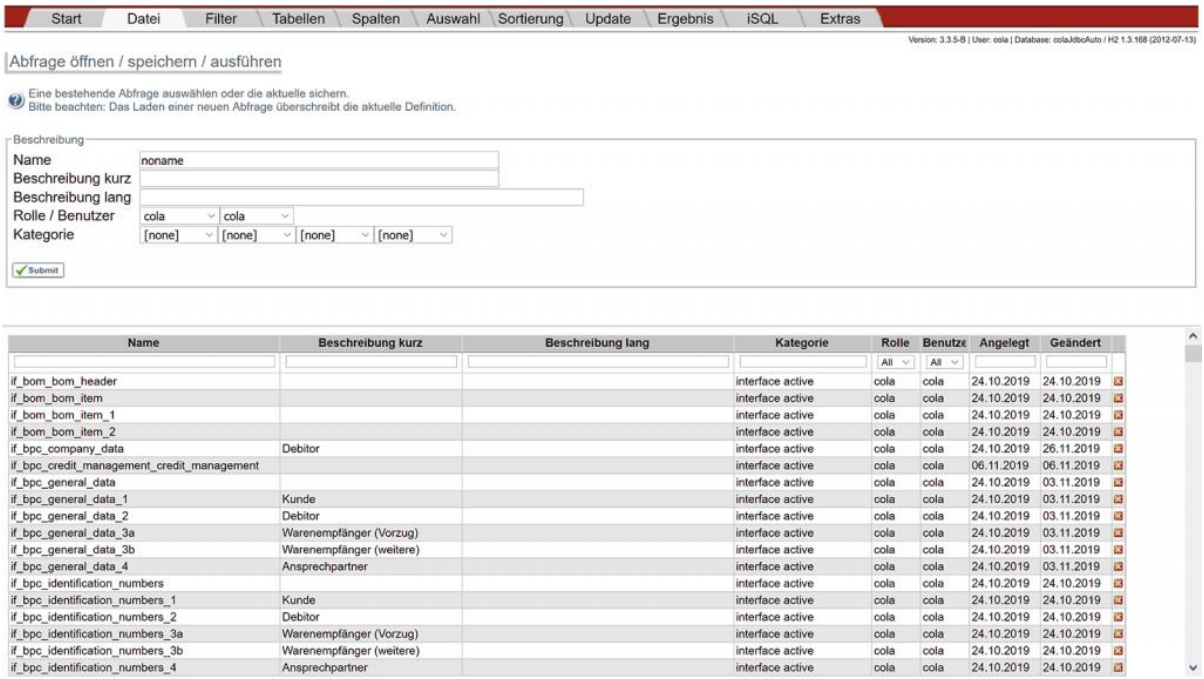
The migration tools of bit-impulse

Good news first: The migration tools described below are free of charge for my customers! If I am commissioned for migration projects, these tools are part of my "toolbox" and included in the daily rate!

The migration tools from bit-impulse (by me) are based on two main pillars: iQuery, with further components, as a so-called ETL tool and sub-way for the administration of the migration objects.

iQuery – as ETL tool



iQuery – Overview of existing queries

An ETL tool is the heart of the technical data migration. The three letters stand for **e**xtraction of data from different sources, **t**ransformation of the data into the schema and format of the target system, and **l**oading of the data into the target system.

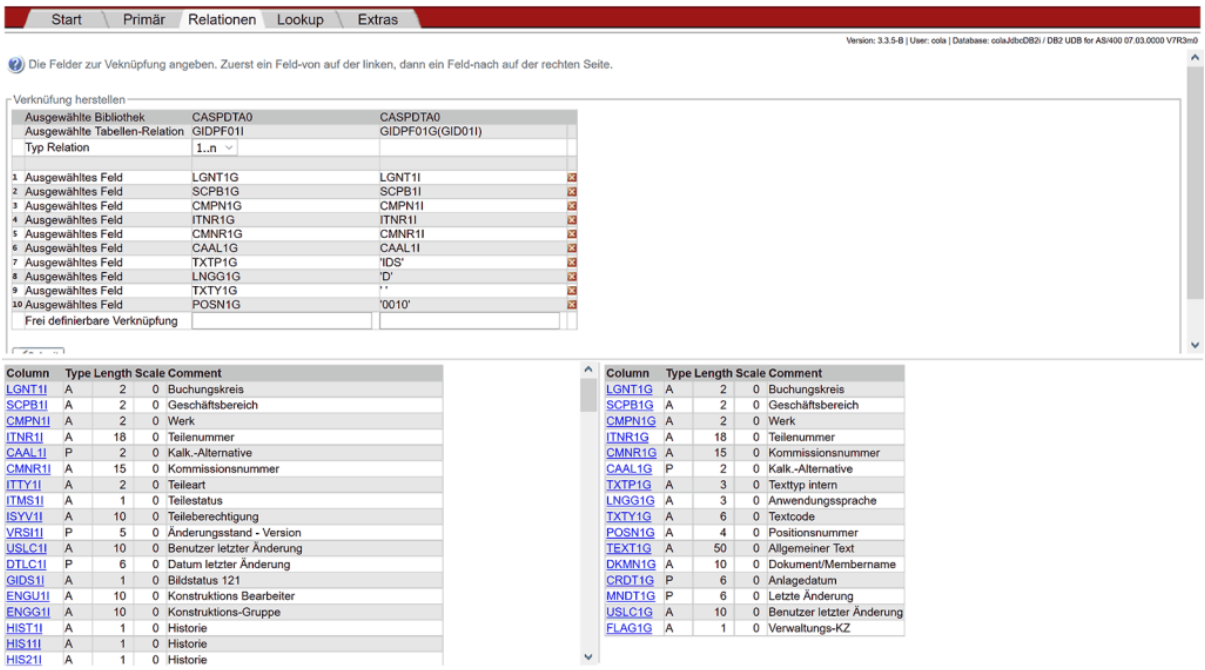
iQuery fully covers all three required steps, but in most projects focuses on the extraction and transformation steps. Loading of the data is usually done with the import tool of the target system manufacturer.

iQuery extracts the data exclusively from SQL-enabled databases. Wherever the data is stored in another form, e.g. in Excel, such data is therefore first transferred to a database enclosed with the iQuery.

bit-impulse
Ralf Tossenberger
Keuperweg 9
73614 Schorndorf

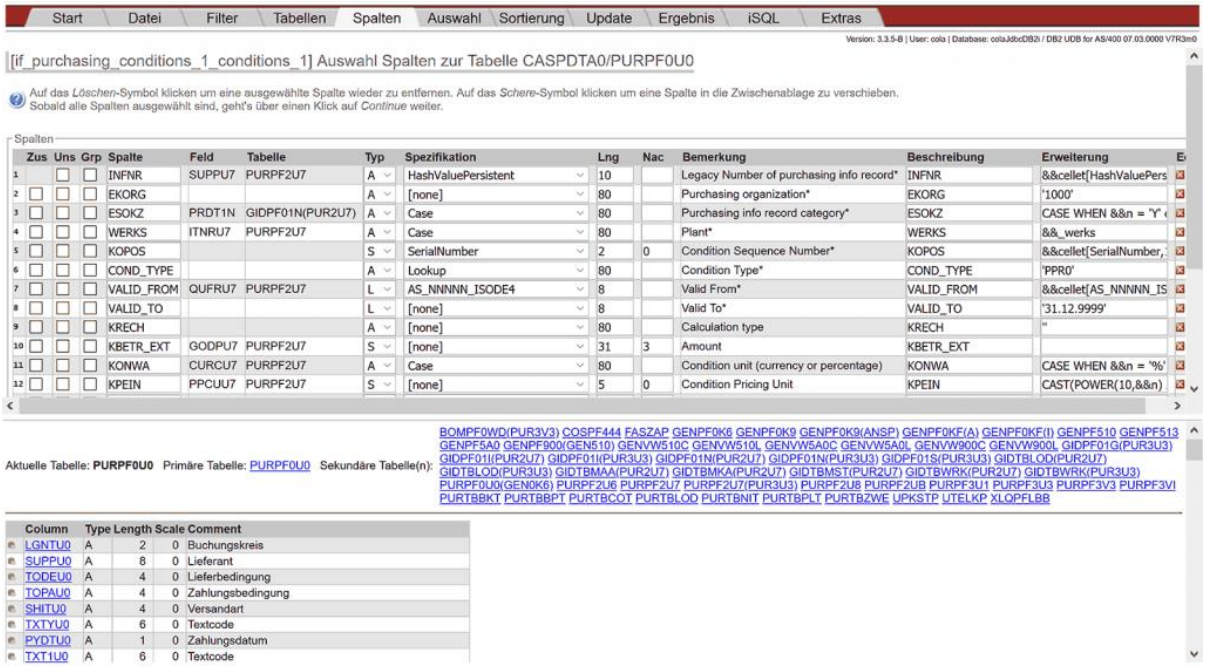
Fon: +49 7181 477010
Fax: +49 7181 477011
www.bit-impulse.de
ralf.tossenberger@bit-impulse.de

The data required for extraction can come from several source systems. iQuery can access these simultaneously in a single query. This allows, for example, the customer data of an ERP system from an SQL Server database to be linked with the CRM data from an Oracle database.



iRepository – Example of a link

The development of an export always starts from a primary SQL table, e.g. that of the customer. At a central location, in iRepository, the link between exemplary customer > address and address > country master takes place. This means that for ALL exports, starting with the primary table Customer, address and country master data can be retrieved.



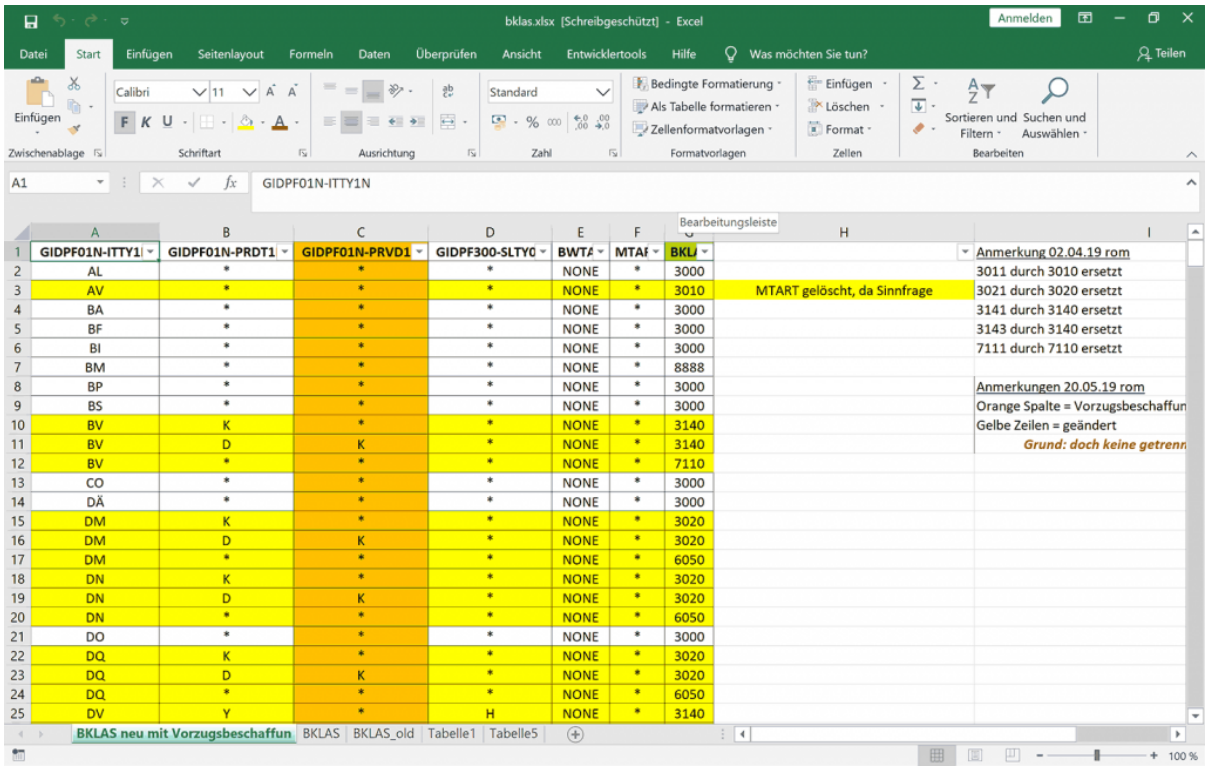
Zus	Uns	Grp	Spalte	Feld	Tabelle	Typ	Spezifikation	Lng	Nac	Bemerkung	Beschreibung	Erweiterung
1			INFNR	SUPPU7	PURPF2U7	A	HashValuePersistent	10		Legacy Number of purchasing info record*	INFNR	&&cellet[HashValuePers
2			EKORG			A	[none]	80		Purchasing organization*	EKORG	'1000'
3			ESOKZ	PRDT1N	GIDPF01N(PUR2U7)	A	Case	80		Purchasing info record category*	ESOKZ	CASE WHEN &&n = 'Y'
4			WERKS	ITNRU7	PURPF2U7	A	Case	80		Plant*	WERKS	&&_werks
5			KOPOS			S	SerialNumber	2	0	Condition Sequence Number*	KOPOS	&&cellet[SerialNumber,
6			COND_TYPE			A	Lookup	80		Condition Type*	COND_TYPE	'PPR0'
7			VALID_FROM	QUFRU7	PURPF2U7	L	AS_NNNNN_ISODE4	8		Valid From*	VALID_FROM	&&cellet[AS_NNNNN_IS
8			VALID_TO			L	[none]	8		Valid To*	VALID_TO	'31.12.9999'
9			KRECH			A	[none]	80		Calculation type	KRECH	''
10			KBETR_EXT	GODPU7	PURPF2U7	S	[none]	31	3	Amount	KBETR_EXT	
11			KONWA	CURCU7	PURPF2U7	A	Case	80		Condition unit (currency or percentage)	KONWA	CASE WHEN &&n = '%'
12			KPEIN	PPCUU7	PURPF2U7	S	[none]	5	0	Condition Pricing Unit	KPEIN	CAST(POWER(10,&&n)

Column	Type	Length	Scale	Comment
LGNTU0	A	2	0	Buchungskreis
SUPPU0	A	8	0	Lieferant
TODEU0	A	4	0	Lieferbedingung
TOPAU0	A	4	0	Zahlungsbedingung
SHITU0	A	4	0	Versandart
TXTYU0	A	6	0	Textcode
PYDTU0	A	1	0	Zahlungsdatum
TXT1U0	A	6	0	Textcode

iQuery – Example column definition

For each object to be exported, e.g. BOM header, BOM position, BOM position text, an SQL query is created in iQuery. The output fields are arranged according to the import specifications of the manufacturer of the target system. This also applies to the basic formatting of these fields, as alpha fields, numeric fields, date fields, etc., in the respective desired formatting.

The connection between source and target fields is then different. In the simplest case, a direct assignment is made, as for example with the customer name. The country key may not be directly assignable and must be converted from two-digit to three-digit. To make the translation rule known to the tool, an Excel lookup is stored there. The left column(s) of this Excel then contains the previous country key, the right column the new one.



	A	B	C	D	E	F	H	I
1	GIDPF01N-ITTY1	GIDPF01N-PRDT1	GIDPF01N-PRVD1	GIDPF300-SLTY0	BWTA	MTAA	BKLA	Anmerkung 02.04.19 rom
2	AL	*	*	*	NONE	*	3000	3011 durch 3010 ersetzt
3	AV	*	*	*	NONE	*	3010	MTART gelöscht, da Sinnfrage 3021 durch 3020 ersetzt
4	BA	*	*	*	NONE	*	3000	3141 durch 3140 ersetzt
5	BF	*	*	*	NONE	*	3000	3143 durch 3140 ersetzt
6	BI	*	*	*	NONE	*	3000	7111 durch 7110 ersetzt
7	BM	*	*	*	NONE	*	8888	
8	BP	*	*	*	NONE	*	3000	Anmerkungen 20.05.19 rom
9	BS	*	*	*	NONE	*	3000	Orange Spalte = Vorzugsbeschaffun
10	BV	K	*	*	NONE	*	3140	Gelbe Zeilen = geändert
11	BV	D	K	*	NONE	*	3140	Grund: doch keine getrenn
12	BV	*	*	*	NONE	*	7110	
13	CO	*	*	*	NONE	*	3000	
14	DÄ	*	*	*	NONE	*	3000	
15	DM	K	*	*	NONE	*	3020	
16	DM	D	K	*	NONE	*	3020	
17	DM	*	*	*	NONE	*	6050	
18	DN	K	*	*	NONE	*	3020	
19	DN	D	K	*	NONE	*	3020	
20	DN	*	*	*	NONE	*	6050	
21	DO	*	*	*	NONE	*	3000	
22	DQ	K	*	*	NONE	*	3020	
23	DQ	D	K	*	NONE	*	3020	
24	DQ	*	*	*	NONE	*	6050	
25	DV	Y	*	H	NONE	*	3140	

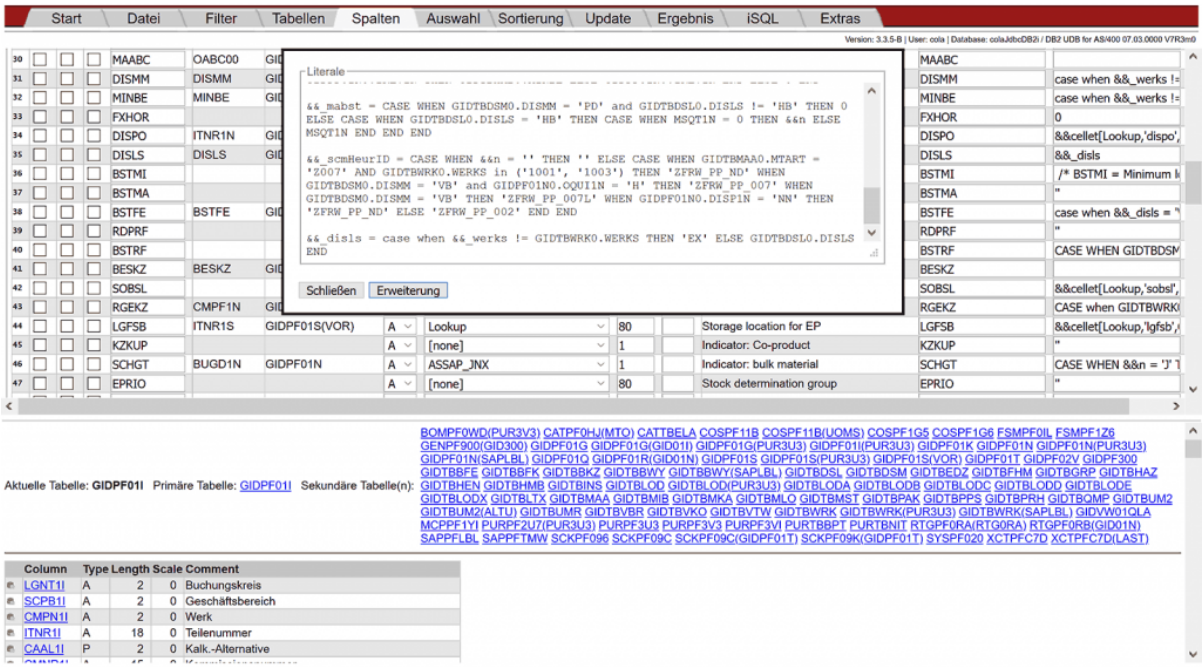
iQuery – Example of a (complex) lookup

Lookups are an important backbone of iQuery and can be used in a variety of ways. For example, it is always necessary to "invent" data that is required by the target system but is not available in the source system. Such data can be "generated" on the basis of lookups. In such a lookup the desired initial values are defined based on the pattern of several input values.

One of the best things about lookups is that these Excel spreadsheets can be maintained by the specialist department without the need for training in any tools!

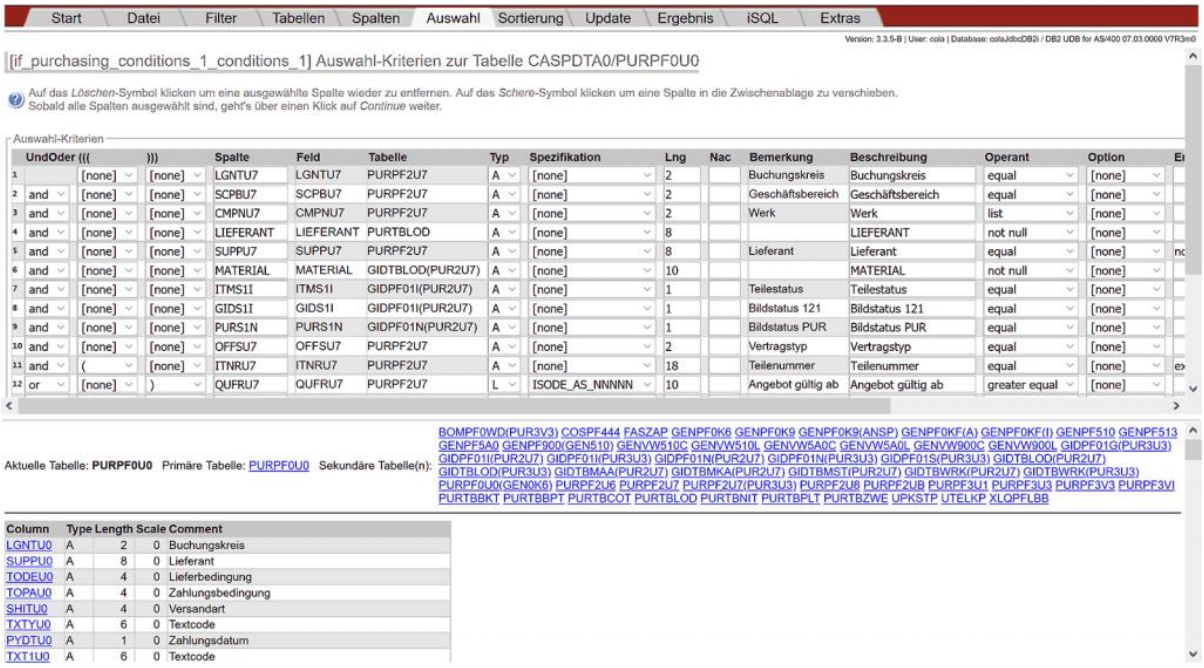
In this way, 80 to 90 percent of the data required for export can be "clicked together" with iQuery very quickly. For the more " difficult " cases, there are several procedures:

For recurring tasks, iQuery can be extended, both generally and project-specifically. For example, SAP expects conversions between units of measure as a fraction (numerator and denominator in two fields) and not as a decimal number, as is usually the case. For this purpose, iQuery was extended by this, by no means trivial, functionality: Wherever necessary, the determination of a value can also be done directly in SQL (fragments), which looks like this as an example: **`CASE WHEN field < 3 THEN 'A' WHEN field < 7 THEN 'B' ELSE 'C' END`**. iQuery then inserts these individual SQL fragments into the generated SQL statement.



iQuery – Example of a complicated query

Even where things get really complicated in individual cases, iQuery stays on the ball. Long and confusing SQL expressions can be broken down into so-called literals. These literals can then be accessed, similar to text containers. Literals can contain one or more literals. All this simplifies the complexity, which is not always avoidable in individual cases.



iQuery – Example selection definition

Usually not all data sets can be transferred from the source system. Delimitations can be made, for example, by the current status or the timestamp in the data records. Such selections can be easily "clicked together" in iQuery.

But even complex cases can be mapped. For example, a BOM should only be transferred if ALL BOM items have a certain minimum status. Here too, the query can be extended by an SQL snippet such as `EXISTS SELECT * FROM table … WHERE … AND STATUS >= 10`.

Exports created with iQuery can be validated "on sight" before import into the target system. Typically, the export is converted to Excel and this Excel is then stored centrally or distributed directly. Here iQuery supports the current Excel XLSX format, which allows up to 1.4 million rows in an Excel document.

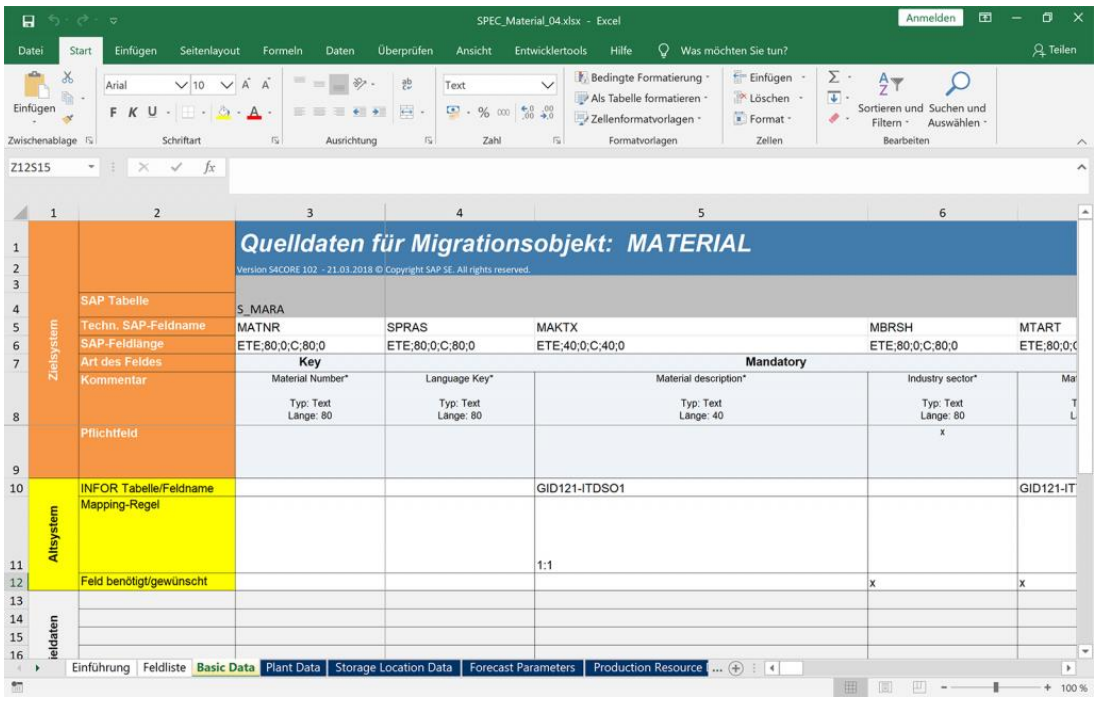
Finally, the export is done into the desired, technical target format for the import prepared by the manufacturer. iQuery offers various output adapters, e.g. for CSV in all variations, Excel, XML data and also directly into database tables.

bit-impulse
Ralf Tossenberger
Keuperweg 9
73614 Schorndorf

Fon: +49 7181 477010
Fax: +49 7181 477011
www.bit-impulse.de
ralf.tossenberger@bit-impulse.de

sub-way – as administration tool

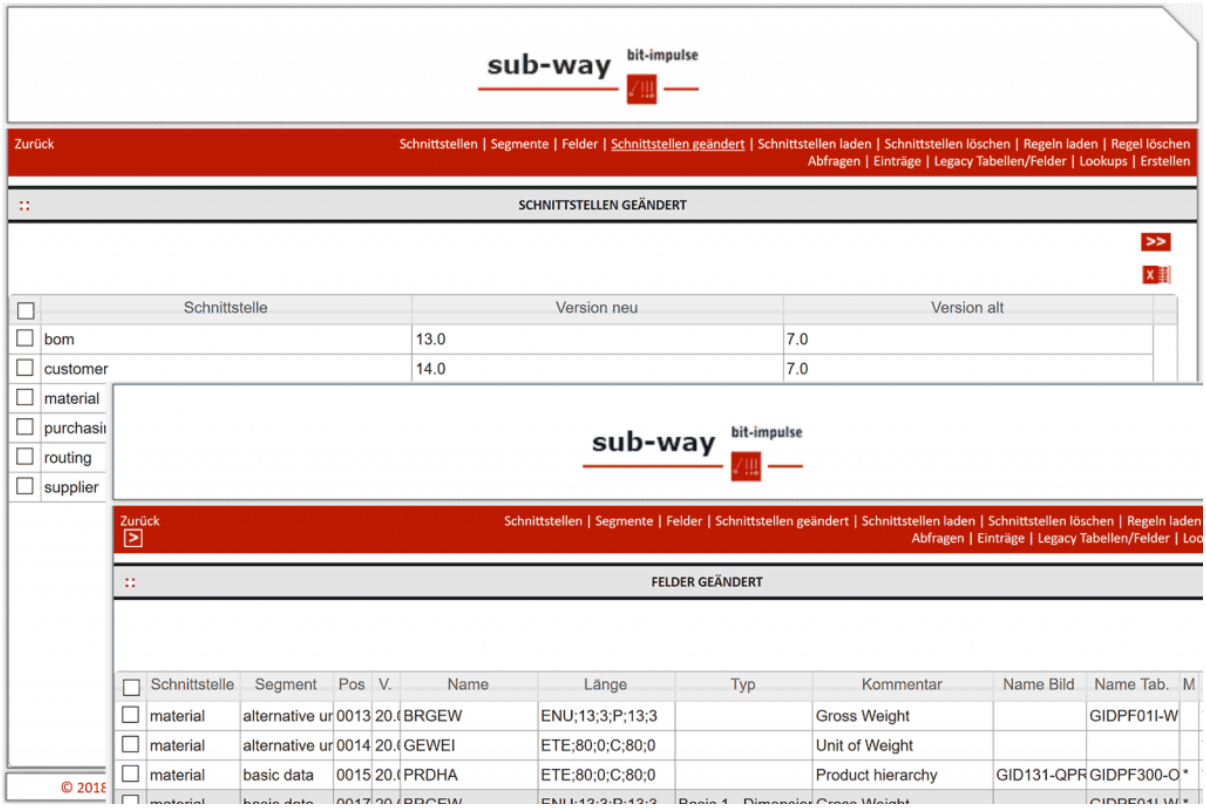
Administration and documentation – both terms are rather at the lower end of the popularity scale for migrations than at a popular top position. sub-way therefore contains some functionality to "reward" its application immediately.



sub-way – Example of migration documentation

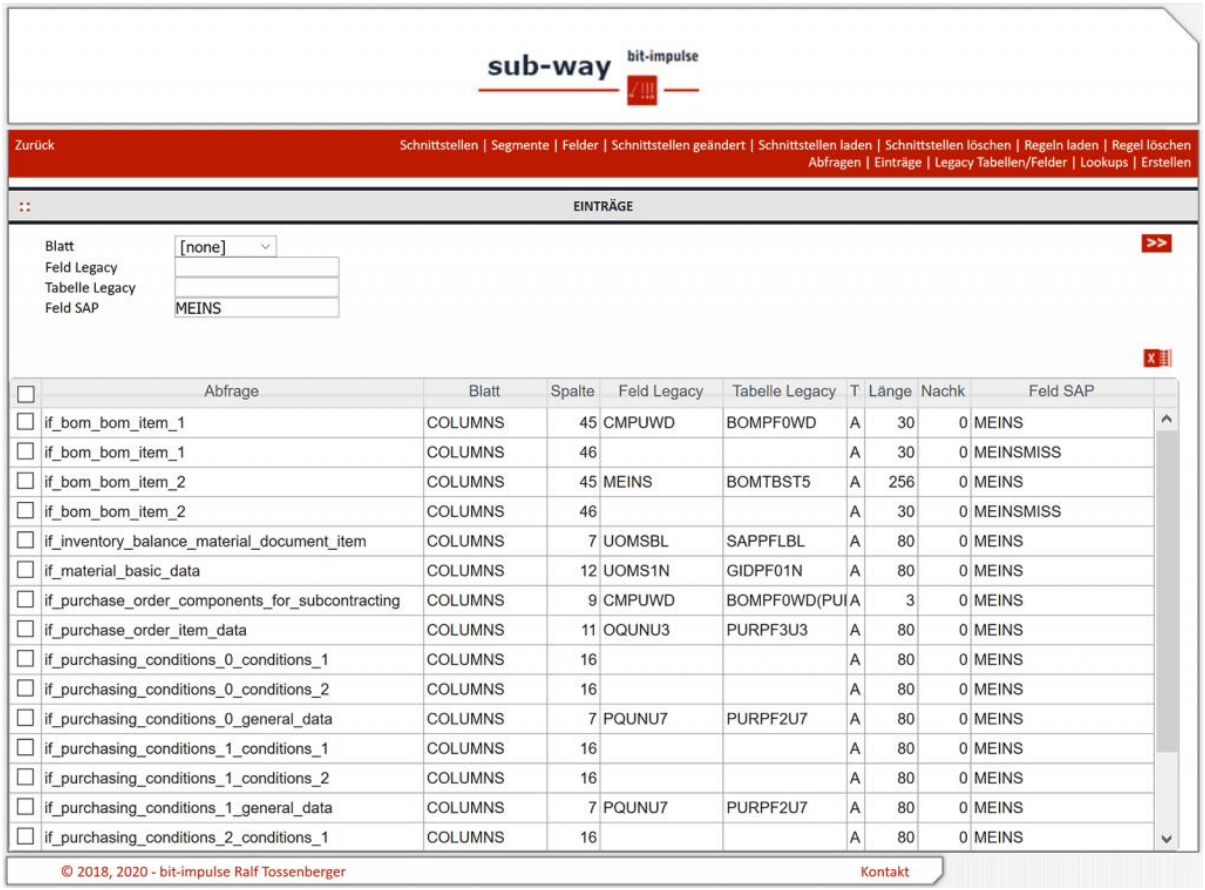
The migration objects, e.g. customer and address, have to be described in some form. Minimum are the technical properties of the target fields (datatype, length, ...) and the migration-rule to the respective target field.

If this description is done in Excel according to a template provided by sub-way, the documentation and the tool can interlock. If at least the technical properties of the fields of the target system are described clearly, the tool can automatically create the SQL queries of the export in their basic structure and thus save a lot of typing work. It also helps to avoid errors in the otherwise manual transfer.



sub-way – Example of a version comparison

sub-way allows a versioning of the interface objects. This option is very useful for the typically iterative process of defining migration rules. The rules are defined, implemented in the tool, an export is performed. After validation of the export on sight or after a test import into the target system, deficiencies in the rules are detected and adjusted. sub-way can detect the differences of a newer version and provide the differences as an "activity list" on field level for the adjustment of the exports.



sub-way – Example of a where-used list

As a rule, migrations must be documented; for example, the auditor requires a transparent view of the transfer of data from the old to the new ERP.

With sub-way, the documentation for the migration can be generated in iQuery at any time based on the existing exports. The documentation then shows which data from the source system, with which translation rules, was exported for the target system.

In addition, a where-used list can be generated for the existing export queries. Which source system tables and fields or which target system fields were used in which queries? Where were which lookups used?

Collection of further information

The migration tools of bit-impulse are Java-based and therefore run on practically any platform. The application/web server necessary for the browser applications can either "run" on an existing server or a separate server is set up for this purpose. The requirements on system resources are quite moderate.

For the actual export, several queries usually have to be executed in a certain order. For this purpose, CMD files are created on the server in which the individual activities are called up using iQuery commands. The desired properties can be specified in detail by command parameters. For the export, e.g. into the Excel-CSV format, it looks like this:

```
DBToCSV [jdbc:]library/table | [jdbc:]iquery csvfile [key:value]
[charset] [semicolon-delimited|comma-delimited|tab-delimited]
[decimal-comma|decimal-point] [no-quotes] [no-header] [crlf|lf|cr]
[supdate|sdelete] [debug]
```

Where iQuery is used as an ETL tool, it is usually also used as a reporting tool for the migration project. With iQuery, the source data for the creation of the transformation rules can first be viewed according to various criteria. Maybe data has to be adjusted in the source system beforehand – for this purpose iQuery can be used to create the necessary to-do lists.

The migration tools from bit-impulse can be used in projects of different sizes. There were the smallest projects where the software simply ran on the consultant's laptop, up to large projects in a large team with up to 373 queries and 102 lookups.