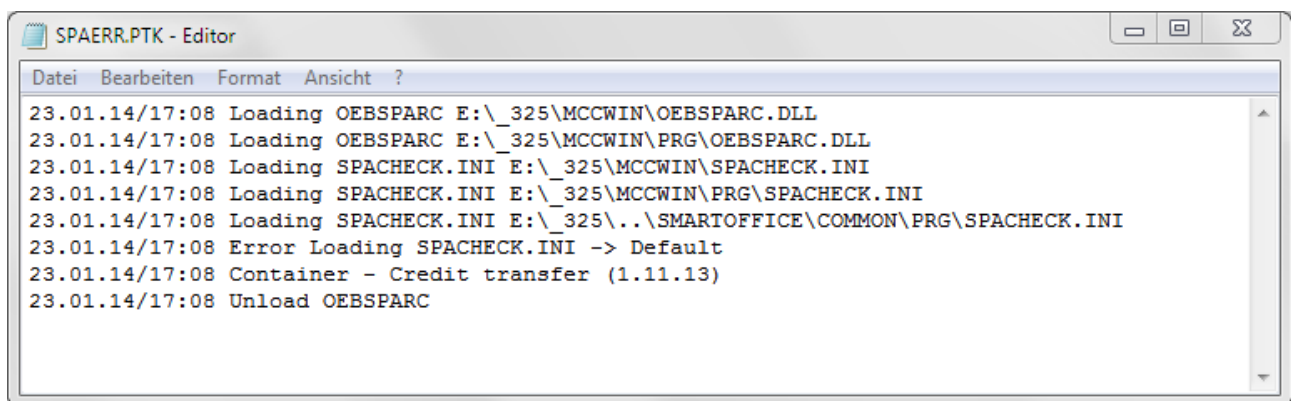
```
[TRACE]
SPAERR=0
```

means, that the error log is not written (**default value**).

```
[TRACE]
SPAERR=1
```

effects, that the error log SPAERR.PTK is written.

Example:

The screenshot shows a window titled "SPAERR.PTK - Editor" with a menu bar containing "Datei", "Bearbeiten", "Format", "Ansicht", and "?". The main text area contains the following log entries:

```
23.01.14/17:08 Loading OEBSPARC E:\_325\MCCWIN\OEBSPARC.DLL
23.01.14/17:08 Loading OEBSPARC E:\_325\MCCWIN\PRG\OEBSPARC.DLL
23.01.14/17:08 Loading SPACHECK.INI E:\_325\MCCWIN\SPACHECK.INI
23.01.14/17:08 Loading SPACHECK.INI E:\_325\MCCWIN\PRG\SPACHECK.INI
23.01.14/17:08 Loading SPACHECK.INI E:\_325\...\SMARTOFFICE\COMMON\PRG\SPACHECK.INI
23.01.14/17:08 Error Loading SPACHECK.INI -> Default
23.01.14/17:08 Container - Credit transfer (1.11.13)
23.01.14/17:08 Unload OEBSPARC
```

4.1.7 Section [PurposeCode] - Special purpose code for capital building benefits

For capital building fringe fortune payments (in Germany: VWL-Zahlungen=**V**ermögens**w**irksame **L**eistungen) a new Purpose Code was requested. It is defined for the payments module in the SEPA.INI by the following entry:

```
[PurposeCode]
VWL=CBFF
```

(The abbreviation CBFF stands for Capital building fringe fortune.)

4.1.8 Section [SessiontypCheck] – Check for assumed file content

If the following parameter is set as follows, a check is made, whether that, what can be assumed from the session type, is included in the file. That is e.g. CCC: Container transfers, CDD: Core direct debits, CD1: COR1 direct debits, CCT: Transfers etc.

```
[SessiontypCheck]  
NoCheck=No
```

By setting the parameter to "Yes", the check can be deactivated consequently.

Remark:

We recommend that you should not change the parameter without consultation with Omikron.

By **default** nothing is used.

4.1.9 Section [MandatePlus] – Disable ILM alerts (SPM module)

If the alert messages in the ILM, which are available when you use the mandate administration plus (module ID SPM), should be suppressed, this can be done for each event (mandate feedback, pre-notification, expiry, inactivation) separately in the [MandatePlus] section using the following entries.

```
ChkFeedback=No  
ChkPreAnnouncement=No  
ChkExpiry=No  
ChkInactive=No
```

If e.g. pre-notifications are not generated within the program, the alert function for this in the ILM can be deactivated by the following entry.

```
[MandatePlus]  
ChkPreAnnouncement=No
```

By **default** nothing is used.

4.1.10 Section [MVPlus] – Send BCC (SPM module)

If for verification purposes (such as mailing the pre-notification) a BCC should be sent to a central E-Mail address, this can be achieved by following entry.

```
[MVPlus]  
MailBCC=<emailadresse>
```

Is the entry used a hidden copy to the given address will be sent for each e-mail message.

By **default**, nothing is set.

4.1.11 Section [COMMON] - No display of large files (SEPA check program SPC)

The processing for the screen display can be suppressed for files starting from a defined size by the following entry in the section [COMMON]. Then, only the check log is available in the check program for SEPA files.

```
[COMMON]  
DONTCheckFileOverMB=1
```

effects, that for files with a size above 1 MB there is no check possibility at application level.

The **default value** for the SEPA check program is "0".

4.1.12 Section [COMMON] - No AWV processing of large files (AWV declaration program CBD)

The AWV processing can be suppressed for files starting from a defined size by the following entry in the section [COMMON].

```
[COMMON]  
DONTCheckFileOverMB_<Session type>=xx1
```

The **default value** is "20" MB.

4.1.13 Section [COMMON] - Non-Liability-Clause on IZV/AZV data import

If a non-liability text should be displayed on importing IZV / AZV data into the SEPA module, this can be defined in the system by the following entry in the section [COMMON]:

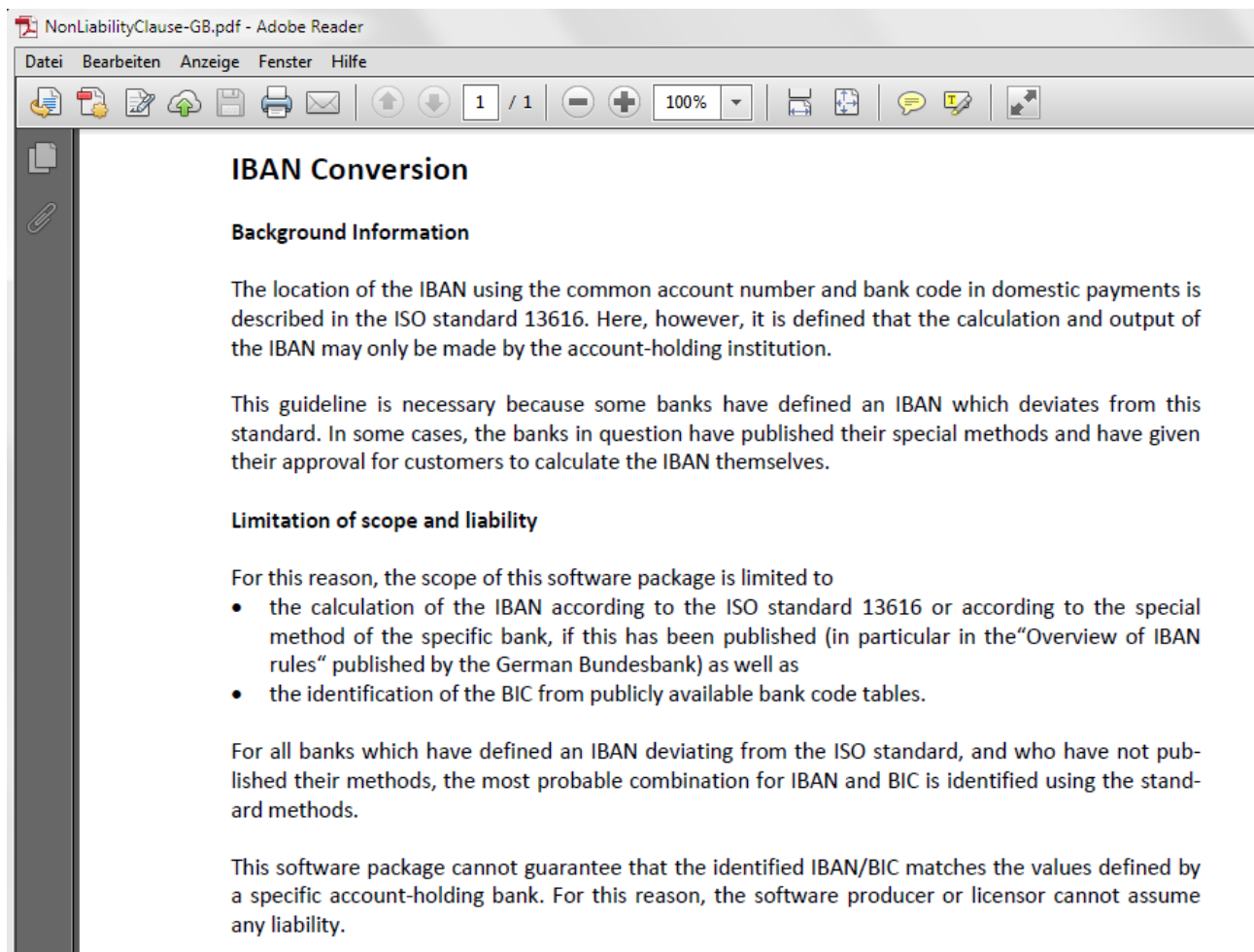
NonLiabilityClause=<Path\File name.Extension>

Example (also **default value** in the payments module):

NonLiabilityClause =..\MCCWIN\DAT\NonLiabilityClause-GB.PDF

Then the defined file is shown, as soon as the button [**Display use conditions**] is clicked under the menu item -Data import-.

Example:



4.1.14 Section [COMMON] - BIC determination from IBAN ("IBAN only")

For SEPA orders within Germany the entry can be simplified in that way, that only the keying of the IBAN is necessary. This is effected by the following parameter in the section [COMMON] (**Default**).

IBANOnly=1

If a German IBAN is entered then, the appropriate BIC is determined automatically on the basis the current bank code number table of the German Federal Bank and the country code from the IBAN (to enable this, the additional module IBAN Finder must be installed).

Vice versa, the automatic addition of the BIC can be switched off by the following entry.

IBANOnly=0

The mandatory entry of BIC for national payments is only necessary until the 31.01.2014, for cross-border payments until the the 31.01.2016. After these dates it is sufficient to enter the IBAN only, the BIC is determined automatically in each time. This procedure is ensured by the following entries.

Example (also **default values** in the payments module):

NATIONAL_END_BIC=31.01.14

TOTAL_END_BIC=31.01.16

Using the parameter **TOTAL_END_BIC_XX** the introduction of IBANONLY can be postponed for individual countries (XX=country code).

4.1.15 Section [COMMON] - Controlling the file creation

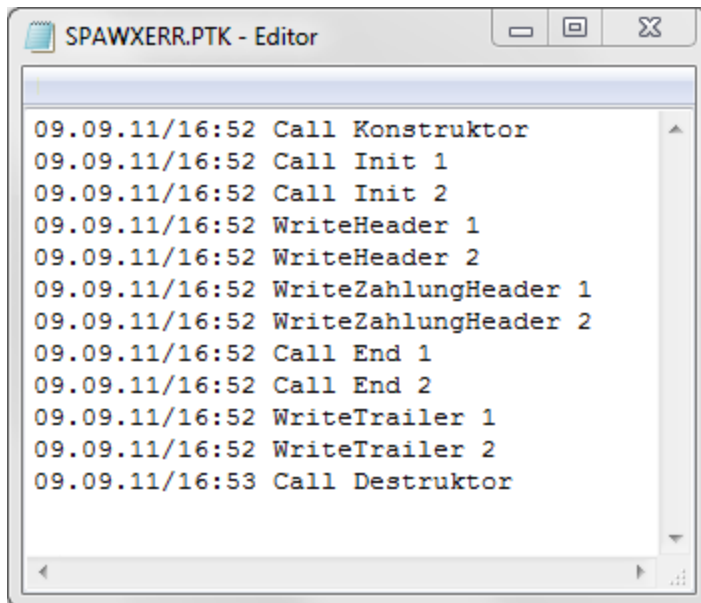
Deviating from the normal handling (DOM) the file creation can be controlled using the following switch in the section [COMMON].

WriteXMLFlag=1

effects that the file is written sequentially. The **default value** in the payments module is "0".

WriteXMLFlag=9

effects that a log named SPAWXERR.PTK is written additionally.



```
09.09.11/16:52 Call Konstruktor
09.09.11/16:52 Call Init 1
09.09.11/16:52 Call Init 2
09.09.11/16:52 WriteHeader 1
09.09.11/16:52 WriteHeader 2
09.09.11/16:52 WriteZahlungHeader 1
09.09.11/16:52 WriteZahlungHeader 2
09.09.11/16:52 Call End 1
09.09.11/16:52 Call End 2
09.09.11/16:52 WriteTrailer 1
09.09.11/16:52 WriteTrailer 2
09.09.11/16:53 Call Destruktor
```

4.1.16 Section [COMMON] - Expiry of mandates not used

The time, after which a mandate not used (calculated from the due date of the most recently submitted direct debit) expires, is specified by the following parameter in section [COMMON]. After this time a new mandate must be requested from the debtor. Example (also **default value** in the payments module):

MandateUsableNoOfMonth=36

is the default setting according to SEPA specifications.

4.1.17 Section [COMMON] - Activate EPC character set restriction

By default, there is no character set restriction. All characters encoded correctly according to UTF-8 are allowed for manual entry and during file check. If necessary, the character set restriction according to EPC definition can be activated for the payment module and the SEPA check program using the following parameter:

RestrictedCharSetEPC=1

Then the module converts during file creation all unallowable characters according to the rules of the EPC.

The **default value** is "0".

4.1.18 Section [COMMON] - Use alternative scheme (Austria)

Deviating from the normal handling the file creation for Austria can be controlled using the following parameter in the section [COMMON]. The parameter

WriteUNIFIforSTUZZA=1

effects, that on file creation STUZZA and rulebook version 6.0 the file is written in UNIFI format instead of the underlying format.

4.1.19 Section [COMMON] - Performance optimization for checking and preparation for the display

For the sending of large files, the data throughput can be enhanced for checking and the creation of the display file.

Since no further optimizations are possible, if all fields of the transaction level are checked comprehensively on application layer, a special mode was introduced, which behaves as follows:

1. Schema check: always the complete file
2. Check on application layer: only headers and batch level

In this special mode the transactions are not checked for valid characters, IBAN check digits, correct summation. For container files, also the hash values are not checked.

This process speeds up file checking by a factor of 4-7 depending on the file content. It is used automatically (**default**) for a file size **exceeding 15 MB for credit transfers** and **1 MB for direct debits**.

The values can be defined separately for credit transfers and direct debits using the following parameters in the SEPA.INI in section [COMMON] to take effect in the Protocol DLL (value = file size in MB):

Examples:

Credit transfers:

CTNoApplLayerCheckTxInf=15

Direct debits:

DDNoApplLayerCheckTxInf=1

Remarks:

1. Limit checks are then carried out exclusively on the basis of totals fields on batch level (only if available, otherwise as before).
2. Because the individual transactions are no longer read, this mode must not be activated for payment orders, if a limit check is configured for individual transactions.

4.1.20 Section [COMMON] - Mark unconfirmed records from IBAN conversion

According to the requirements of the MultiCash working group of banks in Austria, the conversion of bank code/account number into BIC/IBAN must necessarily be made using the STUZZA conversion file. Since this is not very user-friendly, MultiCash converts the data, but so far marked the results in the SEPA module with a question mark at the beginning to ensure, that the customer either

- a) creates the IBAN upload file and converts it via STUZZA or
- b) checks the data and removes the question mark.

The input in both steps was considered to be disproportionate, since in the most cases the automatic conversion is correct. Furthermore, the user must accept the disclaimer before the data import.

Therefore, starting with release 3.20.031 (as well as with Service Pack 2 for release 3.20.030), the processing has been changed as follows:

Bank code/account number are directly converted into BIC/IBAN (without question marks).

The processing can be changed to the old approach (with question marks) by setting the following parameter in the SEPA.INI configuration file in the [COMMON] section:

MARK_UNCONFIRMED_IBAN_CONVERSION=1

The **default value** in the payments module is "0".

4.1.21 Section [COMMON] - Differentiation between "hard technical" and "soft functional" errors

In the future, the quality of errors is distinguished by a new return code, so that the application can react differently:

RC 9: As before - serious technical errors that basically prevent further processing. Here, the processing is aborted.

RC 13: New - functional errors (such as incorrect IBAN, invalid characters, submission deadline for direct debit exceeded). Here, the processing will continue.

If this distinction should be used, then the following parameter must be set in the SEPA.INI in the section [COMMON]:

ActivateSoftError13=1

The **default value** for this parameter of the log DLL is "0".

Remark:

RC 13 is specially interpreted only by SmartOffice initially.

4.1.22 Section [COMMON] – Log level for e-mail sending (SPM module)

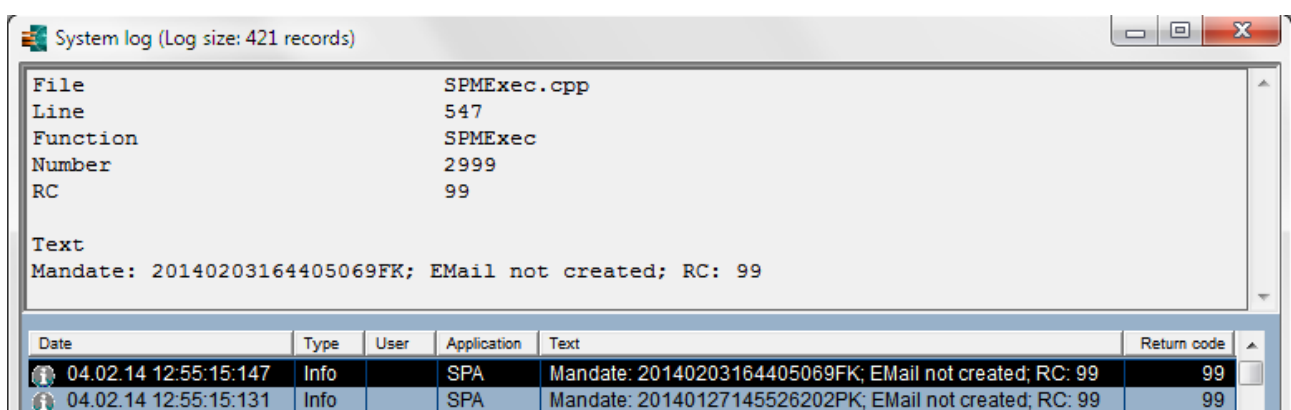
This parameter is associated with a command line tool for sending e-mail (additional program for MultiCash Transfer while using Mandate Administration Plus [module ID SPM]).

The log level can be controlled by the following parameter in the [COMMON] section of SEPA.INI:

MVPlusLogLevel=1

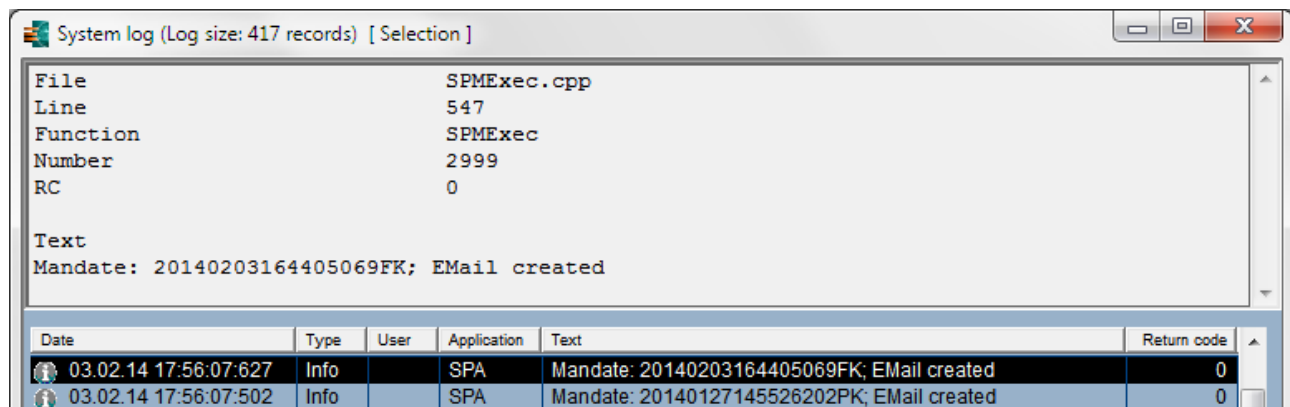
A value unequal to 0 means that only the negative cases are logged in the system log.

Example (negative cases in the system log):



The **default value** for this parameter is "0" (i.e., the positive cases are also logged).

Example (positive cases in the system log):



The command line tool SPMExec.exe searches the databases for open mandates and due pre-notifications and creates the corresponding messages. This call can be started from the command line or integrated in a MultiCash Transfer job, which cyclically searches the databases and generates the corresponding E-Mails.

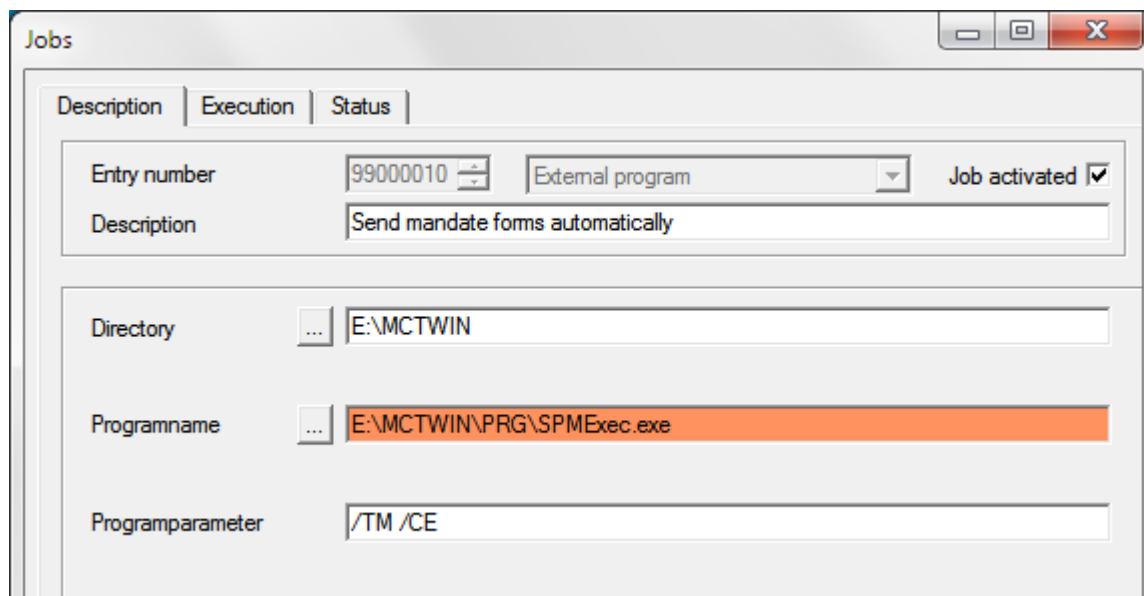
Call:

SPMExec.exe /T<MsgType> /C<Channel> /U<Unit>

MsgType Code, defining the type of letter
 M = Mandate request/Rededication letter
 P = Pre-Notification
 D = Letter concerning mandate deactivation after collection return

Channel Communication channel for the transmission of the message:
 E=E-mail
 L=Letter printout
 C=CSV output for the external generation of serial letters (firstly deferred)

Example (sending of mandate forms via e-mail using a MultiCash Transfer job):

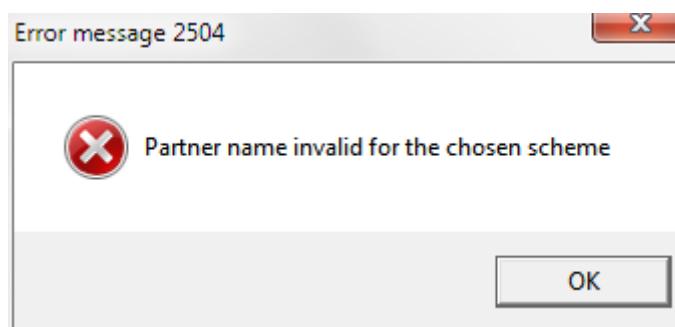


4.1.23 Section [COMMON] – Check urgent transfers for presence of the BIC

Using the following parameter in the section [COMMON] of the SEPA.INI, a check for the presence of a BIC can be activated, if urgent transfers are entered:

CCU_BICmand=1

If this parameter is set, the BIC for the partner becomes mandatory and is checked by the system. If the BIC is missing, an appropriate error message is displayed:



The **default value** for this parameter of the log DLL is "0" (no check).

4.1.24 Section [COMMON] – Only one batch per file

The aggregation of several logical files (so-called batches) in one payment file can be suppressed by the following parameter in the section [COMMON] of the SEPA.INI:

OneBatchperFile=1

If this parameter is activated, a separate physical file is created for each batch.

The **default value** for this parameter is "0", i.e. several logical files are written in one payment file.

4.1.25 Section [COMMON] – Activate format variant for RB4 additionally

Generally, you can restrict the support to the current and the previous format version.

As a fundamental change in Germany is pending, it was made possible, that in addition to the previous format version also RB4 is supported.

For this, the following parameter was introduced:

ACTIVATE_DKRB4=1

The **default value** for this parameter is "0"; it is only valid for the format sub-system.

4.1.26 Section [COMMON] – Add client ID to E2E reference

To keep the payment orders that are generated from different installations unique if applicable, the end-to-end reference can be supplemented with a client ID

For this the following parameter can be used:

ClientID=xxxx

The client ID entered here is appended at the end of the generated end-to-end reference.

Example:

New entry Single orders

Additional information | Direct debit additional | LDGR data | Times | Reversal

Payments | Partner | Ordering party / Internal | Ultimate parties

Order number: MCCT160808000001

End-To-End reference: E2EC160808000001A1B2

Access class: ?

Single order

Credit transfer

4.1.27 Section [COMMON] – Control of the checks for the money transfer regulation

The checks in the format sub-system for the new money transfer regulation (put into effect since 26.06.2017) can be switched off by activating this parameter (=1).

NOMTRCHECK=1

The **default value** for this parameter is "0"; the parameter is only valid in the format sub-system.

4.1.28 Mapping tables for reject information

The remaining sections of the SEPA.INI contain tables, which allow to translate coded information (group status code, transaction status code, status reason code) from reject files into textual information.

[GrpSts]

ACCP=AcceptedCustomerProfile
 ACCR=AcceptedCancellationRequest
 ACSC=AcceptedSettlementCompleted
 ACSP=AcceptedSettlementInProgress
 ACTC=AcceptedTechnicalValidation
 ACWC=AcceptedWithChange
 PART=PartiallyAccepted
 PDNG=Pending
 RCVD=Received
 RJCT=Rejected

[TxSts]

ACCP=AcceptedCustomerProfile
 ACCR=AcceptedCancellationRequest
 ACSC=AcceptedSettlementCompleted

ACSP=AcceptedSettlementInProgress
ACTC=AcceptedTechnicalValidation
ACWC=AcceptedWithChange
PDNG=Pending
RJCT=Rejected

[StsRsnCd]

AC01=IncorrectAccountNumber
AC03=InvalidCreditorAccountNumber
AC04=ClosedAccountNumber
AC06=BlockedAccount
AG01=TransactionForbidden
AG02=InvalidBankOperationCode
AM01=ZeroAmount
AM02=NotAllowedAmount
AM03=NotAllowedCurrency
AM04=InsufficientFunds
AM05=Duplication
AM06=TooLowAmount
AM07=BlockedAmount
AM09=WrongAmount
AM10=InvalidControlSum
BE01=InconsistentWithEndCustomer
BE04=MissingCreditorAddress
BE05=UnrecognisedInitiatingParty
BE06=UnknownEndCustomer
BE07=MissingDebtorAddress
DT01=InvalidDate
ED01=CorrespondentBankNotPossible
ED03=BalanceInfoRequested
ED05=SettlementFailed
MD01=NoMandate
MD02=MissingMandatoryInformationInMandate
MD03=InvalidFileFormatForOtherReasonThanGroupingIndicator
MD04=InvalidFileFormatForGroupingIndicator
MD06=RefundRequestByEndCustomer
MD07=EndCustomerDeceased
MS02=NotSpecifiedReasonCustomerGenerated
MS03=NotSpecifiedReasonAgentGenerated
NARR=Narrative
NOAS=NoAnswerFromCustomer
NOOR=NoOriginalTransactionReceived
RC01=BankIdentifierIncorrect
RC07=InvalidCreditorBICIdentifier
RF01=NotUniqueTransactionReference
TM01=CutOffTime

4.2 Configuration files for code lists SPACD*.INI

Using these configuration files code lists can be defined for Purpose Codes, Regular Reporting and Reversal Reason Codes.

4.2.1 Naming conventions

The code lists, whose values are not fix, can be filled individually using configuration files in dependency of the following values:

1. Field ID: to be filled depending on the respective data field
2. Scheme family: to be filled depending on the scheme family of the ordering party bank
3. Payment type: to be filled depending on the chosen payment type
4. Bank ID: to be filled from the BIC of the ordering party bank

For the configuration files in ..\MCCWIN\DAT, the following naming convention is defined:

SPACD-<Field ID>-<Bank ID>.INI

The individual segments are defined as follows:

Field ID:

Category Purpose:	CP	(payment Category Purpose code on batch level)
Purpose Code:	PC	(payment Purpose Code on transaction level)
Regulatory Reporting:	RR	(Regulatory Reporting)
Reason Code:	RC	(Reversal Reason Codes for Reversals [pain.007])

Bank ID:

The bank ID is represented by the first 4 or 6 places of the BIC. Thus a bank can define individual code lists for all their different countries or for individual countries.

The **access** to the configuration files is effected then in a descending refinement level as follows:

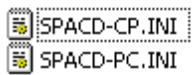
1. SPACD-FF-BBBBCC.INI (BBBB=Bank, CC=Country)
2. SPACD-FF-BBBB.INI
3. SPACD-FF.INI

If no code list is found or if the appropriate field is deactivated in the INI file found, the field is not displayed in the dialog.

Examples:

SPACD-CP-DRESDE.INI:	Code list für Category Purpose of the Dresdner Bank Germany
SPACD-PC-COBA:	Code list für Purpose Code of the Commerzbank generally
SPACD-RR:	Code list for Regulatory Reporting
SPACD-RC:	Code list für general Reason Codes

Example:



4.2.2 Structure

In the configuration files code lists lists can be filled individually depending on the following values:


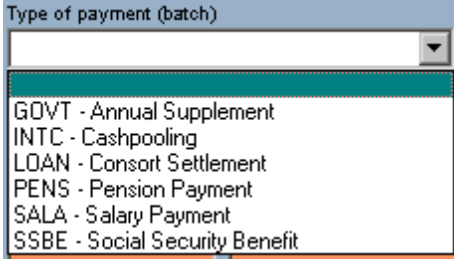
Scheme family:

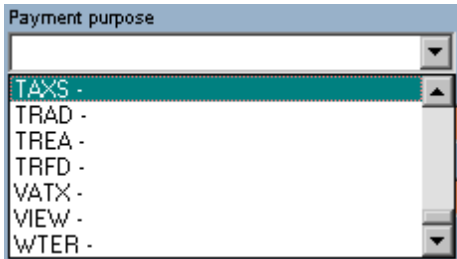

General	00
ZKA-Container	01
ZKA	02
[ExtendedGrpOptions]	03
MultiCash	04
STUZZA	05
MultiCash-CZ	06
UNIFI 2006	07
MultiCash-HU	08
MultiCash-CEE	09
MultiCash-PL	10
MultiCash-SK	11
UNIFI 2009	12
SEPA CH	13
SEPA Italy	14

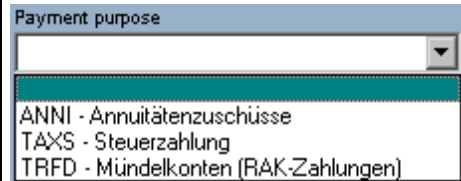
Payment type:

General:	00
Credit transfer:	01
Normal direct debit:	02
B2B direct debit:	03
Urgent transfer:	04

The syntax of the files is as follows:

<pre>[GENERAL] OFF=1</pre>	<p>General settings can be defined with this section. If this entry is present and the value is set to 1, the field is switched off.</p> <p>Our example:</p> <p>CP field generally switched off in SPACD-CP.INI:</p> <pre>[GENERAL] OFF=1</pre> 
<pre>[GENERAL-<Scheme family>-<Payment type>] OFF=0</pre>	<p>With this section a general setting for a special scheme family and/or a payment type can be set.</p> <p>Examples: [GENERAL-01-01] – ZKA format, credit transfers [GENERAL-00-01] – for all credit transfers [GENERAL-01-00] – ZKA format generally</p> <p>Our example:</p> <p>Special keywords (see below) in SPACD-CP.INI for Austria (STUZZA=05) generally switched on:</p> <pre>[GENERAL-05-00] OFF=0</pre> 
<pre>[KEYWORDS] <Keyword>=1</pre>	<p>If an entry is present and the value is set to 1, this keyword is displayed in the appropriate selection list.</p> <p>Example for a PC field: TAXS=1</p> <p>Example:</p>

	<p>Keywords generally switched on (see above):</p> <pre>[KEYWORDS] TAXS=1 TRAD=1 TREA=1 TRFD=1 VATX=1 VIEW=1 WTER=1</pre> <p>Display (no text file available, see Chapter 1.3):</p> 
<p><Keyword>=<Field ID>-<Keyword></p>	<p>If another field ID and a keyword valid there is set for an entry, the appropriate keyword is only displayed in dependence of the selection made there.</p> <p>Example for a PC field in Austria, which depends on a code in a CP field, e.g.:</p> <p>ATBL=CP-SALA</p> 
<p>[KEYWORDS-<Scheme family>-<Payment Type>]</p>	<p>With this section a special code list for one scheme family and/or one payment type can be set.</p> <p>Examples:</p> <p>[KEYWORDS-01-01] – ZKA format, credit transfers</p> <p>[KEYWORDS-00-01] – for all transfers</p> <p>[KEYWORDS-01-00] – ZKA format generally</p> <p>Example:</p>

	<p>Special PC keywords switched on (see above) for Austria:</p> <p>[KEYWORDS-05-00] ANNI=1 TAXS=1 TRFD=1</p> <p>Display (with text, see Chapter 1.3):</p>  <p>The screenshot shows a window titled 'Payment purpose' with a dropdown menu. The menu is open, displaying three items: 'ANNI - Annuitätenzuschüsse', 'TAXS - Steuerzahlung', and 'TRFD - Mündelkonten (RAK-Zahlungen)'. The 'TRFD' item is highlighted with a red border.</p>
--	---

4.2.3 Example

Example configuration file for Reversal Reason Codes SPACD-RC.INI

```
[GENERAL]  
OFF=0
```

```
[KEYWORDS]  
AC01=0  
AC04=1  
AC06=0  
AG01=0  
AG02=1  
AM01=0  
AM02=0  
AM03=0  
AM04=0  
AM05=1  
AM06=0  
AM07=0  
AM09=0  
AM10=0  
BE01=0  
BE04=0  
BE05=0  
BE06=0  
BE07=0  
DT01=0  
ED01=0  
ED03=0  
ED05=0  
MD01=1  
MD02=0  
MD03=0  
MD04=0  
MD05=1  
MD06=0  
MD07=0  
MS02=1  
MS03=1  
NARR=0  
RC01=0  
RF01=0  
TM01=1
```

4.3 Configuration files for code list texts SPACDTEXT*.INI

Using these configuration files appropriate texts can be defined for the codes concerning Purpose Codes, Regular Reporting and Reversal Reason Codes.

4.3.1 Naming conventions

In order to display a correct text for each code, appropriate texts in different languages can also be stored in configuration files:

SPACDTEXT-<Field ID>-<Language>.INI

The individual segments are defined as follows:

Field ID:

Category Purpose:	CP
Purpose Code:	PC
Regulatory Reporting:	RR
Reason Code:	RC

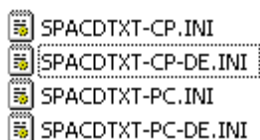
Language:

ISO Country Code

Example:

SPACDTEXT-RC.INI:	Default texts for the Reason Code field (in English)
SPACDTEXT-CP.INI:	Default texts for the Category Purpose field (in English)
SPACDTEXT-CP-FR.INI:	Texts for the Category Purpose field in French

Example:



4.3.2 Structure

The syntax of the files is as follows:

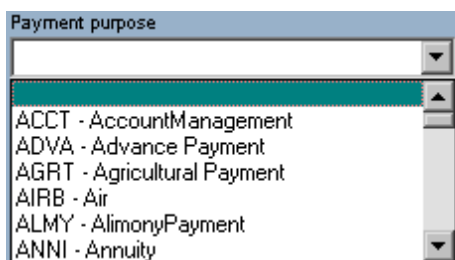
[KEYWORDS] <Keyword>=<Text>	If an entry is present, the assigned text is displayed in the respective selection list. Example: TAXS=Tax payment
[KEYWORDS-<Scheme family>]	With this section a special text list for one scheme family can be set. Example: [KEYWORDS-05] RLWY=Bezüge OeBB

Example:

Payment purpose from SPACDTEXT-PC.INI (97 keywords at the moment):

```
[KEYWORDS]
ACCT=AccountManagement
CASH=Cash management transfer
COLL=CollectionPayment
INTC=IntraCompanyPayment
LIMA=LiquidityManagement
NETT=Netting
AGRT=Agricultural Payment
...
```

Display in the payment entry:



Special payment purpose for scheme family 05 (=Austria) from SPACDTEXT-PC.INI (9 keywords at the moment):

```
[KEYWORDS-05]
ATBA=Bezüge Arbeitslose
ATBB=Bezüge allgemein
ATBL=Bezüge Landesinvaliden
ATBV=Bezüge Landesverteidigung
ATPB=Bezüge Post
```

RLWY=Bezüge/Pensionen OeBB
ANNI=Annuitätenzuschüsse
...

Display in the payment entry:



4.3.3 Example

Configuration files SPACDTEXT-RC.INI and SPACDTEXT-RC-GB.INI for Reversal Reason Code texts:

```
[Keywords]
AC01=Incorrect Account Number
AC04=Closed Account Number
AC06=Blocked Account
AG01=Transaction Forbidden
AG02=Invalid Bank Operation Code
AM01=Zero Amount
AM02=Not Allowed Amount
AM03=Not Allowed Currency
AM04=Insufficient Funds
AM05=Duplication
AM06=Too Low Amount
AM07=Blocked Amount
AM09=Wrong Amount
AM10=Invalid Control Sum
BE01=Inconsistent With End Customer
BE04=Missing Creditor Address
BE05=Unrecognised Initiating Party
BE06=Unknown End Customer
BE07=Missing Debtor Address
DT01=Invalid Date
ED01=Correspondent Bank Not Possible
ED03=Balance Info Requested
ED05=Settlement Failed
MD01=No Mandate
MD02=Missing Mandatory Information In Mandate
MD03=Invalid File Format For Other Reason Than Grouping Indicator
MD04=Invalid File Format For Grouping Indicator
MD05=Collection Not Due
MD06=Refund Request By End Customer
MD07=End Customer Deceased
MS02=Not Specified Reason Customer Generated
MS03=Not Specified Reason Agent Generated
NARR=Narrative
RC01=Bank Identifier Incorrect
RF01=Not Unique Transaction Reference
TM01=CutOffTimeTM01=CutOff Time
```

4.3.4 Concrete example: Special keywords for Germany

For VWL (=Vermögenswirksame Leistungen) payments (=Capital building fringe fortune) in Germany a new Purpose Code was requested. It is defined in the configuration files by the following:

```
[KEYWORDS]
CBFF=Capital building fringe fortune
```

SPACDTEXT-PC-DE.INI:

```
[KEYWORDS]
CBFF=Vermögenswirksame Leistungen
```

4.3.5 Concrete example: Special keywords for Austria

The keywords defined particularly defined for Austria are stored as follows:

For the field "Code Purpose" on batch level:

SPACDTEXT-CP.INI:

```
[KEYWORDS-05]
SALA=Salary Payment
PENS=Pension Payment
LOAN=Consort Settlement
SSBE=Social Security Benefit
GOVT=Annual Supplement
INTC=Cashpooling
```

SPACDTEXT-CP-DE.INI:

```
[KEYWORDS-05]
SALA=Gehaltszahlungen
PENS=Pensionszahlungen
LOAN=Konsortialverrechnungen
SSBE=Sozialversicherungszahlungen
GOVT=Annuitätenzuschüsse
INTC=Cashpooling
```

For the field "Purpose Code" on transactional level:

SPACDTEXT-PC.INI:

```
[KEYWORDS-05]
ANNI=Annuitätenzuschüsse
ATBA=Bezüge Arbeitslose
```

ATBB=Bezüge allgemein
ATBL=Bezüge Landesinvaliden
ATBV=Bezüge Landesverteidigung
ATPB=Bezüge Post
RLWY=Bezüge/Pensionen OeBB
TRFD=Mündelkonten (RAK-Zahlungen)
TAXS=Steuerzahlung

SPACDTEXT-PC-DE.INI:

```
[KEYWORDS-05]  
ANNI=Annuitätenzuschüsse  
ATBA=Bezüge Arbeitslose  
ATBB=Bezüge allgemein  
ATBL=Bezüge Landesinvaliden  
ATBV=Bezüge Landesverteidigung  
ATPB=Bezüge Post  
RLWY=Bezüge/Pensionen OeBB  
SALA=Gehaltszahlungen  
TRFD=Mündelkonten (RAK-Zahlungen)  
TAXS=Steuerzahlung
```

The reduced keyword list to fit specific Austrian needs for the field "Code Purpose" is defined by the following special sections in the configuration file SPACD-CP.INI:

```
[GENERAL-05-00]  
OFF=0  
  
[KEYWORDS-05-00]  
GOVT=1  
INTC=1  
LOAN=1  
PENS=1  
SALA=1  
SSBE=1
```

The dependencies between the keywords of the two levels are stored in the following section in the configuration file SPACD-PC.INI:

```
[KEYWORDS-05-00]  
ANNI=CP-GOVT  
ATBA=CP-SALA  
ATBB=CP-SALA  
ATBL=CP-SALA  
ATBV=CP-SALA  
ATPB=CP-SALA  
RLWY=CP-SALA  
RLWY=CP-PENS  
SALA=CP-SALA
```

4.4 Configuration files for mandate defaults SPAMD*.INI

On data migration of direct debit orders from other payments modules into the SEPA payments module, the direct debit data can be used to automatically generate appropriate mandates from the existing data, which may still need to be completed afterwards.

4.4.1 Naming conventions

By placing INI files in the ..\MC?WIN\DAT directory you can specify mandate defaults for the generated mandates (SEPA mandate defaults). The rules are controlled globally or separately for different creditors by an appropriate naming convention:

General: **SPAMDDEF.INI**

For each creditor ID: **SPAMDDEF-<Creditor ID>.INI** (has priority on access)

4.4.2 Structure

The files have the following structure:

<pre>[GENERAL] MandateID=0 1 2 3 4 5</pre>	<p>The mandate ID predefinition can be specified as follows:</p> <p><i>Starting with release 3.20.031:</i> 0 – The mandate ID is filled with a temporary value (@<YYYYMMDDHHMMSSTTT>), which need to be modified manually. This invalid mandate reference allows the user to save a mandate, even if the reference is not yet known. At the same time, it is preventing that the mandate will be used unintentionally.</p> <p>1 – Timestamp with the structure: YYYYMMDDHHMMSSTTT (Default)</p> <p>2 – Partner IBAN (suitable for a general mandate [in German: "Rahmenmandat"])</p> <p><i>Starting with release 3.20.031:</i> 3 – Global ascending number</p> <p><i>Starting with release 3.20.031:</i> 4 – Ascending number per ordering party</p> <p><i>Starting with release 3.20.034:</i> 5 – No predefinition</p>
--	---

MandateIDNumericLength=0 1 2	<p>This entry is used to define the length of numbers:</p> <p><i>Starting with release 3.20.031:</i></p> <p>0 – 8 digits with leading zeroes (Default) 1 – no leading zeroes 2 – 12 digits with leading zeroes</p>
Signature=<TT.MM.JJ>	Date of signature (Default: 09.07.2012)
Type1=<instrument qualifier>	<p>Type of direct debit / mandate:</p> <p>CORE, B2B</p>
Type2=<usage qualifier>	<p>Type of mandate frequency:</p> <p>OOFF - once RCUR, FRST, FNAL – recurrent</p>
Usage=0 1 2	<p>Identifier relating to the utilization:</p> <p><i>Starting with release 3.20.031:</i></p> <p>0 = still not utilized (Default) 1 = utilized exactly once 2 = already utilized several times</p>
CONTRACT_ID_CRED_PLUS_DEBTOR_BBAN=0 1	<p>Mandate generation from contract of customer and partner bank/account*</p> <p><i>Starting with release 3.20.031:</i></p> <p>Hereby, the special processing with contract access via partner bank/account and ordering party ID can be enabled for mandate creation and payment conversion: 1 (Default is 0)</p>

* Objective:

This processing option allows the generation of mandates from DTAUS (or CSV) files with subsequent use by the converter, without having the mandate or contract information derived from the details of the DTAUS files.

The aim is to achieve a reliable conversion with adding mandate data **without any modification in the details** of the DTAUS files. The mandate allocation is determined from the relation between ordering party (creditor) and the account of the payer (debtor).

Remark:

Can only be used, if there is only one contract with the debtor or a general mandate is used for several contracts.

Processing on mandate generation from import of DTAUS or CSV files:

If the appropriate processing mode is activated, a contract record for this mandate is generated additionally on importing the mandate data, with the **contract number** to be filled as follows:

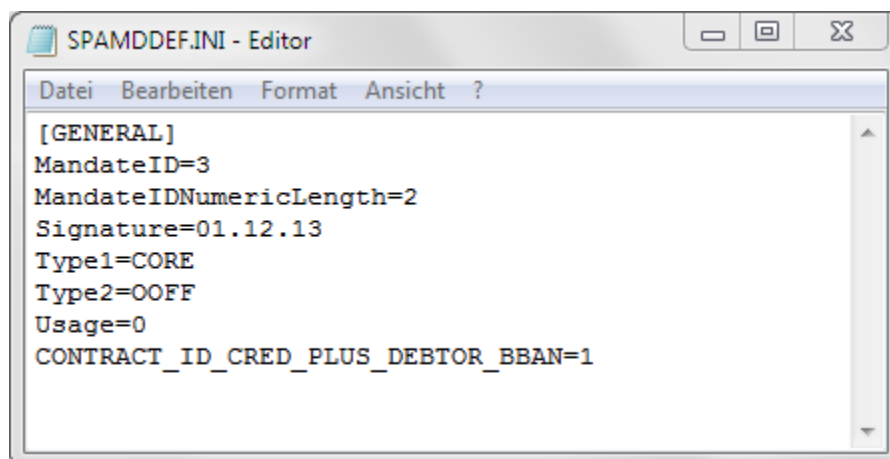
<Bank code+Debtor account number><Ordering party ID>

Processing on conversion:

If the appropriate processing mode is activated, the converter does not interpret the details of the DTAUS file, but determines the **mandate ID** as follows:

1. Finding the ordering party ID via reference table of the SEPA module by name
2. Generation of the contract number as described above
3. Determining the mandate via contract table

4.4.3 Example



4.5 Rules for the detection of R transactions using INFO-MAPPING.INI

The cash management module is able to extract defined transaction information.

The rules for this can be defined in the können INFO-MAPPING.INI configuration file with the following general structure:

```
[DEFAULT]
Modules=<MMM>    [<MMM> [...]]

[COMMON-<MMM>]
STC=ccc[ccc [...]]    3-digit SWIFT Transaction Code or empty
GVC=ggg    [ggg [...]]    3-digit Business Type Code (=GVC in German) or empty
Details=dd...dd        any character string in the details or empty
Partner=dd...dd        any character string in the partner name

[BANK-XXX xx...xx]    xx.xx = up to 12-digit bank ID
[ACCT-XXX xx...xx yy...yy]    yy.yy = up to 24-digit account number
```

In addition to the general rules, bank- or account-related rules can be entered, with account-related definitions having priority over bank entries of same bank.

Especially for the detection of R transactions by the SEPA module a separate section is specified in the INFO-MAPPING.INI configuration file.

The SEPA direct debit R transactions for Germany can be detected using the relevant business transaction code (BTC=GVC in German). On this basis, the following is defined in the configuration file.

Example:

```
[DEFAULT]
Modules=SPA

[COMMON-SPA]
GVC=108 109 159 181 184
```

```
GVC 108 : SDD (debits; reversal debit, B2B)
GVC 109 : SDD (debits; reversal debit, Core)
GVC 159 : SCT (credits; reversal)
GVC 181 : SDD (credits; re-credited amount, Core)
GVC 184 : SDD (credits; re-credited amount, B2B)
```

4.6 Deactivation of mandates using MANDATE-STOP.INI

From the list of the return reasons for SEPA direct debits, those codes were transferred to the MANDATE-STOP.INI configuration file, which lead to the deactivation of a mandate (alphanumeric SEPA Reason Code or numeric text code supplement).

Example:

```
[COMMON]
AC01=Incorrect Account Number
AC04=Closed Account Number
AC06=Blocked Account
AG01=Payment Type Not Allowed
MD01=No Valid Mandate
MD07=End Customer Deceased
901=Incorrect Account Number
902=ClosedAccountNumber
903=BlockedAccount
904=Payment Type Not Allowed
909=No Valid Mandate
913=End Customer Deceased
```

Explanation of the return reason codes for SEPA direct debits leading to mandate deactivation:

AC01/901 : Account identifier incorrect (e.g. invalid IBAN)
AC04/902 : Account closed
AC06/903 : Account blocked for direct debits by the debtor
AG01/904 : Transaction forbidden
MD01/909 : No valid mandate
MD07/913 : End customer deceased

5 Configuration options on bank side (Omikron bank server)

5.1 Configuration via SEPA.INI



Please note:

The bank server uses default values. Only if e.g. the acceptance of the files generated by the different procedures should be limited **on Omikron bank server side**, a SEPA.INI file has to be created there in the ..\ADMIN\DAT directory (a SEPA.INI is not provided!).

In analogy to the parameters described in Chapter 1.1 for the customer side, you can specify e.g. the allowed procedures then.

5.1.1 Section [COMMS_ALLOWED] – Allowed files

The entry, which provides all procedures available in the client system, is e.g. as follows:

```
[COMMS_ALLOWED]
NUMBERS=10
COMMS1=Container
COMMS2=ZKA
COMMS3=Extended Grouping Options
COMMS4=MultiCash
COMMS5=STUZZA
COMMS6=MultiCash-CZ
COMMS7=MultiCash-HU
COMMS8=UNIFI
COMMS9=MultiCash-CEE
COMMS10=MultiCash-PL
```

5.1.2 Section [DEL_LEADING_ZERO] - Handling of leading zeroes (CZ, SK only)

Using this entry leading zeroes can be removed from bank codes extracted from IBAN respectively (relevant for CZ and SK):

```
[DEL_LEADING_ZERO]
DELETE=1
```

effects, that leading zeroes before the bank codes are removed.

[DEL_LEADING_ZERO]
DELETE=0

effects, that leading zeroes before the bank codes remain (**Default**).

5.1.3 Section [TRACE] - Tracing

Using the following entry, problems in the context of XML (so-called exceptions) can be traced.

SEPAXML=0

means, that the trace file is not written (**Default**).

SEPAXML=1

effects, that the trace file (SEPAXML.PTK) is written.

Using a further entry, events can be traced, which are only relevant for bank servers of the version 3.02.

P32=0

means, that the trace file is not written (**Default**).

P32=1

effects, that the trace file (SEPA.PTK) is written. In detail it is traced, at which point the SEPA subsystem returns the return code for "file structure incorrectly".

5.1.4 Section [SessiontypCheck] – Check for assumed file content

If the following parameter is set as follows, a check is made, whether that, what can be assumed from the session type, is included in the file. That is e.g. CCC: Container transfers, CDD: Core direct debits, CD1: COR1 direct debits, CCT: Transfers etc.

[SessiontypCheck]
NoCheck=No

By setting the parameter to "Yes", the check can be deactivated consequently.

Remark:

We recommend that you should not change the parameter without consultation with Omikron.

By **default** nothing is used.

5.1.5 Section [COMMON] - New format sub-system under bank server 3.21

After installation of the new format sub-system under MCB 3.21 the function import/export is no longer available.

In order to restore the functionality between the new SEPA format sub-system and the old file interface the following need to be added in the SEPA.INI in MCB 3.21:

FileFlag=1

The **default** is "0".

5.1.6 Section [COMMON] - Switch off IBAN and CIN check

Usually (the **default** is "0") IBAN and creditor identification number (CIN) are checked for validity and files with format errors are rejected. In order that these files can be accepted nevertheless, because e.g. individual sentences are post-processed in the back office of the bank manually, the check can be switched off by the following entry in the section [COMMON]:

DONTCHECKIBAN=1

5.1.7 Section [COMMON] - Limit acceptance to the last 2 scheme versions

Using a parameter you can define that the format subsystem accepts only the two last SEPA variants. For it the following switch must be set in the section [COMMON]:

OnlyLast2VerAllowed=1

The **default** is "0".

5.1.8 Section [COMMON] – Control of the checks for the money transfer regulation

The checks for the new money transfer regulation (put into effect since 26.06.2017) can be switched off by activating this parameter (=1).

NOMTRCHECK=1

The **default value** for this parameter of the format sub-system is "0".

5.1.9 Section [DD_TIME_RANGES] - Submission deadlines for direct debits / Check

The entries in this section control the submission deadlines for SEPA direct debits (DD).

In order to ensure the submission in time the following lead time values are defined as default . Examples (the values given for CORE1, CORE2, COR1 and B2B are also the **default values** in the payments module and in the log DLL):

```
[DD_TIME_RANGES]
CORE1=7
CORE2=4
COR1=3
B2B=3
ChkDDTimeRanges=0 (Default is "0")
CutOffTime=15:00 (Default is "12:00")
ReqdColltnDtModification=0 (Default is "0")
```

Using this, the submission deadlines can be adapted by the banks if need be.

The further parameters:

The control of the submission times is activated by setting the parameter

ChkDDTimeRanges=1

(Deactivate: set to "0").

The **default value** in the log DLL is "0" (i.e. the deadline check is deactivated by default starting from release 3.20.031), since most of the banks also accept SEPA direct debits, if the submission deadline has been exceeded.

Since the submission of SEPA direct debits in time is very important, on determining the lead time the current time is to be considered. If the determination is carried out after cut-off time, the count of the days for the lead starts at the following day.

The cut-off time to be considered is given behind the parameter `CutOffTime=..`.
Default cut-off time is 15:00.

If the due date should be modifiable, this can be made by setting the parameter

ReqdColltnDtModification=1 enabled (only in the context of a special solution, please do not modify!).