



Simultaneous meta and data manipulation in Blaise

Marien Lina
Statistics netherlands



1. Separate meta and data handling



Data and Meta



used to be separated issues...

data handling



meta handling



In separate applications...



data handling

- DEP
- Manipula



meta handling

- Delta
- Cameleon
- Structure Browser



2. The blaise API



Blaise API

Enables you to:

- Combine data and meta handling in one VB application
- Use the VB application in Blaise and Manipula



Blaise API



For some there may be some constraints to use the API:

- Programming skills (VB) required
- API Components must be registered on every machine that uses the application



3. Calling Manipula from the DEP in Blaise 4.7



Blaise 4.7



Blaise 4.7 feature:

- Calling Manipula in the DEP

This could be activated
via the DEP menu.



Blaise 4.7



If you activate Manipula calls via the DEP menu (in Blaise 4.7) you can use the setting:

INTERCHANGE=TRANSIT

in the Manipula setup.



Blaise 4.7



- INTERCHANGE=TRANSIT

A copy of the current DEP record is available in the Manipula. Changed values are returned to the DEP.



4. Calling Manipula from the DEP in Blaise 4.8



Blaise 4.8



- New development: BASIL
- A downloadable Data Entry Program that runs on its own
- With API like functionality and layout features
- Without having to install API components on every machine



Blaise 4.8



Solution:
More flexible communication
between Rules and Manipula.

Making it possible to call
Manipula **procedures** directly
from the rules



How?



Call a Manipula procedure as:

- Alien Procedure
- Alien Router in the rules
- Alien Router connected to a field type (datamodel properties)



Notice



In Blaise 4.8 you can use
INTERCHANGE=SHARED

The current DEP record is available in Manipula,
including routing and error information!
It is not a copy!

In Blaise 4.7 with
INTERCHANGE=TRANSIT

The DEP record was **copied** to manipula



Blaise 4.8



Running Manipula in the DEP

Example



Write a backup of the current record in a temporary file



5. Meta in Manipula



Meta in Manipula



Blaise 4.8 introduced new methods for addressing meta information in Manipula

Makes it possible to combine meta and data information in one Manipula setup



Meta in Manipula



Main methods:

- GETFIELDINFO
- GETMETAINFO
- GETTYPEINFO

Can for example be used to disclose:

Field names, answer names, values, datamodel name, texts, types, category names, value range, current language, field text in language[x], field tags and many things more



Meta in Manipula



- Use Manipula to Address META

Example





Meta in Manipula



- Combining Meta and Data, Manipula can produce files that contain meta and data (XML).
- Manipula may be used to write setups for the same purposes as Cameleon scripts



6. Meta in the Rules



Meta in the Rules



How to use Meta in the rules?

- Manipula reads Meta info
- Rules calls Manipula
(alien procedures or routers).



Meta in the Rules



Example: read the name of an enumeration and assign it to a field

Example





Meta in the Rules



The example used the method `GETTYPEINFO` to get the category name when you know a category number.



7. Flexible Meta



Flexible MSU's



USES MyMeta

Using this syntax, a manipula setup always needs to be prepared again to use it for another meta (or after a change of the checksum).



Flexible MSU's



USES MyMeta (VAR)

Using this syntax, you can pass ANY meta file name to the setup and it will not be rejected for manipulation.



Flexible MSU's



USES MyMeta (VAR)

The Meta information of MyMeta is not included in the MSU file.

The BMI is connected at run-time.

This is called “Late Binding”.



Flexible MSU's



USES MyMeta (VAR)

When calling Manipula in the alien procedure, the current meta must be connected to MyMeta (the variable meta name in Manipula).

/KMyMeta=\$dictionaryname



Meta in the Rules



Adapted example of using a variable meta (VAR)

Example





7. Conclusion



Conclusion



Powerful extension in 4.8

- Combination of data and meta information in Manipula
- Flexible way to define Manipula calls in the rules
- Making meta information available to the DEP



Thank You