Data Scanner Update

An Old Problem Now Better Solved

Many of us know the problem that a print job is delivered by a protocol which does not transfer a job name. TCP direct socket or IBM channel are the most common known of such types of input connectivity. Nevertheless in some cases a qualified job name is required, maybe for archiving or further processing of the printed data.

The solution:

By determining data from the printing data stream, you may build up your own job name. With the Mercury Data Scanner you will get an excellent tool for this - and you will be even able to define the destination address of your print jobs depending on the printing data.

Dynamically Using Determined Content Data

Since version 4.0d it is supported to determine content data using the Mercury Data Scanner and use this as e.g. print job names. Until now this was just a simple 1:1 assignment. With the upcoming version 4.11 we will provide significantly more features.

Now search patterns can assist the recognition of content data. Data fields recognised from the printing data stream may now be combined to strings. At the same time a set of functions for managing these strings is now available as also a conditional assignment. If a condition is true, variant A is used, if not then variant B is used.

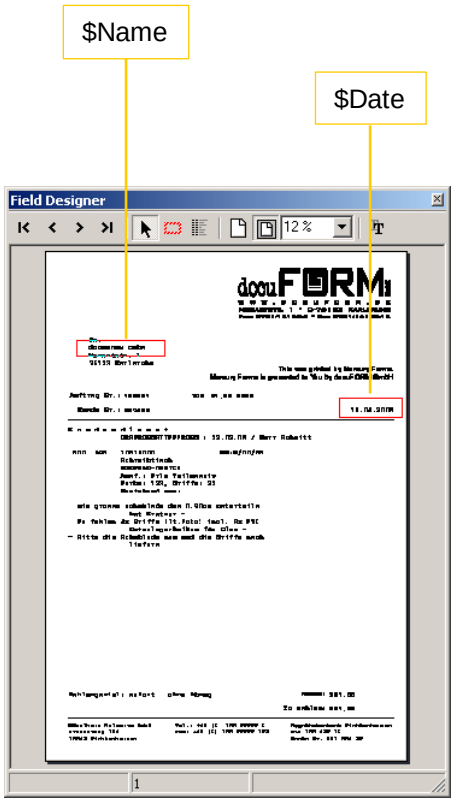
“Prefix-”+ $Feld1 + “-” + $Feld2 + “-Suffix”

LEFT ($Feld1;3) + UPPER($Feld2) + RIGHT($Feld3;3)

IF ($Feld1=“Test" & $Feld2=“Admin"; “Testjob"; $User)

VAL(string)
TEXT(value)
LENGTH(string)
LEFT(string [; #chars])
MID(string; pos; #chars)
RIGHT(string [; #chars])
LOWER(string [;#chars])
UPPER(string [;#chars])

Example Job Name - dynamically built



The field *Name* will be recognised within the left red frame.

The field *Date* within the right red frame, using the search pattern „*1-2 digits; slash; 1-2 digits; slash; 2-4 digits*“:

‚*\d{1+}/ \d{1+}/ \d{2+}*‘

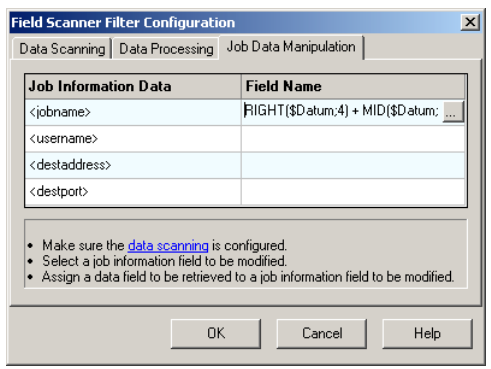
The following expression builds the job name from the date and the first six letters of the name in upper case. The date was rotated to be able to be used for sorting

```
RIGHT($Date;4) + LEFT($Date;2) + MID($Date;3;2) + UPPER(LEFT($Name;6))
```

Name = ‚**docuFORM**‘

Date = ‚**01/16/2006**‘

Jobname = ‚**20060116DOCUFO**‘



Summary

- Usage of search patterns improves accuracy during recognition of content data
- String functions for combination of recognised content data
- Building own job names from content data if e.g. a job name is not available from the input protocol
- Recognising the user name from content data if e.g. a user name is not delivered by the input protocol
- Control of the destination address of a print job. E.g. sending as e-mail with the SMTP output to an address, which was recognised from the content of the print job
- and much more...