BW on HANA Transformations with AMDP Script: Start Thinking Set Based

In a quest to speed up BW transformations, we investigated AMDP (ABAP Managed Database Procedures). By using it, we want to achieve complete push-down of transformation execution. HANA-based transformations have been around since BW 7.4 SP09. However, only as of BW 7.5 SP04, you can choose whether to unleash the full power of your HANA based setup. Start, End and Expert routines can now be written in two languages: ABAP or SQLScript.

How it all starts

Go ahead, take a deep breath and click the right button to start your journey. It will allow you to create an ABAP Managed Database Procedure. This means that within an ABAP shell, a piece of SQLScript can be written. The advantage of AMDP

Sharing is caring

(https://www.facebook.com/dialog/feed?app_id=561659087587[illegible]&link=https://www.just-bi.nl/bw-hana-transformations-amdp-script-start-thinking-

(https://twitter.com/intent/tweet?text=https://www.just-bi.nl/bw-hana-transformations-amdp-script-start-thinking-

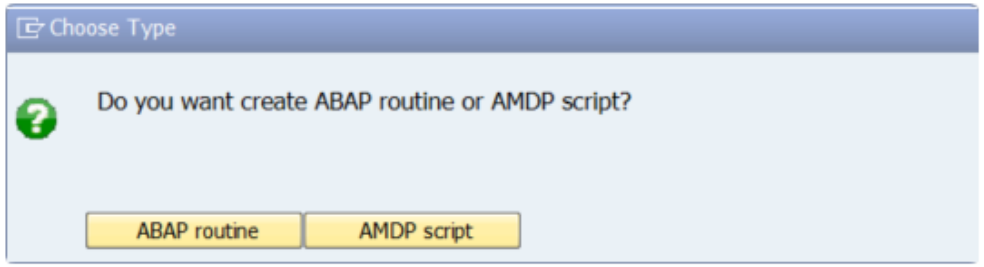
(https://www.just-bi.nl/bw-hana-transformations-amdp-script-start-thinking-

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.

See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

thinking-set-based/&decription=&redirect_uri=https%3A%2F%2Fwww.just-bi.nl%2Fbw-hana-transformations-amdp-script-start-thinking-set-based%2F#facebook-close&display=popupary=&source=https://www.just-bi.nl/bw-hana-transformations-amdp-script-start-thinking-set-based/)

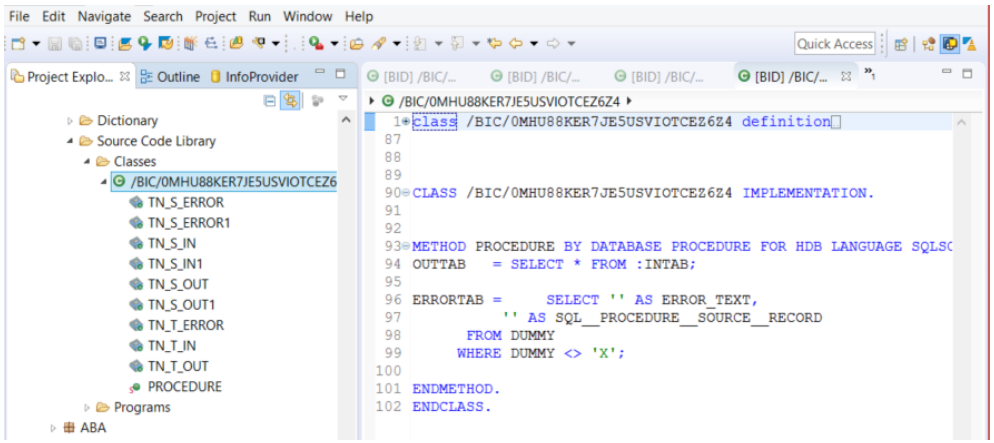
over ABAP is that logic will now be executed on the database level, instead of on the application layer.



(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure1-amdp.png)

Figure 1 Choose AMDP here

After pressing the right button, an ABAP class is created in your personal $TMP folder. If you are working in eclipse already, you will be taken to the class directly. If you are using the SAP GUI separately, you can find your class in an Eclipse ABAP project using the search functionality.

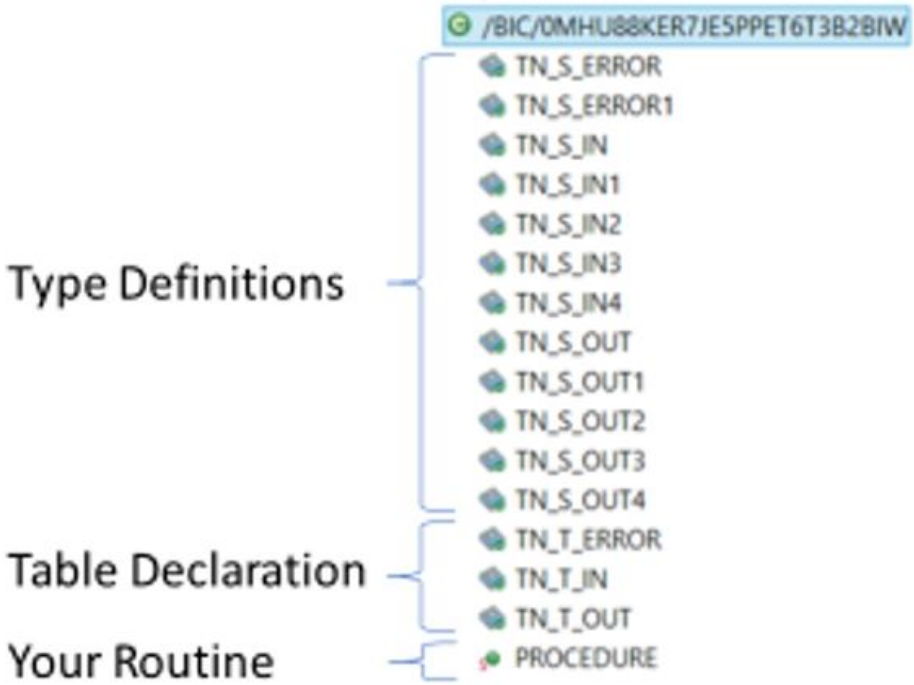


(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure2-amdp.png)

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.
See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

Figure 2 Overview of the AMDP editor and class structure

The class you open, consists of a few parts. Similar to ABAP routines, these classes contain a 'sort of' source package and/or a result package: Two tables are defined, named the inTab and the outTab. Based on whether you created a start, end or expert routine, these are located before the structure of the source, in between the source and the target, or after the target.



(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure3-amdp1.png)

Figure 3 The structure of the routine class

Setting up the actual routine

The following line of code is the starting point for your routine:

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.

```
METHOD PROCEDURE BY DATABASE PROCEDURE FOR HDB LANGUAGE SQLSCRIPT USING
```

See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure4-amdp.png)

Figure 4 The start of our ABAP Managed Database Procedure

Below this line, it is up to you to create a piece of SQLScript that pulls data from the inTab into the outTab. Make sure to address all fields in the inTab and outTab in the right order, just like you would have to do in ABAP based routines. A good starting point is to create a piece of SQLScript that selects all fields (SELECT *) from the inTab into the

outTab, and adjust it from there. When you need to name the fields you select, make sure to put them in the right order. Of course, if you have access to the HANA backend, you could generate a SELECT statement including all the fields from the catalog directly.

```
outTab = SELECT
"DELIV_ITEM",
"DELIV_NUMB",
"RECORDMODE",
"ACT_GI_DTE",
"BASE_UOM",
"BATCH",
"BILBLK_DL",
"BILLTOPRTY",
"BILL_BLOCK",
"BUS_AREA",
"BWAPPLNM",
```

(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure5-amdp.png)*Figure 5: ..and so on and so forth*

Once you have done this tedious exercise, we need to do one more thing. There is an errorTab to identify problematic records. If we decide not to use error handling yet, we can bypass it by filling it with emptiness. Then just save and activate your class, activate your transformation and give your transformation a go by starting the DTP.

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.

See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

```
errorTab = SELECT
'' as "ERROR_TEXT",
'' as "SQL__PROCEDURE__SOURCE__RECORD"
FROM :inTab;
```

(//www.just-bi.nl/wp-content/uploads/posts/2017/post_5271/figure6-amdp.png)

Figure 6 Bypassing the error tab

The value of AMDP

So, is it currently worth it to convert your existing routines into AMDP, or to even use it when creating new ones? Probably. We use SQLScript because its execution can be optimized by HANA. The records from the tables are not processed line by line, but simultaneously. And of course, data is not transferred to the application server for calculations. Performance of master data lookups could potentially be enhanced by simply doing a join with the table containing the master data, instead of looping over the master data in an ABAP routine. Similarly, unit conversions can be done by joining the table with the UOM master data and performing the arithmetic in your SELECT statement.
When we are asked for more complex logic, there is an important pitfall. If you are used to ABAP routines, you may be tempted to process the data line by line with so-called imperative logic. These loops, if-statements and so on are not executed in parallel, but sequentially. In this case, we do have the performance increase of the calculation being executed in the database, but not the parallel processing power. The trick is to let go of this record-based way of thinking, and opting for a declarative (set-based) method. For example, when trying to transpose data from account-based to key figure based, I was tempted to do it line by line. However, after giving it some more thought I was able to come up with a set-based solution which involved unioning a table with itself multiple times.

To close it all off, I believe there is value in using AMDP when building transformations that cannot be done using standard functionalities. Try to let go of the record-based mindset and use regular SQL statements to build your routine. Then you will make use of the full power that BW on HANA has to offer.

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.
5022 views
Go to overview (https://www.just-bi.nl/inspiration-feed/)
See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

This article belongs to

Blog Data Management

HANA & BW/4HANA

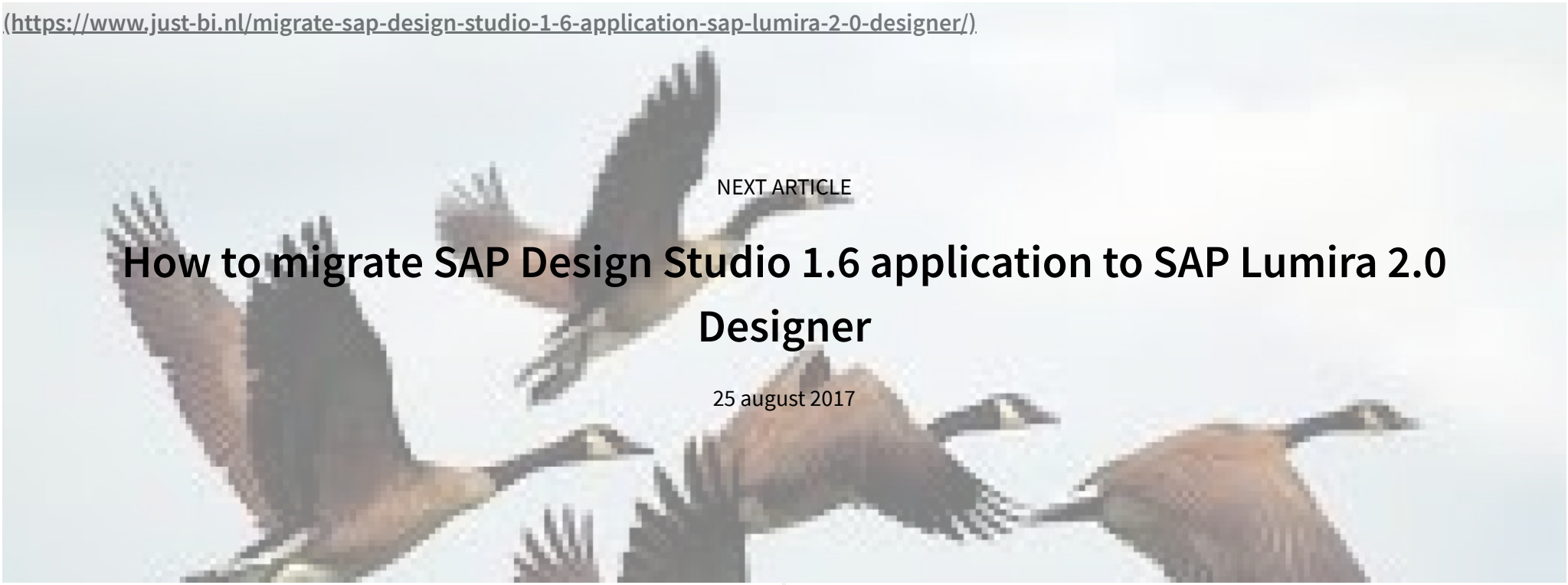
Tags

AMDP BW HANA SAP

Author

Paul Schoondermark

(https://www.just-bi.nl/migrate-sap-design-studio-1-6-application-sap-lumira-2-0-designer/)



NEXT ARTICLE

How to migrate SAP Design Studio 1.6 application to SAP Lumira 2.0 Designer

25 august 2017

View all expert fields (https://www.just-bi.nl/expertise/)

View all our services (https://www.just-bi.nl/services/)

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.

See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |

(https://www.just-bi.nl)

Home (https://www.just-bi.nl/) Our expertises (https://www.just-bi.nl/expertise/)
Services (https://www.just-bi.nl/services/) About (https://www.just-bi.nl/about/)
Partners (https://www.just-bi.nl/partners/) Inspiration feed (https://www.just-bi.nl/inspiration-feed/)
Get in touch with us (https://www.just-bi.nl/contact/) Just-Care (http://www.just-care.nl/)

© 2019 Just-BI

General Data Protection Regulation (https://www.just-bi.nl/general-data-protection-regulation/)

Terms & Conditions (https://www.just-bi.nl/terms-conditions/)

Follow us (https://www.linkedin.com/company/just-bi/)

By continuing to browse our website you are agreeing to our use of cookies.
For more details about cookies see our cookie policy.

See details (https://www.just-bi.nl/privacy-policy/) | Accept Decline |