TOPAL XML DESCRIPTION

Technical guide

Version: 4.1

Table of Contents

General Information

Topal includes an XML import and export wizard. You may export data from a Topal Database to an XML file, as well as import from an XML file, giving you the choice to:

- Migrate Data from external Database (e.g. another Accounting Software)
- Merge Data (export data from one client and import to another)
- Insert Data from applications developed by third parties

This document describes the Topal XML structure and explains you how to create your own XML document that you can import to Topal. It also includes a description of entities (tables) and some specific information for third party developers.

We are continually improving our software. Some changes may impact the Topal Database with new attributes (fields) and so XML data structure may change, too. These changes will be documented in release notes.

1. Data Structure

In this chapter we describe the Data Structure of Topal, how it is organized and how you can use it.

Although it is not essential to read this chapter, the information here will help you to understand the process of data creation, and as a result, it will help you using Topal Data Migration System.

1.1. Data Organization

Topal uses the relational model as database scheme (the database's structure of how the data is arranged), which represents all information in the form of multiple related tables each consisting of rows and columns. This model represents relationships by the use of values common to more than one table.

That means the Topal Database is organized in tables that are related between each others. Each record within a table is identified by a unique identifier (ID), which means the value of this field (key) in this table cannot be repeated, it is unique. In some cases several fields are used to identify uniquely a record (composed key).

1.1.1. Internal ID rules

As mentioned above each table has a special ID field, or in some cases, several ID fields. We use, additionally to the standard rules for relational Databases, some special conventions for this ID fields (columns):

- As mentioned above all values in ID field should (must) be original, this means must not be repeated for one table.
- Type for ID fields must be int (integer) or Guid (unique identifier).
- We used the following name convention for ID fields; ID field is composed by table name followed by "ID". For example in entity "Account" ID field name is "accountID", in entity "Client" ID field name is "clientID".

- ID type int cannot be less or equal zero, ID type Guid cannot be empty ‘00000000-0000-0000-0000-000000000000’.

1.1.2. Invalid values

All values of data record must not be equal to null (DBNull).

However, this is not critical for end users, as Topal System solves this problem by using defaults (mentioned below).

1.2. Data Relation

As mentioned above Topal System has a relation data structure. That means most tables can be and are linked to others by relations. These relations are defined by linking ID fields of one table with fields of another table. The field or fields in this other record is called Foreign Identifier (FID) or foreign key. Remember, that ID fields identify unambiguously a record.

1.2.1. FID Rules

As for ID fields, there are also some special rules and conventions for foreign ID fields:

- The FID may not be unique because several records can link to same entity.
- As well as ID fields, FID type must be int (integer) or Guid (unique identifier), and of course they must be of the same type the corresponding ID filed is
- Again, we use the following name convention for FID fields: Table name plus FID, For example, table “FiscalYear” has a link to table “Client”, so the field name, that contains a link to a record of “Client”, is “clientFID”.
- The FID type should be the same type as the ID type of the linked entity.

1.2.2. Empty FID

Some of the relations are not mandatory; that means the FID field may not contain a link to another record (in this case the field is empty).

For example in the posting record we have FID to “VAT”. The field in “Posting” which contains this link is called as “vatFID”, but some posting do not contain VAT, so this FID would be empty.

It is impossible to set null value to FID (as mentioned in 2.1.2.).

However, it is solved in another way. As mentioned above, FID has the same type as ID (int or GUID) with restriction for values ID. So for empty values in FID fields, these values must be -1. In case of int and '00000000-0000-0000-0000-000000000000' in case of GUID.

1.3. Defaults

As already mentioned, all fields should have a value and must not be null, but it is not necessary to assign all the fields. Some fields are not mandatory and in most of the cases, their value is the same for all the records. For example "Posting" record has a field "freeCode". If no value is assigned then the default value will be used (in current example that would be an empty string).

Essentially, the d*efault value is the value of the field with which field is initialized.*

2. Entities Description

This chapter describes the entities (tables), which can be migrated (exported/imported).

In the row over the table header of the description table, you can find the general description of the entity.

The following columns are used in the description table:

- "Pos" - contains number of the field in the entity.
- "Field Name" – name of the field, which is used in XML.
- "Field Description" – description
- "Type" – field type.
- "Req." – shows whether field is optional (opt.), required (req.) or not definable by the user (none). The "opt. /req." value means that field is required in special cases and in other cases optional. For example vatFID is required if it is a VAT posting. In other cases it is an optional field.
- "Min" – contains minimum length for string fields.
- "Min" – contains maximum length for string fields.

- "Default" – Here we describe all available fields. It is not necessary do define them in XML.

2.1. Client

Accounting Clients list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	backupTS	Reserved	datetime	opt.			01.01.1753 0:00:00
2	buildNumber	Build number on which client was created	string	opt.	0	128	0.0.0.0
3	city	Client city	string	opt.	0	22	empty string
4	clientID	Client unique identifier	int	req.			-1
5	code	Client code	string	opt.	0	10	empty string
6	creationDate	Date of client creation	datetime	opt.			current date
7	isDiscountSwissRounded	Shows will discount be Swiss rounded(true = 1) or not (false = 0)	bit	opt.			True
8	isKCSwissRounded	Shows key currency amounts will be Swiss rounded(true = 1) or not (false = 0)	bit	opt.			True
9	isTemplate	Shows is this client template(true = 1) or not (false = 0)	bit	opt.			True
10	isVATSwissRounded	Shows VAT will be Swiss rounded(true = 1) or not (false = 0)	bit	opt.			True
11	lastTimestamp	Reserved	datetime	opt.			01.01.1753 0:00:00
12	modifyDate	Date of last instance modifying	datetime	opt.			current date
13	name	Client name	string	req.	1	35	empty string
14	person	Client person	string	opt.	0	22	empty string
15	remarks	Remarks to client	string	opt.	0	40	empty string
16	street	Client address street	string	opt.	0	35	empty string
17	telephone	Client telephone	string	opt.	0	22	empty string
18	userFID	Link to user which edit instance last time	int	opt.			-1
19	vatAgreed	Reserved	bit	opt.			True
20	vatNumber	Client VAT number	string	opt.	0	18	empty string
21	vatPaymentBased	Reserved	bit	opt.			False
22	vatRequired	Is VAT required for postings	bit	opt.			True
23	vatSimplefied	Is VAT calculations simplified	bit	opt.			False
26	xmlBackupState	Default state when XML backup processed (0 - Do not backup, 1 - Backup when changed, 2 - Backup always)	int	opt.			1
27	zip	Zip of client	string	opt.	0	35	empty string

2.2. IsDefault

Default items per client

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientConfig	Archived XML Client config (handled by the system)	image	none			empty
2	clientFID	Link to Client	int	req.			-1
3	currPartyNum	Current Party Number	int	req.			1
4	currPurchaseInvoiceNum	Current Purchase Invoice Number	int	req.			1

5	currReminderNum	Current Reminder Number	int	req.	1
6	currSalesInvoiceNum	Current Sales Invoice Number	int	req.	1
7	defAccrualAccountFID	Default account which used in accruals recurrent transactions	int	opt.	-1
8	defaultID	Default unique identifier	int	opt.	-1
9	defAutoDiscountAmount	Default discount amount	double	opt.	0
10	defBankAccountFID	Link to default Client's Internal Bank Account	int	opt.	-1
11	defBudgetPeriodFID	Kind of default budget representation (4 = Quarters, 8 = Months)	int	opt.	-1
12	defCountryFID	Link to Client's Country	int	req.	-1
13	defCreditorAccountFID	Link to default creditor Account (used in Payment(Transaction) Method)	int	opt.	-1
14	defCreditorDiscountAccountFID	Link to creditor discount Account	int	opt.	-1
15	defCreditorWriteOffAccountFID	Link to creditor write Off Account	int	opt.	-1
16	defCreditPaymentTypeFID	Link to default credit payment type	int	req.	-1
17	defCurrencyFID	Link to Client's Currency	int	req.	-1
18	defDebitPaymentTypeFID	Link to default debit payment type	int	req.	-1
19	defDebtorAccountFID	Link to default debtor Account (used in Payment(Transaction) Method)	int	opt.	-1
20	defDebtorDiscountAccountFID	Link to debtor discount Account	int	opt.	-1
21	defDebtorWriteOffAccountFID	Link to debtor write Off Account	int	opt.	-1
22	defDeferralAccountFID	Default account which used in deferrals recurrent transactions	int	opt.	-1
23	defFeeAccountFID	Link to default fee Account	int	opt.	-1
24	defLanguageFID	Link to Client's Language	int	req.	-1
25	defMoneyAccountFID	Link to default money Account	int	opt.	-1
26	defNoReminderAmount	Maximum Amount with which invoices do not proposed to reminders	double	opt.	0
27	defPartyCountryFID	Default Country for Party	int	opt.	-1
28	defPartyLanguageFID	Default Language for Party	int	opt.	-1
29	defPaymentTermFID	Link to default Payment Term	int	req.	-1
30	defRecurrentDateFrom	Default begin date of recurrent transaction	datetime	opt.	current date
31	defRecurrentDateUntil	Default end date of recurrent transaction	datetime	opt.	current date
32	defRepLanguageFID	Default Report Language	int	opt.	-1
33	inclAddressInReport	Whether the client address included to reminder report	bit	opt.	True
34	isLeftPosition	Position of text in reminder reports	bit	opt.	True
35	useCreditorFreeInvoiceNum	Wether the free invoice number should be used in payables reporting	bit	opt.	False
36	useCreditorFreePartyNum	Wether the free party number should be used in payables reporting	bit	opt.	False

37	useDebtorFreeInvoiceNum	Wether the free invoice number should be used in receivables reporting	bit	opt.			False
38	useDebtorFreePartyNum	Wether the free party number should be used in receivables reporting	bit	opt.			False

2.3. FiscalYear

Fiscal year list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	closedFlag	Shows is FiscalYear closed	bit	opt.			False
3	dateFrom	FiscalYear date from	datetime	req.			current date
4	dateUntil	FiscalYear date until	datetime	req.			current date
5	defBudgetFID	Link to default budget	int	opt.			-1
6	defCompoundNumRangeFID	Default number range of Compund Transction	int	opt.			-1
7	defDocumentNumberFID	Link to default Number Range	int	req.			-1
8	defESRPaymentNumRangeFID	Default number range of ESR Payment	int	opt.			-1
9	defPurchaseInvoiceNumRangeFID	Default number range of Purchase Invoice	int	opt.			-1
10	defPurchaseManualNumRangeFID	Default number range of Purchase Manual Payment	int	opt.			-1
11	defPurchasePaymentRunNumRangeFID	Default number range of Purchase Payment Run	int	opt.			-1
12	defSalesInvoiceNumRangeFID	Default number range of Sales Invoice	int	opt.			-1
13	defSalesManualNumRangeFID	Default number range of Sales Manual Payment	int	opt.			-1
14	defSalesPaymentRunNumRangeFID	Default number range of Sales Payment Run	int	opt.			-1
15	defSimpleNumRangeFID	Default number range of Simple Transction	int	opt.			-1
16	defValuationDate	Default valuation date	datetime	opt.			current date
17	docdateFrom	Date from which transactions can be created in FiscalYear	datetime	req.			current date
18	docdateUntil	Date until which transactions can be created in FiscalYear	datetime	req.			current date
19	fiscalYearID	FiscalYear unique identifier	int	req.			-1
20	modifyDate	Date of last instance modifying	datetime	opt.			current date
21	name	FiscalYear name	string	opt.	1	24	empty string
22	resultAccountFID	Link to result Account	int	opt.			-1
23	userFID	Link to user which edit instance last time	int	opt.			-1

2.4. NumRange

Number range for transactions

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	code	NumRange code	string	req.	1	16	empty string
2	currNum	NumRange current number	int	req.			1
3	fiscalYearFID	Link to Fiscal Year	int	req.			-1
4	name	NumRange name	string	opt.	1	48	empty string
5	numRangeID	NumRange unique identifier	int	req.			-1
6	startNum	NumRange start number	int	req.			1

2.5. DefaultText

Default tests for postings of different transaction types

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	defaultTextID	DefaultText unique identifier	uniqueidentifier	req.			empty guid
3	jokerTypeFID	Type of joker(Invoice = 1, Payment = 2, PaymentRun = 3, ESR = 4, DTA = 5, EZAG = 6, LSV = 7)	int	req.			-1
4	text	Default text	string	opt.	1	64	empty string
5	textIndex	Contain index of text within current joker type	int	req.			-1

2.6. ESRDetail

Settings for parsing ESR files

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	discountToleranceAmount	Amount used for automatic discount calculation	double	req.			0
3	esrDetailID	ESRDetail unique identifier	int	opt.			-1
4	invoiceNumIndex	Index in ESR file from where invoice num begin	int	req.			0
5	invoiceNumLength	Length of invoice number within ESR file	int	req.			0
6	partyNumIndex	Index in ESR file from where party num begin	int	req.			0
7	partyNumLength	Length of party number within ESR file	int	req.			0
8	useInvoiceFreeNumber	Whether the invocie free number used to search invoice	bit	req.			False
9	usePartyFreeNumber	Whether the party free number used to search invoice	bit	req.			False
10	useReferenceNumber	Whether the reference number used to search invoice	bit	req.			False
11	writeOffToleranceAmount	Amount used for automatic write off calculation	double	req.			0

2.7. VATPeriod

Periods of VAT rules

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	dateFrom	Date VAT period starts from	datetime	req.			current date
3	vatPeriodID	VATPeriod unique identifier	int	req.			-1

2.8. CostCenter

Cost center list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	code	CostCenter code	string	req.	1	11	empty string
3	comment	Commentary to CostCenter	string	opt.	0	128	empty string
4	costcenterID	CostCenter unique identifier	int	req.			-1
5	endDate	Project end date	datetime	opt.			current date
6	isInactive	Whether BankAccount is inactive	bit	opt.			False
7	isProject	Whether cost center is project	bit	opt.			False
8	modifyDate	Date of last instance modifying	datetime	opt.			current date
9	name	CostCenter name	string	opt.	1	64	empty string
10	startDate	Project start date	datetime	opt.			current date
11	userFID	Link to user which edit instance last time	int	opt.			-1

2.9. CostCenterMethod

Cost center methods per master cost cente in percents

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	costCenterChildFID	Link to child CostCenter	int	req.			-1
2	costCenterMasterFID	Link to master CostCenter	int	req.			-1
3	costCenterMethodID	CostCenter Method unique identifier	int	opt.			-1
4	percentage	Method percentage	double	req.			0

2.10. Country

Countries list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to Client	int	req.			-1
2	code	Country code	string	req.	1	3	empty string
3	countryID	Country unique identifier	int	req.			-1
4	name	Country name	string	opt.	1	30	empty string

2.11. Language

Language list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to Client	int	req.			-1
2	code	Language code	string	req.	1	3	empty string
3	languageID	Language unique identifier	int	req.			-1
4	name	Language name	string	req.	1	35	empty string

2.12. Currency

Currency list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	currencyID	Currency unique identifier	int	req.			-1
3	isoCode	Currency ISO code	string	req.	3	3	empty string
4	lossAccountFID	Link to loss difference account	int	opt.			-1
5	name	Currency name	string	opt.	1	50	empty string
6	profitAccountFID	Link to profit difference account	int	opt.			-1
7	rate	Currency rate to client currency	double	req.			1
8	symbol	Currency symbol	string	opt.	0	5	empty string
9	unit	Currency unit	double	opt.			1

2.13. Bank

Bank list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	address	Bank address	string	opt.	0	64	empty string
2	bankID	Bank unique identifier	int	req.			-1
3	bankPC	Bank postal code	string	opt.	0	22	empty string
4	city	Bank city	string	opt.	0	64	empty string
5	clientFID	Link to client	int	req.			-1
6	code	Bank code	string	req.	1	12	empty string
7	countryFID	Link to country	int	opt.			-1
8	name	Name of bank	string	opt.	1	64	empty string
9	postAddress	Bank post address	string	opt.	0	64	empty string
10	swift	SWIFT	string	opt.	0	12	empty string
11	zipCode	Bank zip code	string	opt.	0	22	empty string

2.14. Depreciation

Depreciation list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	depreciationID	Depreciation unique identifier	int	req.			-1
3	name	Depreciation name	string	req.	1	22	empty string
4	rate	Depreciation rate	double	req.			0

2.15. Partner

Partner list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to Client	int	req.			-1
2	description	Partner description	string	opt.	0	50	empty string
3	name	Partner name	string	opt.	1	22	empty string
4	participation	Partner participation in percent	double	req.			0
5	partnerID	Partner unique identifier	int	req.			-1

2.16. ReminderFee

Texts for reminder report according to reminder level

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	body	Reminder body	string	opt.	0	0	empty string
2	footer	Reminder footer	string	opt.	0	0	empty string
3	header	Reminder header	string	opt.	0	0	empty string
4	languageFID	Link to language message is written in	int	req.			-1
5	reminderFeeFID	Link to reminderFee	int	req.			-1
6	reminderMessageID	ReminderMessage unique identifier	uniqueidentifier	req.			empty guid
7	salutation	Reminder salutation	string	opt.	0	0	empty string

2.17. AccountArea

COA Account areas

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accAreaID	Account Area unique identifier	int	req.			-1
2	balanceOrNominal	Shows Account Area is balance (true = 1) or nominal (false = 0)	bit	req.			False
3	clientFID	Link to client	int	req.			-1
4	name	Name of Account Area	string	req.	1	30	empty string

2.18. Interest

Interests list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to Client	int	req.			-1
2	interestID	Interest unique identifier	int	req.			-1
3	name	Interest name	string	opt.	1	22	empty string

2.19. InterestRate

Interest details - from date, debit and credit interests

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	creditInterest	Credit Interest Rate	double	opt.			0
2	debitInterest	Debit Interest Rate	double	opt.			0
3	fromDate	InterestRate date from	datetime	req.			current date
4	interestFID	Link to Interest	int	req.			-1
5	interestRateID	InterestRate unique identifier	int	opt.			-1

2.20. PayTerm

Payment term in days

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	code	PayTerm code	string	req.	1	8	empty string
3	description	PayTerm description	string	opt.	0	40	empty string
4	dueDays	PayTerm Due Days, set due date of invoice by adding this value to invoice date	int	req.			0
5	payTermID	PayTerm unique identifier	int	req.			-1

2.21. PayTermDetail

Payment term detail - which percent discount is valid within days number

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	discount	PayTermDetail discount, percentage	double	opt.			0
2	payTermDetailID	PayTermDetail unique identifier	int	req.			-1
3	payTermFID	Link to payTerm	int	req.			-1
4	withinDays	Number of days within discount is valid	int	req.			0

2.22. BankAccount

Bank Account list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	accountNum	Bank account number	string	opt.	0	36	empty string
3	bankAccountID	Bank Account unique identifier	int	req.			-1
4	bankAccountTypeFID	Type of bankAccount (BankAccount = 1, PostalAccount = 2,GLAccount = 3)	int	req.			-1
5	bankFID	Link to bank	int	opt./req.			-1
6	clientFID	Link to client	int	req.			-1
7	creditLine	Credit line(limit) of bank account	double	opt.			0
8	iban	IBAN	string	opt.	0	22	empty string
9	isInactive	Whether BankAccount is inactive	bit	opt.			False
10	name	Name of bank account	string	req.	1	52	empty string

2.23. VAT

VAT list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	clientFID	Link to client	int	req.			-1
3	code	VAT code	string	req.	1	12	empty string
4	description	VAT description	string	opt.	0	80	empty string
5	isInclusive	Whether vat amount included = true or excluded = false to posting amount	bit	opt.			True
6	isInputTax	Whether vat is input tax (used in reports)	bit	opt.			False
7	vatID	VAT unique identifier	int	req.			-1

2.24. VATRate

VAT details - rate and quote

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	quote	VAT quote(percentage)	double	opt.			100
2	rate	VAT rate	double	req.			0
3	vatFID	Link to VAT	int	req.			-1
4	vatPeriodFID	Link to VAT period	int	req.			-1
5	vatRateID	VATRate unique identifier	int	req.			-1

2.25. VATGroupDetail

Contains VAT group used for reports

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	vatFID	Link to VAT	int	req.			-1
3	vatGroupDetailID	VATGroupDetail unique identifier	int	req.			-1
4	vatGroupFID	Type of VAT Group	int	req.			-1

2.26. COA

Chart of Accounts

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	clientFID	Link to client	int	req.			-1
2	coaID	COA unique identifier	int	req.			-1
3	creationDate	COA creation date	datetime	req.			current date
4	isMaster	Shows is this COA master(true = 1) or not (false = 0)	bit	req.			True
5	name	COA name	string	req.	1	50	empty string

2.27. COAView

Visual representation of chart of accounts

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	coaViewID	COA View unique identifier	int	req.			-1
2	coaFID	Link to COA	int	req.			-1
3	name	COA View name	string	req.	0	50	empty string

2.28. COATreeView

Chart of accounts tree (list of COA items)

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accAreaFID	Link to Account Area	int	req.			-1
2	accountFID	Link to Account	int	opt./req.			-1
3	coaFID	Link to COA	int	req.			-1
4	coaTreeViewID	COA TreeView unique identifier	int	req.			-1
5	code	Code of COA item	string	req.	2	22	empty string
6	compareFID	COA TreeView item with which current item will compare balance	int	opt.			-1
7	groupID	Link to tree group which contains this item	int	req.			-1
8	isCredit	Sign in report	bit	req.			False
9	isInvisible	Shows is COA item visible in reports (false = 0)	bit	opt.			False
10	linesBefore	Number of empty lines before this item in reports	int	opt.			0
11	modifyDate	Date of last instance modifying	datetime	opt.			current date
12	name	Name of COA item	string	req.	1	60	empty string
13	name2	Name2 of COA item	string	opt./req.	0	60	empty string
14	name2usage	Shows to use name2 instead name in reports(true = 1) or not (false = 0)	bit	opt.			False
15	noAccBalance	Reserved	bit	opt.			False
16	pageBefore	Shows is page break in reports before this COA item (true = 1)	bit	opt.			False
17	parentID	Link to previous tree item	int	req.			-1
18	remarks	Remarks for item	string	opt.	0	256	empty string
19	type	Type od COA item (Account = 1, AccountGroup = 2, AccountCopy = 3, AccountMirror = 4, Comment = 5, Counter = 6, Result = 7)	tinyint	req.			0
20	userFID	Link to user which edit instance last time	int	opt.			-1

2.29. CounterDetail

Detail information of counter per COA item

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	coaTreeViewFID	Link to COATreeView item which included to counter calculation	int	req.			-1
2	counterDetID	Counter Detail unique identifier	int	opt.			-1
3	counterFID	Link to counter(COATreeView item)	int	req.			-1
4	state	Sign with which item will be added to counter (true or 1 - plus, false or 0 - minus)	bit	req.			False

2.30. Account

GL Account list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountID	Account unique identifier	int	req.			-1
2	accTypeFID	Account type (Assets = 1, Liabiliies = 2, Revenue = 3 or Cost = 4)	int	req.			-1
3	coaTreeViewFID	Link to Master COA TreeView	int	req.			-1
4	costCenterFID	Link to Cost Center which will be used as default	int	opt.			-1
5	currencyFID	Link to Currency	int	req.			-1
6	inclInValuation	Include account in FC valuation	bit	opt.			True
7	isCcRequired	Shows whether cost center requiered for posting with this account(true = 1) or not (false = 0)	bit	opt.			False
8	isInactiv	Shows whether this account selectable for posting	bit	opt.			False
9	isMoneyAcc	Shows whether it's money account(true = 1) or not (false = 0)	bit	opt.			False
10	noAccDetails	Shows whether account details will be shown in account statement report(true = 1) or not (false = 0)	bit	opt.			False
11	noDirectPosting	Shows whether it's possible to make direct posting	bit	opt.			False
12	noZeroPrint	Shows whether zero balance will be printed	bit	opt.			False
13	startingBalance	Starting balance of the account (only for first fiscal year)	double	opt.			0
14	startingBalanceFC	Starting balance of the account in foreign currency	double	opt.			0
15	vatFID	Link to VAT which will be used as default	int	opt.			-1

2.31. Budget

Budget list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	budgetID	Budget unique identifier	int	req.			-1
2	comment	Commentary to budget	string	opt.	0	64	empty string
3	fiscalYearFID	Link to fiscal year	int	req.			-1
4	name	Name of budget	string	opt.	1	24	empty string

2.32. BudgetPeriod

Budget amount per account and period (month or quarter)

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	amount	Period amount	double	req.			0
3	budgetFID	Link to budget which contains this period	int	req.			-1
4	budgetPeriodID	Budget period unique identifier	int	opt.			-1
5	periodDate	Period end date (one month)	datetime	req.			current date

2.33. Party

Party list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	address1	Party address line 1	string	opt.	0	36	empty string
2	address2	Party address line 2	string	opt.	0	36	empty string
3	addressBlock	Address block	string	opt.	0	128	empty string
4	city	Party city	string	opt.	0	22	empty string
5	clientFID	Link to client	int	req.			-1
6	countryFID	Link to country	int	req.			-1
7	email	Party email	string	opt.	0	60	empty string
8	freePartyNum	Free number - defined by user	string	opt.	0	64	empty string
9	isPrivate	Whether is private person or not (f.ex. company)	bit	opt.			False
10	languageFID	Link to language	int	req.			-1
11	line1	Name line 1	string	opt.	0	60	empty string
12	line2	Name line 2	string	opt.	0	60	empty string
13	modifyDate	Date of last instance modifying	datetime	opt.			current date
14	name	Party name	string	req.	1	60	empty string
15	partyID	Party unique identifier	int	req.			-1
16	partyNum	Party number (handled by server) unique within client	int	req.			-1
17	remarks	Remarks to party	string	opt.	0	50	empty string
18	shortName	Party short name	string	req.	1	22	empty string
19	telefax	Party telefax	string	opt.	0	24	empty string
20	telephone	Party telephone	string	opt.	0	24	empty string
21	userFID	Link to user which edit instance last time	int	opt.			-1
22	web	Party web link	string	opt.	0	60	empty string
23	zip	Party zip	string	opt.	0	22	empty string

2.34. Person

Person info per party, debtor and creditor

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	email	Person email	string	opt.	0	60	empty string
2	firstName	Person first name	string	opt.	0	24	empty string
3	isMale	Shows is person male(true = 1) or female(false = 0)	bit	req.			False
4	jobTitle	Person jobTitle	string	opt.	0	25	empty string
5	lastName	Person last name	string	opt.	0	24	empty string
6	middleName	Person middle name	string	opt.	0	24	empty string
7	mobile	Person mobile number	string	opt.	0	24	empty string
8	partyFID	Link to party	int	req.			-1

9	personID	Person unique identifier	int	req.			-1
10	personRoleFID	Person Role (1 = Party, 2 = Debtor, 3 = Creditor)	int	req.			-1
11	salutation	Person Salutation (used in reporting e.g. Reminder)	string	opt.	0	128	empty string
12	telephone	Person telephone	string	opt.	0	24	empty string
13	title	Person title	string	opt.	0	24	empty string
14	titleSuffix	Person titleSuffix(e.g. 'Mr.', 'Herr,' etc)	string	opt.	0	24	empty string

2.35. PayMethod

Payment methods of party

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	accountNum	PayMethod accountNum	string	opt.	0	36	empty string
3	bankAccountFID	Link to Bank Account	int	req.			-1
4	bankFID	Link to bank	int	opt./req.			-1
5	name	PayMethod name	string	opt.	1	52	empty string
6	partyFID	Link to party	int	req.			-1
7	payMethodID	PayMethod unique identifier	int	req.			-1
8	payslipCode	PayMethod Payslip Code	string	opt.	0	36	empty string
9	payTypeFID	Link to payType	int	req.			-1
10	personRole	Shows to which type of party paymenthod linked (1 = Debtor, 2 = Creditor)	int	req.			-1

2.36. Debtor

Party debtor info

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	opt.			-1
2	creditLine	Debtor Credit Line(limit)	double	opt.			0
3	debtorID	Debtor unique identifier	int	opt.			-1
4	freeCode	Debtor Free Code user defined	string	opt.	0	13	empty string
5	isInactiv	Whether Debtor will appear in invoice window (if true then will not appear)	bit	opt.			False
6	modifyDate	Date of last instance modifying	datetime	opt.			current date
7	noReminders	Default flag which is set for invoice	bit	opt.			False
8	noRemindersFee	Reserved	bit	opt.			False
9	partyFID	Link to party	int	req.			-1
10	payMethodFID	Link to default debtor payment method	int	req.			-1
11	payTermFID	Link to Payment Term	int	req.			-1
12	userFID	Link to user which edit instance last time	int	opt.			-1

2.37. Creditor

Party creditor info

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	opt.			-1
2	creditLine	Creditor credit line (limit)	double	opt.			0
3	creditorID	Creditor unique identifier	int	req.			-1
4	freeCode	Creditor freeCode	string	opt.	0	13	empty string
5	isInactiv	Whether Creditor will appear in invoice window (if true then will not appear)	bit	opt.			False
6	modifyDate	Date of last instance modifying	datetime	opt.			current date
7	ourCustomerNum	Creditor ourCustomerNum	string	opt.	0	20	empty string
8	partyFID	Link to party	int	req.			-1
9	payMethodFID	Link to default payment method of Creditor which proposed in invoice	int	req.			-1
10	payTermFID	Link to payTerm	int	req.			-1
11	userFID	Link to user which edit instance last time	int	opt.			-1
12	vatNumber	Creaditor VAT number	string	opt.	0	18	empty string

2.38. Transaction

Transaction list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	docDate	Date of transaction	datetime	req.			current date
2	docNum	Transaction document number	int	req.			-1
3	docType	Transaction Document Type	string	opt.	0	2	empty string
4	fiscalYearFID	Link to fiscal year	int	req.			-1
5	isConfirmed	Shows is transaction confirmed (true = 1) Used for Payment Run	bit	opt.			True
6	modifyDate	Date of last transaction modifying	datetime	opt.			current date
7	transactionID	Transaction unique identifier	uniqueidentifier	req.			empty guid
8	transactionTypeFID	Type of transaction(Simple = 1, Compound = 2, Invoice = 3, Payment = 4)	int	req.			0
9	userFID	Link to user which edit transaction last time	int	opt.			-1

2.39. Posting

Transaction postings list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	amount	Amount of posting	double	req.			0
3	contraAccountFID	Link to contra account if = -1 then 'Div.'	int	opt./req.			-1
4	costCenterFID	Link to cost center	int	opt./req.			-1
5	exchangeRate	Exchange rate of posting (If posting in foreign currency)	double	opt./req.			1
6	fcAmount	Amount of posting in foreign currency	double	req.			0
7	freeCode	Free code of posting	string	opt.	0	13	empty string
8	invoiceFID	Link to invoice(for invoice and payment transaction)	uniqueidentifier	opt./req.			empty guid
9	isDebit	Shows is posting debit (true = 1) or credit (false = 0)	bit	req.			False
10	isInclusive	Is VAT amount included or not	bit	opt.			False
11	isInvisible	Whether posting invisible used for recurrent transaction	bit	opt.			False
12	postingID	Posting unique identifier	uniqueidentifier	opt.			empty guid
13	postingIndex	Index of posting within transaction	int	opt./req.			0
14	postingTypeFID	Type of posting(Compound = 1, Invoice = 2, Payment = 3, WriteOff = 4, Discount = 5, Fee = 6, Difference = 7)	int	req.			-1
15	text	Posting text	string	opt.	0	64	empty string
16	transactionFID	Link to transaction	uniqueidentifier	req.			empty guid
17	vatAccountFID	Link to VAT account	int	opt./req.			-1
18	vatAmount	VAT amount of posting	double	opt./req.			0
19	vatFcAmount	VAT amount of posting in foreign currency	double	opt./req.			0

20	vatFID	Link to vat	int	opt./req.			-1
21	vatPercent	VAT percent	double	opt./req.			0
22	vatQuote	VAT Quote	double	opt./req.			100

2.40. Invoice

Invoice list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	authorizerFID	Link to user authorizer	int	opt.			-1
2	barCode	Invoice barcode	string	opt.	0	64	empty string
3	clientFID	Link to Client	int	req.			-1
4	dueDate	Date due when invoice has to be paid	datetime	req.			01.01.1753 0:00:00
5	exchangeRate	Invoice exchange rate, if invoice in foreign currency else = 1	double	opt./req.			1
6	freeInvoiceNum	Free Invoice number defined by user	string	opt.	0	64	empty string
7	invoiceDate	Date of invoice creation	datetime	req.			current date
8	invoiceID	Invoice unique identifier (same as transaction if exists)	uniqueidentifier	req.			empty guid
9	invoiceNum	Invoice number (handled by server), unique within client and invoice type(person role)	int	req.			-1
10	isAuthorized	Shows is invoice authorized(true = 1)	bit	opt.			True
11	modifyDate	Date when invoice was modified last time	datetime	opt.			current date
12	noReminders	If set to true, invoice will not be proposed to reminders	bit	opt.			False
13	partyFID	Link to Party	int	req.			-1
14	payMethodFID	Link to payMethod	int	req.			-1
15	payslipCode	Invoice Payslip Code	string	opt.	0	64	empty string
16	payTermFID	Link to payTerm	int	req.			-1
17	personRoleFID	Show type of invoice (2 = Debtor (Sales), 3 = Creditor (Purchase)	int	req.			-1
18	reminderLevel	Reminder level with which invoice created (start level)	int	opt.			0
19	text	Invoice text	string	opt.	0	64	empty string
20	totalAmount	Invoice amount	double	req.			0
21	totalAmountFC	Invoice amount in foreign currency	double	req.			0
22	userFID	Link to user which edit invoice last time	int	opt.			-1
23	vatFID	Link to VAT	int	opt.			-1

2.41. Payment

Payment list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	accountFID	Link to account	int	req.			-1
2	amount	Payment amount	double	req.			0
3	fcAmount	Payment amount in foreign currency	double	req.			0
4	partyFID	Link to party, in case of payment run = -1	int	opt./req.			-1
5	paymentID	Payment unique identifier (same as transaction)	uniqueidentifier	req.			empty guid
6	paymentRate	Payment exchange rate	double	opt./req.			1
7	paymentTransactionTypeFID	Type of payment (1 = Manual, 2 = ESR, 2 = Payment Run	int	req.			-1
8	personRoleFID	Show type of payment (1 = Debtor (Sales), 2 = Creditor (Purchase))	int	req.			-1
9	text	Payment text	string	opt.	0	64	empty string

2.42. PaymentRun

Payment run list

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	bankAccountFID	Link to internal Bank Account	int	req.			-1
2	isPrinted	Shows was Payment Run printed(true = 1)	bit	opt.			False
3	isSent	Shows whether Payment Run was sent (true = 1	bit	opt.			False
4	paymentRunID	PaymentRun unique identifier(same as payment and transaction)	uniqueidentifier	req.			empty guid
5	valueDate	Date, on which Payment Run was proposed	datetime	req.			current date

2.43. Reminder

Reminder per party about not payd invoices

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	consideredDate	Payment considered until	datetime	req.			current date
2	currencyFID	Link to currency	int	req.			-1
3	isPrinted	Shows was reminder printed (true = 1)	bit	opt.			False
4	modifyDate	Date of last reminder modifying	datetime	opt.			current date
5	partyFID	Link to party	int	req.			-1
6	reminderDate	Date of reminder	datetime	req.			current date
7	reminderID	Reminder unique identifier	uniqueidentifier	req.			empty guid
8	reminderLevel	Reminder level	int	req.			0
9	userFID	Link to user which edit reminder last time	int	opt.			-1
10	valueDate	Date due when reminder was proposed	datetime	req.			current date

2.44. ReminderDetail

Detail of reminder per invoive

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	invoiceFID	Link to invoice	uniqueidentifier	req.			empty guid
2	reminderDetailID	ReminderDetail unique identifier	uniqueidentifier	req.			empty guid
3	reminderFID	Link to reminder	uniqueidentifier	req.			empty guid

2.45. ReminderMessage

Texts for reminder report according to reminder level

Pos	Field Name	Description	Type	Req.	Min	Max	Default
1	reminderMessageID	ReminderMessage unique identifier	uniqueidentifier	req.			empty guid
2	languageFID	Link to language message is written in	int	req.			-1
3	reminderFeeFID	Link to reminderFee	int	req.			-1
4	header	Reminder header	string	opt.	0	0	empty string
5	body	Reminder body	string	opt.	0	0	empty string
6	footer	Reminder footer	string	opt.	0	0	empty string
7	salutation	Reminder salutation	string	opt.	0	0	empty string
8	isLeftPosition	Position of text in reminder reports	bit	opt.			True

3. XML Structure

3.1. General XML Description

The **Extensible Markup Language** (**XML**) is a W3C-recommended general-purpose markup language that supports a wide variety of applications. XML languages or 'dialects' are easy to design and to process. They are also reasonably human legible, and to this end, terseness was not considered essential in its structure. XML is a simplified subset of Standard Generalized Markup Language (SGML). Its primary purpose is to facilitate the sharing of data across different information systems, particularly systems connected via the Internet

Topal System uses the XML format as intermediate source for Data Migration.

For example, when you export data from Topal, Topal System creates the XML representation of exported data. Then this XML can be used as source for data importing.

In the same way solved import migration process from other systems (e.g. SAGE Sesam). XML is an intermediate source between parsing from Sesam and importing into Topal.

The XML is also used for Mandalink interface.

You can read more about XML by using link - http://www.w3.org/XML/.

3.2. XML Schemes

XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML document in more detail.

It is not necessary to define schema in for import XML, but it’s recommended. The schemas defining help to prevent error in XML document.

Anyway, Topal System will “understand” XML even without schema but of course all other description should be correct.

You can read more about XML schemas by using link - http://www.w3.org/XML/Schema

3.3. Topal Entities XML Representation

If you use .NET then you have comfortable classes for creation or parsing Topal Import XML file. With ADO.NET objects, you have very powerful tools.

There are tools existing in most languages, which make your work with XML much simpler then do it manually.

You will find XML Schema description below**.**

Therefore, we can start to create Import XML Source.

It is recommended to add XML Description tag:

<?xml version="1.0" standalone="yes"?>.

Then you have to define XML root element, but be aware that only one root element can be defined within XML document. For example:

```
<NewDataSet>
</NewDataSet>
```

You see that tag has to be closed (</>).

Within root element, you can add Topal instances in XML representation.

For example the following XML describe client.

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Client>
    <clientID>2</clientID>
    <code>1          </code>
    <name>My First Client                    </name>
    <street>Morgenstrasse 129                  </street>
    <zip>3018                              </zip>
    <city>Bern                  </city>
    <telephone>031 355 55 55        </telephone>
    <person xml:space="preserve">                    </person>
    <remarks xml:space="preserve">                              </remarks>
    <vatNumber>123 456          </vatNumber>
    <vatRequired>true</vatRequired>
    <vatSimplified>false</vatSimplefied>
    <vatAgreed>true</vatAgreed>
    <vatPaymentBased>false</vatPaymentBased>
    <isTemplate>false</isTemplate>
    <backupTS>2006-05-15T14:03:15.283+03:00</backupTS>
    <lastTimestamp>2006-05-15T14:03:15.283+03:00</lastTimestamp>
    <isKCSwissRounded>true</isKCSwissRounded>
    <isVATSwissRounded>true</isVATSwissRounded>
    <isDiscountSwissRounded>true</isDiscountSwissRounded>
  </Client>
</NewDataSet>
```

When importing this XML you will create new client. This client will be empty – without Master Data, Fiscal Years, VAT and COA. It happens because we define only client instance.

As mentioned above it is not necessary to define all the fields, some fields are optional, the following XML will be correct too:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Client>
    <clientID>2</clientID>
    <code>2        </code>
    <name>My Second Client</name>
  </Client>
</NewDataSet>
```

As you see there, we define only few attributes, all other client properties will be set as default.

4. Using Topal Data Migration

The current chapter contains a help for using Topal Data Migration System.

4.1. Export

The best way to see correct Topal XML is using Topal Export. You can create Export client, save it in local file, and then look how Topal create XML.

First, you have to run Topal Client. Now you can set current client.

Only current client will be exported to XML.

Open Export window from main menu “File → Export”

There are several kinds of export available in the Export window. The most important is the “Full Client” and “Fiscal Year”.

The full client export includes current client to the XML and all fiscal years of current client.

The fiscal year option means exporting of current client but only current fiscal year will be exported. All data, which is fiscal year dependent (e.g. transactions, postings, invoice, payment etc.) will be exported only with current fiscal year. For example, only transactions of current fiscal year will be exported.

In addition, the “Custom” option is available. By using this option, you can select tables from XML, which you want to export.

4.2. Import

When you create XML source for Topal you can import it in the Topal System.

Open window from main menu “File → Import”.

The similar options as in Export are available here. You can see “Client”, “Fiscal Year” and “Custom” import there.

As exporting the “Client” import always, means creation of new client. Therefore, you have to define client instance in you XML source.

If you select “Fiscal Year” import, the Topal System will not create new client. Even if you include client instance to your XML the Topal System will use this instance for updating current client, but will not create new one.

By using “Custom” import, you can select items from XML source, which you want to import into Topal. It is the most flexible kind of importing which needs more or less deep understanding of Topal Migration process.

5. Example of Topal XML Source

In current chapter, you will find the simple example of XML creation process.

Let’s try to create source, which define client with fiscal year and some master data.

As mentioned in the chapter 3.3 we create general description and root element:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>


</NewDataSet>
```

We will create client with one fiscal year, few country (with client default country), currency (with client currency) and language (with client language).

We will not include all tables and all fields in the tables.

First, we can define schema. Our XML will look like:

```
      <?xml version="1.0" standalone="yes"?>
      <NewDataSet>
        <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
          <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
            <xs:complexType>
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="Client">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="clientID" type="xs:int" minOccurs="0" />
                      <xs:element name="code" type="xs:string" minOccurs="0" />
                      <xs:element name="name" type="xs:string" minOccurs="0" />
                      <xs:element name="street" type="xs:string" minOccurs="0" />
                      <xs:element name="zip" type="xs:string" minOccurs="0" />
                      <xs:element name="city" type="xs:string" minOccurs="0" />
                      <xs:element name="telephone" type="xs:string" minOccurs="0" />
                      <xs:element name="person" type="xs:string" minOccurs="0" />
                      <xs:element name="remarks" type="xs:string" minOccurs="0" />
                      <xs:element name="vatNumber" type="xs:string" minOccurs="0" />
                      <xs:element name="vatRequired" type="xs:boolean" minOccurs="0" />
                      <xs:element name="vatSimplefied" type="xs:boolean" minOccurs="0" />
                      <xs:element name="vatAgreed" type="xs:boolean" minOccurs="0" />
                      <xs:element name="vatPaymentBased" type="xs:boolean" minOccurs="0" />
                      <xs:element name="isTemplate" type="xs:boolean" minOccurs="0" />
                      <xs:element name="backupTS" type="xs:dateTime" minOccurs="0" />
                      <xs:element name="lastTimestamp" type="xs:dateTime" minOccurs="0" />
                      <xs:element name="isKCSwissRounded" type="xs:boolean" minOccurs="0" />
                      <xs:element name="isVATSwissRounded" type="xs:boolean" minOccurs="0" />
                      <xs:element name="isDiscountSwissRounded" type="xs:boolean" minOccurs="0" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
```

```
<xs:element name="IsDefault">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="defaultID" type="xs:int" minOccurs="0" />
      <xs:element name="clientFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCountryFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCurrencyFID" type="xs:int" minOccurs="0" />
      <xs:element name="defLanguageFID" type="xs:int" minOccurs="0" />
      <xs:element name="defBankAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defDebitPaymentTypeFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCreditPaymentTypeFID" type="xs:int" minOccurs="0" />
      <xs:element name="defPaymentTermFID" type="xs:int" minOccurs="0" />
      <xs:element name="defDebtorDiscountAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCreditorDiscountAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defDebtorWriteOffAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCreditorWriteOffAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defFeeAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defDebtorAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defCreditorAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defMoneyAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="defBudgetPeriodFID" type="xs:int" minOccurs="0" />
      <xs:element name="defAutoDiscountAmount" type="xs:decimal" minOccurs="0" />
      <xs:element name="defNoReminderAmount" type="xs:decimal" minOccurs="0" />
      <xs:element name="currPartyNum" type="xs:int" minOccurs="0" />
      <xs:element name="currPurchaseInvoiceNum" type="xs:int" minOccurs="0" />
      <xs:element name="currSalesInvoiceNum" type="xs:int" minOccurs="0" />
      <xs:element name="currReminderNum" type="xs:int" minOccurs="0" />
      <xs:element name="clientConfig" type="xs:base64Binary" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="FiscalYear">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="fiscalYearID" type="xs:int" minOccurs="0" />
      <xs:element name="clientFID" type="xs:int" minOccurs="0" />
      <xs:element name="defDocumentNumberFID" type="xs:int" minOccurs="0" />
      <xs:element name="defBudgetFID" type="xs:int" minOccurs="0" />
      <xs:element name="resultAccountFID" type="xs:int" minOccurs="0" />
      <xs:element name="name" type="xs:string" minOccurs="0" />
      <xs:element name="dateFrom" type="xs:dateTime" minOccurs="0" />
      <xs:element name="dateUntil" type="xs:dateTime" minOccurs="0" />
      <xs:element name="docdateFrom" type="xs:dateTime" minOccurs="0" />
      <xs:element name="docdateUntil" type="xs:dateTime" minOccurs="0" />
      <xs:element name="closedFlag" type="xs:boolean" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Country">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="countryID" type="xs:int" minOccurs="0" />
      <xs:element name="clientFID" type="xs:int" minOccurs="0" />
      <xs:element name="code" type="xs:string" minOccurs="0" />
      <xs:element name="name" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Language">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="languageID" type="xs:int" minOccurs="0" />
```

```
                <xs:element name="clientFID" type="xs:int" minOccurs="0" />
                <xs:element name="code" type="xs:string" minOccurs="0" />
                <xs:element name="name" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Currency">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="currencyID" type="xs:int" minOccurs="0" />
                <xs:element name="clientFID" type="xs:int" minOccurs="0" />
                <xs:element name="accountFID" type="xs:int" minOccurs="0" />
                <xs:element name="isoCode" type="xs:string" minOccurs="0" />
                <xs:element name="symbol" type="xs:string" minOccurs="0" />
                <xs:element name="name" type="xs:string" minOccurs="0" />
                <xs:element name="rate" type="xs:decimal" minOccurs="0" />
                <xs:element name="unit" type="xs:decimal" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</NewDataSet>
```

Now we can insert client item:

```
<Client>
 <clientID>1</clientID>
 <code xml:space="preserve">          </code>
 <name>My Client            </name>
 <vatRequired>true</vatRequired>
 <vatSimplefied>false</vatSimplefied>
 <isTemplate>true</isTemplate>
 <isKCSwissRounded>true</isKCSwissRounded>
 <isVATSwissRounded>true</isVATSwissRounded>
 <isDiscountSwissRounded>true</isDiscountSwissRounded>
</Client>
```

Now you see here that we define clientID, the IDs within XML it is not a real ID from DB. The main requirement is – ID of different instances of the same entity should be original.

Let’s insert the fiscal year:

```
<FiscalYear>
 <fiscalYearID>1</fiscalYearID>
 <clientFID>1</clientFID>
 <defDocumentNumberFID>1</defDocumentNumberFID>
 <defBudgetFID>-1</defBudgetFID>
 <resultAccountFID>-1</resultAccountFID>
 <name>2006                </name>
```

```
  <dateFrom>2006-01-01T00:00:00+02:00</dateFrom>
  <dateUntil>2006-12-31T00:00:00+02:00</dateUntil>
  <docdateFrom>2006-01-01T00:00:00+02:00</docdateFrom>
  <docdateUntil>2006-12-31T00:00:00+02:00</docdateUntil>
  <closedFlag>false</closedFlag>
</FiscalYear>
```

With next step, we can insert master data instances:

```
<Country>
  <countryID>1</countryID>
  <clientFID>1</clientFID>
  <code>CH </code>
  <name>CH                           </name>
</Country>
<Country>
  <countryID>2</countryID>
  <clientFID>1</clientFID>
  <code>US </code>
  <name>US                           </name>
</Country>
<Language>
  <languageID>1</languageID>
  <clientFID>1</clientFID>
  <code>DE </code>
  <name>Deutsch                      </name>
</Language>
<Language>
  <languageID>2</languageID>
  <clientFID>1</clientFID>
  <code>EN </code>
  <name>English                      </name>
</Language>
<Language>
  <languageID>3</languageID>
  <clientFID>1</clientFID>
  <code>FR </code>
  <name>Fran├зais                     </name>
</Language>
<Language>
  <languageID>4</languageID>
  <clientFID>1</clientFID>
  <code>IT </code>
```

```
   <name>Italiano                  </name>
  </Language>
  <Currency>
   <currencyID>30</currencyID>
   <clientFID>1</clientFID>
   <accountFID>-1</accountFID>
   <isoCode>CHF</isoCode>
   <symbol>CHF  </symbol>
   <name>CHF                              </name>
   <rate>1.0000</rate>
   <unit>1.0000</unit>
  </Currency>
  <Currency>
   <currencyID>160</currencyID>
   <clientFID>1</clientFID>
   <accountFID>-1</accountFID>
   <isoCode>USD</isoCode>
   <symbol>USD  </symbol>
   <name>USD                              </name>
   <rate>1.2500</rate>
   <unit>1.0000</unit>
  </Currency>
```

As you can see, here we add several instances of same entities – two currencies, two countries and several languages. Look to the country with code “CH” and language with code “DE”, the IDs of these instances are the same but it doesn’t matter because the entities are different.

Now we can insert “IsDefault” instance, which contains default values of the client. We do not need all defaults:

```
  <IsDefault>
   <defaultID>1</defaultID>
   <clientFID>1</clientFID>
   <defCountryFID>1</defCountryFID>
   <defCurrencyFID>30</defCurrencyFID>
   <defLanguageFID>1</defLanguageFID>
  </IsDefault>
```

As you see, we set default country with ID 1, default currency with ID 30 and default language with ID 1.

It means the client, which we will import, would have currency “CHF”, live in the Switzerland (country code “CH”) and his native language is German (language code “DE”).

Now you can import this XML with “Client” kind of import and check it.

The final XML file will looks like:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
 <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
   <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
     <xs:element name="Client">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="clientID" type="xs:int" minOccurs="0" />
        <xs:element name="code" type="xs:string" minOccurs="0" />
        <xs:element name="name" type="xs:string" minOccurs="0" />
        <xs:element name="street" type="xs:string" minOccurs="0" />
        <xs:element name="zip" type="xs:string" minOccurs="0" />
        <xs:element name="city" type="xs:string" minOccurs="0" />
        <xs:element name="telephone" type="xs:string" minOccurs="0" />
        <xs:element name="person" type="xs:string" minOccurs="0" />
        <xs:element name="remarks" type="xs:string" minOccurs="0" />
        <xs:element name="vatNumber" type="xs:string" minOccurs="0" />
        <xs:element name="vatRequired" type="xs:boolean" minOccurs="0" />
        <xs:element name="vatSimplefied" type="xs:boolean" minOccurs="0" />
        <xs:element name="vatAgreed" type="xs:boolean" minOccurs="0" />
        <xs:element name="vatPaymentBased" type="xs:boolean" minOccurs="0" />
        <xs:element name="isTemplate" type="xs:boolean" minOccurs="0" />
        <xs:element name="backupTS" type="xs:dateTime" minOccurs="0" />
        <xs:element name="lastTimestamp" type="xs:dateTime" minOccurs="0" />
        <xs:element name="isKCSwissRounded" type="xs:boolean" minOccurs="0" />
        <xs:element name="isVATSwissRounded" type="xs:boolean" minOccurs="0" />
        <xs:element name="isDiscountSwissRounded" type="xs:boolean" minOccurs="0" />
       </xs:sequence>
      </xs:complexType>
     </xs:element>
     <xs:element name="IsDefault">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="defaultID" type="xs:int" minOccurs="0" />
        <xs:element name="clientFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCountryFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCurrencyFID" type="xs:int" minOccurs="0" />
        <xs:element name="defLanguageFID" type="xs:int" minOccurs="0" />
        <xs:element name="defBankAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defDebitPaymentTypeFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCreditPaymentTypeFID" type="xs:int" minOccurs="0" />
        <xs:element name="defPaymentTermFID" type="xs:int" minOccurs="0" />
        <xs:element name="defDebtorDiscountAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCreditorDiscountAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defDebtorWriteOffAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCreditorWriteOffAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defFeeAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defDebtorAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defCreditorAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defMoneyAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="defBudgetPeriodFID" type="xs:int" minOccurs="0" />
        <xs:element name="defAutoDiscountAmount" type="xs:decimal" minOccurs="0" />
        <xs:element name="defNoReminderAmount" type="xs:decimal" minOccurs="0" />
        <xs:element name="currPartyNum" type="xs:int" minOccurs="0" />
        <xs:element name="currPurchaseInvoiceNum" type="xs:int" minOccurs="0" />
        <xs:element name="currSalesInvoiceNum" type="xs:int" minOccurs="0" />
        <xs:element name="currReminderNum" type="xs:int" minOccurs="0" />
        <xs:element name="clientConfig" type="xs:base64Binary" minOccurs="0" />
       </xs:sequence>
      </xs:complexType>
     </xs:element>
     <xs:element name="FiscalYear">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="fiscalYearID" type="xs:int" minOccurs="0" />
        <xs:element name="clientFID" type="xs:int" minOccurs="0" />
        <xs:element name="defDocumentNumberFID" type="xs:int" minOccurs="0" />
        <xs:element name="defBudgetFID" type="xs:int" minOccurs="0" />
        <xs:element name="resultAccountFID" type="xs:int" minOccurs="0" />
        <xs:element name="name" type="xs:string" minOccurs="0" />
        <xs:element name="dateFrom" type="xs:dateTime" minOccurs="0" />
        <xs:element name="dateUntil" type="xs:dateTime" minOccurs="0" />
```

```
        <xs:element name="docdateFrom" type="xs:dateTime" minOccurs="0" />
        <xs:element name="docdateUntil" type="xs:dateTime" minOccurs="0" />
        <xs:element name="closedFlag" type="xs:boolean" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="countryID" type="xs:int" minOccurs="0" />
        <xs:element name="clientFID" type="xs:int" minOccurs="0" />
        <xs:element name="code" type="xs:string" minOccurs="0" />
        <xs:element name="name" type="xs:string" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Language">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="languageID" type="xs:int" minOccurs="0" />
        <xs:element name="clientFID" type="xs:int" minOccurs="0" />
        <xs:element name="code" type="xs:string" minOccurs="0" />
        <xs:element name="name" type="xs:string" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Currency">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="currencyID" type="xs:int" minOccurs="0" />
        <xs:element name="clientFID" type="xs:int" minOccurs="0" />
        <xs:element name="accountFID" type="xs:int" minOccurs="0" />
        <xs:element name="isoCode" type="xs:string" minOccurs="0" />
        <xs:element name="symbol" type="xs:string" minOccurs="0" />
        <xs:element name="name" type="xs:string" minOccurs="0" />
        <xs:element name="rate" type="xs:decimal" minOccurs="0" />
        <xs:element name="unit" type="xs:decimal" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:choice>
```

```
    </xs:complexType>
  </xs:element>
</xs:schema>
<Client>
  <clientID>1</clientID>
  <code xml:space="preserve">          </code>
  <name>My Client            </name>
  <vatRequired>true</vatRequired>
  <vatSimplefied>false</vatSimplefied>
  <isTemplate>true</isTemplate>
  <isKCSwissRounded>true</isKCSwissRounded>
  <isVATSwissRounded>true</isVATSwissRounded>
  <isDiscountSwissRounded>true</isDiscountSwissRounded>
</Client>
<IsDefault>
  <defaultID>1</defaultID>
  <clientFID>1</clientFID>
  <defCountryFID>1</defCountryFID>
  <defCurrencyFID>30</defCurrencyFID>
  <defLanguageFID>1</defLanguageFID>
</IsDefault>
<FiscalYear>
  <fiscalYearID>1</fiscalYearID>
  <clientFID>1</clientFID>
  <defDocumentNumberFID>1</defDocumentNumberFID>
  <defBudgetFID>-1</defBudgetFID>
  <resultAccountFID>-1</resultAccountFID>
  <name>2006                    </name>
  <dateFrom>2006-01-01T00:00:00+02:00</dateFrom>
  <dateUntil>2006-12-31T00:00:00+02:00</dateUntil>
  <docdateFrom>2006-01-01T00:00:00+02:00</docdateFrom>
  <docdateUntil>2006-12-31T00:00:00+02:00</docdateUntil>
  <closedFlag>false</closedFlag>
</FiscalYear>
<Country>
  <countryID>1</countryID>
  <clientFID>1</clientFID>
  <code>CH </code>
  <name>CH                          </name>
</Country>
<Country>
```

```
  <countryID>2</countryID>
  <clientFID>1</clientFID>
  <code>US </code>
  <name>US                            </name>
 </Country>
 <Language>
  <languageID>1</languageID>
  <clientFID>1</clientFID>
  <code>DE </code>
  <name>Deutsch                       </name>
 </Language>
 <Language>
  <languageID>2</languageID>
  <clientFID>1</clientFID>
  <code>EN </code>
  <name>English                       </name>
 </Language>
 <Language>
  <languageID>3</languageID>
  <clientFID>1</clientFID>
  <code>FR </code>
  <name>Fran├зais                      </name>
 </Language>
 <Language>
  <languageID>4</languageID>
  <clientFID>1</clientFID>
  <code>IT </code>
  <name>Italiano                      </name>
 </Language>
 <Currency>
  <currencyID>30</currencyID>
  <clientFID>1</clientFID>
  <accountFID>-1</accountFID>
  <isoCode>CHF</isoCode>
  <symbol>CHF  </symbol>
  <name>CHF                                         </name>
  <rate>1.0000</rate>
  <unit>1.0000</unit>
 </Currency>
 <Currency>
  <currencyID>160</currencyID>
```

```
    <clientFID>1</clientFID>
    <accountFID>-1</accountFID>
    <isoCode>USD</isoCode>
    <symbol>USD  </symbol>
    <name>USD                                        </name>
    <rate>1.2500</rate>
    <unit>1.0000</unit>
  </Currency>
</NewDataSet>
```

6. Topal XML Validation

Topal Data Migration System support several import sources. It could be for example XML, CVS, and TAFF.

There are no common rules for data organization for all software. Topal Data Structure created with ID approach, it is useful because you can be sure that any entity has its own unique identify (ID) at least within table and will not be mixed with other. As mentioned, Topal System creates special field in the entity record to use it as ID, but it is possible to link several entities within XML by using special *validation* fields.

Using ID is not always user friendly and obvious way to link instances within XML document.

It is possible to make instance identification from one or several field to use alternate way for instance identifying by using validation fields.

6.1. ID approach problem

Even already existing client sometimes needs an additional data importing – updating of Master Data or COA elements, migration of new transaction, etc. Such kind of migration called as *partly importing*.

Topal Data Structure it a most right way to make correct import source because that's exactly an immediate representation of data between external source and internal DB. XML document created in that way allows defining all importable Topal entities and most appropriate to import whole client. But in case of *partly* importing this approach could be less suitable.

Each entity imported correctly should have completely filled fields and links to other entities (FID). If you describe in XML document an instance which is already exists in DB it could be corrupted if some fields have wrong values.

When you want to import one transaction you have to describe posting for it. The posting has a link to account(s), and since IDs in XML are not same as DB IDs you have to define account in XML source before giving their FID to posting instance. But account has also FID to other instances like COATreeView and since COATreeView has a treelike structure, to describe at least one COA item the neighborhood items which has its own neighborhoods should be described, so all tree items will be described. Its only simple example but in the real life one transaction needs most of Master Data describing in the XML source.

For solving this problem the *validation* approach was created.

Instead of FID you can use special validation fields which define a link to other entities. Hence, validation it's an alternate way to define relations.

When you create transaction and describe the posting you have to define link to the posting's account. For that you can use "accountCode" field instead of "accountFID", and "vatCode" instead of "vatFID". And if there is an account with this code exists in the Topal database you don't have to describe account in the XML. The import with validation accept data source created in such way.

6.2. Validation process

The main function of validation process it's a recognizing FIDs by the special validation fields. Most of entities have special validation fields which will help to find according instance and link it.

For example we have posting record with field accountCode value "1000". First, the validation module looks to the import XML to find account record with code "1000" there. If it is not present there then such account will be searched in the database. If there is no such account in the DB, then new account record will be created based on validation values and other fields will be default. If account will be found then it will linked to the posting.

So, the first step of instance validation (recognizing FIDs by validation fields) it's a searching of appropriate linked instances within source XML. But as mentioned in case of partly importing it's not always possible. In this case linked instance searched in the DB. In case of partly importing XML source should be imported to the current client, it means only instance of current client could be found in DB during validation.

In our example we have posting with "accountCode" equal "1000". This field's value could help to find accountFID. There is no such **Account** instance in the XML. If current client (in which XML source imported to) do not have such account, the new account instance will be created. All the properties of this instance will be default, except “code” which will be equal “1000”. But this posting will not be imported, because account is not valid. There is no information about account area and account type into which new account should be placed.

If you make XML with Bank instance, where you define all necessary information, with “countryCode” attribute equal “CH” but there is no such country in the database, country with code “CH” will be created and imported, because it is enough to know only code of country that it will be importable. After this bank record will be imported too.

6.3. Example of import with validation

Let’s try to import one transaction.

In the source XML we will define not IDs but validation fields.

Please create new client based on the template client, we will use it for testing. The next step creates following XML:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>

  <Transaction>
    <transactionTypeFID>2</transactionTypeFID>
    <docNum>2</docNum>
    <docDate>2006-10-24</docDate>
  </Transaction>

  <Posting>
    <docNum>2</docNum>
    <accountCode>1000</accountCode>
    <vatCode>UStn</vatCode>
    <vatAccountCode>2000</vatAccountCode>
    <amount>743.5000</amount>
    <fcAmount>743.4900</fcAmount>
    <exchangeRate>1.0000</exchangeRate>
    <vatAmount>56.5000</vatAmount>
    <vatFcAmount>56.5000</vatFcAmount>
    <vatPercent>7.6000</vatPercent>
    <vatQuote>100.0000</vatQuote>
    <text>Posting 1</text>
    <isDebit>false</isDebit>
    <isInclusive>true</isInclusive>
    <postingIndex>1</postingIndex>
  </Posting>

  <Posting>
    <docNum>2</docNum>
    <accountCode>3200</accountCode>
    <amount>800.0000</amount>
    <fcAmount>800.0000</fcAmount>
    <exchangeRate>1.0000</exchangeRate>
    <text>Posting 2</text>
    <isDebit>true</isDebit>
    <postingIndex>2</postingIndex>
  </Posting>
</NewDataSet>
```

As you see here we have only one transaction with two postings.

Have a look to the first posting. There is a tag <docNum>, it's a validation field. Instead of **transactionFID** we use **docNum** field to link posting to transaction. We set **accountCode** field with value "1000", but do not define this account in the XML, it means that such account should be found in the DB (same for second posting with **accountCode** equal "3200"). First posting has a VAT amount and it means it is has VAT link. We use **vatCode** property to make link instead of **vatFID** and **vatAccountCode** field instead of **vatAccountFID**.

So, to import this XML accounts with code 1000, 2000 and 3200 should be presented in our test client, also VAT with code UStn should exists.

Open import window, select XML import, and choose "Custom" import option. But it is not enough to switch on pre validation of source XML. Please check the "With Data source validation" option. After this validation procedure will be executed before importing start.

If you made all correct then you will see the in transaction grid the document with number 2 in the 2006 fiscal year.

Now we will make similar XML but with one difference – the will be the posting with links to accounts which is not presented in database.

The XML is following:

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>

  <Account>
    <code>1000.1</code>
    <name>Test assets account</name>
    <accTypeFID>1</accTypeFID>
    <currencyCode>CHF</currencyCode>
    <accAreaName>TestBilanz</accAreaName>
  </Account>
  <Account>
    <code>2000.1</code>
    <name>Test liability account</name>
    <accTypeFID>2</accTypeFID>
    <currencyCode>CHF</currencyCode>
    <accAreaName>TestBilanz</accAreaName>
  </Account>

  <Transaction>
    <transactionTypeFID>2</transactionTypeFID>
    <docNum>3</docNum>
    <docDate>2006-10-24</docDate>
  </Transaction>

  <Posting>
    <docNum>3</docNum>
    <accountCode>1000.1</accountCode>
    <vatCode>UStn</vatCode>
    <vatAccountCode>2000.1</vatAccountCode>
    <amount>743.5000</amount>
    <fcAmount>743.4900</fcAmount>
    <exchangeRate>1.0000</exchangeRate>
    <vatAmount>56.5000</vatAmount>
    <vatFcAmount>56.5000</vatFcAmount>
    <vatPercent>7.6000</vatPercent>
    <vatQuote>100.0000</vatQuote>
    <text>Posting 1</text>
    <isDebit>false</isDebit>
    <isInclusive>true</isInclusive>
    <postingIndex>1</postingIndex>
  </Posting>

  <Posting>
    <docNum>3</docNum>
    <accountCode>3200</accountCode>
    <amount>800.0000</amount>
    <fcAmount>800.0000</fcAmount>
    <exchangeRate>1.0000</exchangeRate>
    <text>Posting 2</text>
    <isDebit>true</isDebit>
    <postingIndex>2</postingIndex>
  </Posting>
</NewDataSet>
```

As you see the first posting linked to account with code “1000.1”, there is no such account in our test client, that why we define this **Account** instance in our source XML. The account with code “2000.1” was defined also, because of link in the first posting through the vatAccountCode (the real field which contain link value it a **vatAccountFID** which will be recognized during validation of vatAccountCode field). Each account should have a link to the currency we set **currencyCode** equal “CHF”, this currency exists in test client. Since accounts “1000.1” and has a type 1 (Assets) and account “2000.1” has a type 2 (Liabilities) we can put them into the same balance account area. The **accAreaName** field of both accounts has same value – “TestBilanz”. There is no such account area in our test client, so it should be created.

Now as before we open Import window and select XML custom import, select XML source and check “With data source validation” box, and then starts importing.

After successful import new transaction with number 3 should appear in 2006 fiscal year (if you create client with another fiscal year dates range, then please change transaction date (**docDate** field) to the according value).

Now please enter to COA mode and have a look to the account areas. As you see the new one was created – “Test Bilanz”, go to this tab. At the beginning of the tree the new account items appeared with code “1000.1” and “1000.2”. This was happened because we defined new **Account** instances in the source XML document. It is possible to import new account by defining only few attributes of this entity and validation fields (links to entities). Accounts have much more properties, but they with default values and we can to do not define them in XML.

When account records was validating it was linked to Account Area. “TestBilanz” area was not exists in DB. That’s why new one was created. After account area was created, we can get ID of new record and make link in the account record. Really there is no physical link from account to account area. But during accounts validation new **COATreeView** items was created and then linked to account area through the **accAreaFID** field.

To insert new account the CoaTreeView item should be created, because all accounts presented in COA tree view, and hence has a link to CoaTreeView item. During validation was found that there are no appropriate tree items in existing COA, and new items were created. After this mew items should also be validated. Codes of new items are same as accounts code. It is possible to recognize in which account area new COA items should be placed, from the **accAreaName** attribute of the accounts we define in XML. Now COA items has an **accAreaName** attribute filled. The **accAreaName** attribute of **COATreeView** it is also validation field and therefore could be used for recognizing of **accAreaFID** field of **COATreeView** entity. As mentioned there is no “TestBilanz” account area in the DB, that’s why new **AccountArea** instance was created. After validation of new **AccountArea** instance we can link COA item to it. After this validation of **COATreeView** was finished and new **Accounts** was linked to **COATreeView** instances. After inserting of new accounts they were used for linking to the posting of defined transaction.

Since we do not define **COATreeView** items they were inserted with default position of the COA tree – at the beginning.

7. Entities Validation Fields Description

In current chapter you can find the description of validation fields of all entities which has them.

The following columns are used in the description table:

- "Field Name" – name of the validation field, which is used in XML.
- "Field Description" – description
- "Validation Field" – field which could be validated with validation field.
- "Reference Field" – the field which value used for searching entity.
- "Default" – The default value of the validation field

7.1. Account

Field Name	Description	Validation Field	Reference Field	Default
currencyCode	Code of linked currency	currencyFID	[Currency].isoCode	empty string
vatCode	Code of linked VAT	vatFID	[VAT].code	empty string
costCenterCode	Code of linked cost center	costCenterFID	[CostCenter].code	empty string
accAreaName	Name of account area in which linked COA element presented	coaTreeViewFID	[COATreeViewAccountArea].name	empty string
accAreaFID	Link to Account area through the COA element	coaTreeViewFID	[COATreeView].accAreaFID	-1
name	Name of linked COA element	coaTreeViewFID	[COATreeView].name	empty string
Code	Code of linked COA element	coaTreeViewFID	[COATreeView].code	empty string

7.2. Bank

Field Name	Description	Validation Field	Reference Field	Default
countryCode	Code of linked country	countryFID	[Country].code	empty string

7.3. BankAccount

Field Name	Description	Validation Field	Reference Field	Default
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
bankName	Name of linked currency	bankFID	[Bank].name	empty string
bankCode	Code of linked bank	bankFID	[Bank].code	empty string

7.4. BudgetPeriod

Field Name	Description	Validation Field	Reference Field	Default
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
fiscalYearName	Name of fiscal year in which budget period presents	budgetFID	[BudgetFiscalYear].name	empty string
budgetName	Name of budget	budgetFID	[Budget].name	empty string

7.5. COATreeView

Field Name	Description	Validation Field	Reference Field	Default
coaName	Name of COA to which item linked	coaFID	[COA].name	empty string
isMaster	isMaster property of COA to which item linked	coaFID	[COA].isMaster	True
accAreaName	Name of account area in which element presents	accAreaFID	[AccountArea].name	empty string
balanceOrNominal	Whether element linked to balance or nominal account area	accAreaFID	[AccountArea].balanceOrNominal	False
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.6. Creditor

Field Name	Description	Validation Field	Reference Field	Default
freePartyNum	Free number of party to which creditor is linked	partyFID	[Party].freePartyNum	empty string
partyNum	Number of party to which creditor is linked	partyFID	[Party].partyNum	-1
contraAccountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
payTermCode	Code of linked payterm	payTermFID	[PayTerm].code	empty string
payMethodAccountNum	Account number of linked default paymethod	payMethodFID	[PayMethod].accountNum	empty string
payMethodBankCode	Bank code of linked default paymethod	payMethodFID	[PayMethodBank].code	empty string
payMethodAccountCode	Account code of linked default paymethod	payMethodFID	[PayMethodAccountCOATreeView].code	empty string
payMethodName	Name of linked default paymethod	payMethodFID	[PayMethod].name	empty string
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.7. Currency

Field Name	Description	Validation Field	Reference Field	Default
profitAccountCode	Code of linked profit account	profitAccountFID	[AccountCOATreeView].code	empty string
lossAccountCode	Code of linked loss account	lossAccountFID	[AccountCOATreeView].code	empty string

7.8. Debtor

Field Name	Description	Validation Field	Reference Field	Default
freePartyNum	Free number of party to which creditor is linked	partyFID	[Party].freePartyNum	empty string
partyNum	Number of party to which creditor is linked	partyFID	[Party].partyNum	-1
contraAccountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
payTermCode	Code of linked payterm	payTermFID	[PayTerm].code	empty string
payMethodBankCode	Bank code of linked default paymethod	payMethodFID	[PayMethodBank].code	empty string
payMethodAccountCode	Account code of linked default paymethod	payMethodFID	[PayMethodAccountCOATreeView].code	empty string
payMethodName	Name of linked default paymethod	payMethodFID	[PayMethod].name	empty string
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.9. Fiscal Year

Field Name	Description	Validation Field	Reference Field	Default
accountCode	Code of linked result account	resultAccountFID	[AccountCOATreeView].code	empty string
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.10. InterestRate

Field Name	Description	Validation Field	Reference Field	Default
name	Name of interest to which interest rate is linked	interestFID	[Interest].name	empty string

7.11. Invoice

Field Name	Description	Validation Field	Reference Field	Default
docNum	Number of transaction to which invoice linked	invoiceID	[Transaction].docNum	-1
freePartyNum	Free Number of party to which invoice is linked	partyFID	[Party].freePartyNum	empty string
partyNum	Number of party to which invoice is linked	partyFID	[Party].partyNum	-1
payTermCode	Code of linked payterm	payTermFID	[PayTerm].code	empty string
payMethodCurrencyUnit	Unit of currency of account of linked paymethod	payMethodFID	[PayMethodAccountCurrency].unit	1
payMethodFreePartyNum	FreePartyNum of linked paymethod	payMethodFID	[PayMethodParty].freePartyNum	empty string
payMethodAccountCode	Account Code of linked paymethod	payMethodFID	[PayMethodAccountCOATreeView].code	empty string
payMethodName	Name of linked paymethod	payMethodFID	[PayMethod].name	empty string
authorizerName	Name of authorizer	authorizerFID	[UserAccount].name	empty string
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.12. NumRange

Field Name	Description	Validation Field	Reference Field	Default
fiscalYearName	Name of fiscal year in which range presents	fiscalYearFID	[FiscalYear].name	empty string

7.13. Payment

Field Name	Description	Validation Field	Reference Field	Default
paymentNum	Number of linked transaction	paymentID	[Transaction].docNum	-1
bankAccountName	Name of linked bank account	paymentID	[PaymentRunBankAccount].name	empty string
bankName	Name of linked bank	paymentID	[PaymentRunBankAccountBank].name	empty string
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
partyNum	Number of linked party	partyFID	[Party].partyNum	-1

7.14. PaymentRun

Field Name	Description	Validation Field	Reference Field	Default
paymentNum	Number of linked transaction	paymentRunID	[Transaction].docNum	-1
bankAccountName	Name of linked bank account	bankAccountFID	[BankAccount].name	empty string
bankName	Name of linked bank	bankAccountFID	[BankAccountBank].name	empty string

7.15. PayMethod

Field Name	Description	Validation Field	Reference Field	Default
freePartyNum	Free number of party to which paymethod linked though the debtor or creditor	partyFID	[Party].freePartyNum	empty string
partyNum	Number of party to which paymethod linked though the debtor or creditor	partyFID	[Party].partyNum	-1
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
bankCode	Code of linked bank	bankFID	[Bank].code	empty string
bankAccountName	Name of linked bank account	bankAccountFID	[BankAccount].name	empty string

7.16. PayTermDetail

Field Name	Description	Validation Field	Reference Field	Default
code	Code of payterm to which detail is linked	payTermFID	[PayTerm].code	empty string

7.17. Posting

Field Name	Description	Validation Field	Reference Field	Default
userName	Name of linked user	transactionFID	[TransactionUserAccount].name	empty string
docNum	Number of linked transaction	transactionFID	[Transaction].docNum	-1
freeInvoiceNum	Free number of linked transaction	invoiceFID	[Invoice].freeInvoiceNum	empty string
freePartyNum	Free Number of linked party	invoiceFID	[InvoiceParty].freePartyNum	empty string
partyNum	Number of linked party	invoiceFID	[InvoiceParty].partyNum	-1
invoiceNum	Number of linked invoice	invoiceFID	[Invoice].invoiceNum	-1
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string
realVatAccountFID	AccountFID of linked VAT	vatFID	[VAT].accountFID	-1
vatCode	Code of linked VAT	vatFID	[VAT].code	empty string
vatAccountCode	Code of linked VAT account	vatAccountFID	[AccountCOATreeView].code	empty string

7.18. Reminder

Field Name	Description	Validation Field	Reference Field	Default
partyNum	Number of linked party	partyFID	[Party].partyNum	-1
currencyCode	Code of linked currency	currencyFID	[Currency].isoCode	empty string
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.19. Transaction

Field Name	Description	Validation Field	Reference Field	Default
userName	Name of linked user	userFID	[UserAccount].name	empty string

7.20. VAT

Field Name	Description	Validation Field	Reference Field	Default
quote	VAT quote	vatID	[VATRate].quote	100
rate	VAT rate	vatID	[VATRate].rate	0
accountCode	Code of linked account	accountFID	[AccountCOATreeView].code	empty string

7.21. VATRate

Field Name	Description	Validation Field	Reference Field	Default
code	Code of the VAT to which rate is linked	vatFID	[VAT].code	empty string