

Week 4 Unit 2: ABAP Managed Database Procedures



ABAP Managed Database Procedures

Outline

Content

- ABAP Managed Database Procedures
 - Concept
 - Definition
 - Consumption
- Exception Handling
- Additional Information:
 - Debugging AMDPs
 - AMDP Extension Concept



ABAP Managed Database Procedures

Migration to SAP HANA

1. Detect

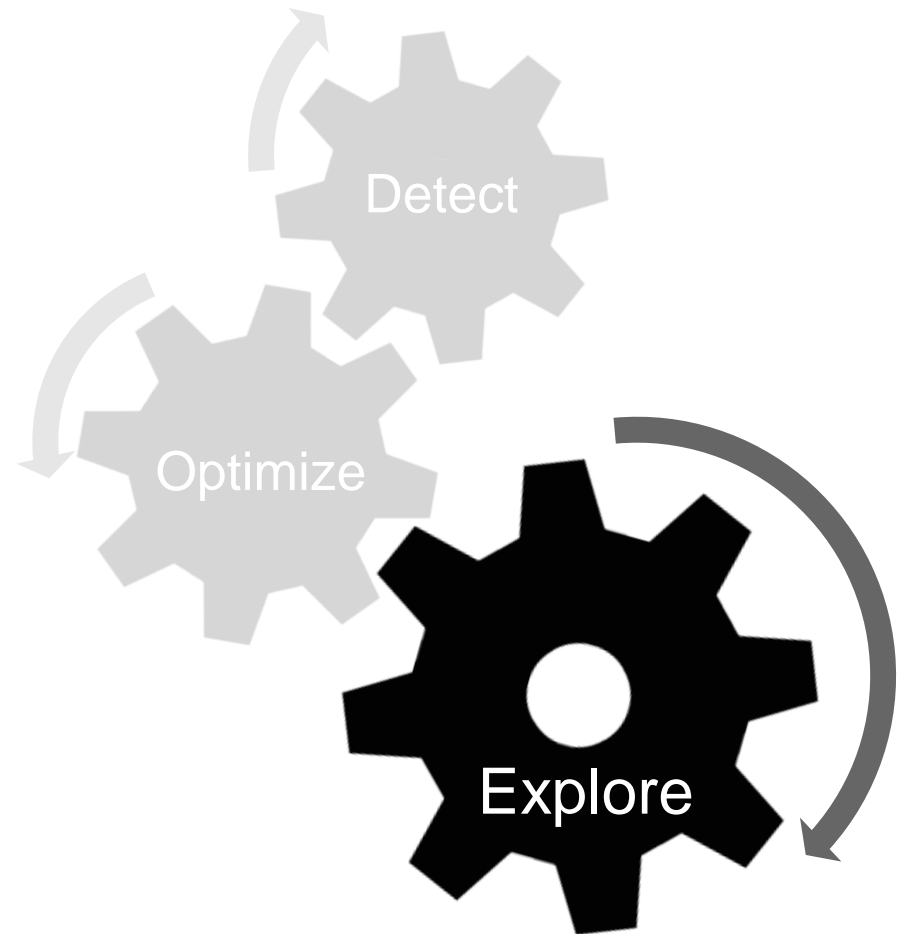
- Functional correctness
- Performance optimization potential

2. Optimize

- Database-oriented programming

3. Explore

- Use SAP HANA-specific features
- Rethink & innovate



ABAP Managed Database Procedures

Introduction of AMDPs

Code-to-Data Paradigm

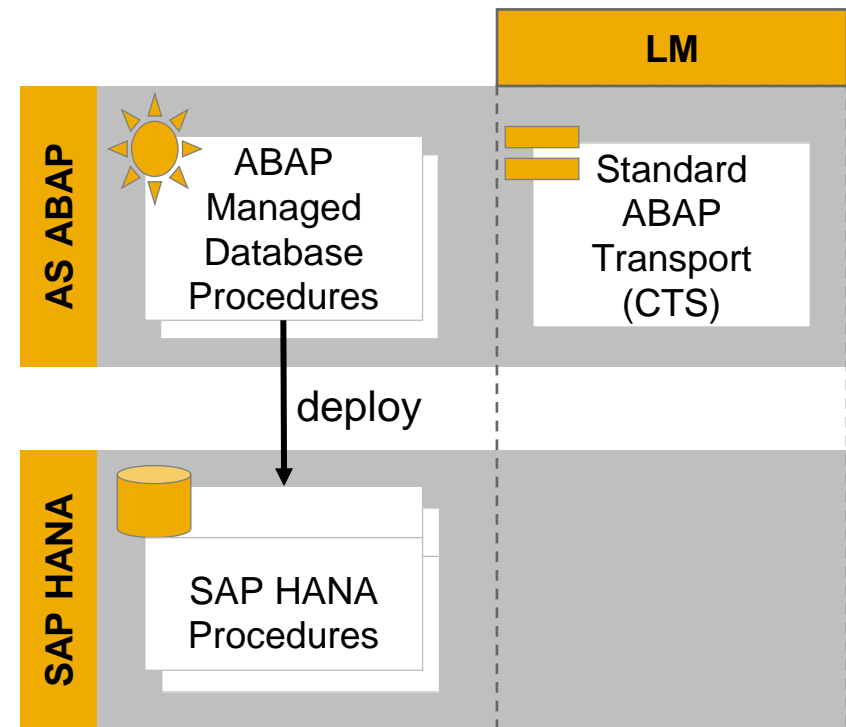
- Supported through embedding native database procedure coding

Definition & Consumption of AMDPs

- Definition / maintenance via ABAP Development Tools in Eclipse
- Standard ABAP class method as containers for database procedures coding
→ Corresponding SAP HANA artifacts created automatically
- Consumption like any other ABAP class method

Fully integrated into the ABAP infrastructure

- Consistent lifecycle management with all other ABAP artifacts (only transport of ABAP objects required)
- Syntax check provided for SQLScript code
- Detailed analysis of ABAP runtime errors



ABAP Managed Database Procedures

ABAP Managed Database Procedures: Demo

ABAP Managed Database Procedures

- Definition
- Implementation
- Consumption
- Runtime Error Analysis

More information

- On AMDPs:
<http://scn.sap.com/docs/DOC-51146>
- On SQLScript:
[SAP HANA SQLScript Reference](#)



ABAP Managed Database Procedures

Class Definition



Additional
Material

Prerequisites in class definition

- Classes with AMDPs must use interface IF_AMDP_MARKER_HDB
- All AMDP method parameters must be passed by value (like RFC)
- All AMDP parameters must be tables with elementary components or scalar types
- AMDPs support (secondary) database connections to the primary database via input parameter **CONNECTION** (type DBCON_NAME)

```
CLASS zcl_amdp_simple_00 DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    ▶ INTERFACES: if_amdp_marker_hdb.

    METHODS get_customer_infos
      IMPORTING
        ▶ VALUE(<input>) TYPE <input_type>
      EXPORTING
        VALUE(<output>) TYPE <output_type>
      RAISING <amdp_exception> .
ENDCLASS.
```

ABAP Managed Database Procedures

Class Implementation



Additional
Material

Extended method implementation syntax: BY DATABASE PROCEDURE ...

- Indicates method body contains database-specific code not executable on the ABAP server
- Database platform (currently only SAP HANA supported)
- Database procedure language (for example SQLScript)
- Used ABAP Dictionary tables, Dictionary views, and other AMDP methods
- Native SAP HANA SQLScript source code

```
METHOD get_customer_infos  
  ► BY DATABASE PROCEDURE  
  ► FOR HDB  
  ► LANGUAGE SQLSCRIPT  
  ► OPTIONS READ-ONLY  
  ► USING <dictionary_artifacts>.  
  
  ► --meaningful SQLScript coding  
  
  et_customer_info =  
    SELECT ... FROM ...;  
  
ENDMETHOD.
```



AMDP consumption like any other ABAP method call

AMDP Runtime:

- At first call of an AMDP, several SAP HANA artifacts are created in the SAP<SID> schema, such as the SAP HANA database procedure
- Artifact creation can alternatively been triggered via ABAP report **RSDBGEN_AMDP**
- When an AMDP is processed, the ABAP stack calls the corresponding database procedure in SAP HANA

```
REPORT zr_amdp_01_simple_call.  
  
DATA(lo_amdp) = NEW zcl_amdp_simple_00( ).  
  
lo_amdp->get_customer_infos(  
    IMPORTING  
        et_customer_info = DATA(lt_result)  
    ).
```


ABAP Managed Database Procedures

Exception Handling

Catchable Exceptions

- Several AMDP runtime errors have a corresponding (catchable) exception
- Naming convention:
 <ERROR_NAME> →
 CX_<ERROR_NAME>
- To-Dos for AMDP
 Developers/Consumers:
 - Add RAISING clause to the AMDP method definition
 - Enclose the AMDP call in a TRY... CATCH block



```
"definition
METHODS <method_name>
    <method_interface>
    RAISING cx_amdp_error.
...

"consumption
TRY.
    <method_call>

    CATCH cx_amdp_execution_failed INTO DATA(lx).
        "do some meaningful error handling
ENDTRY.
```

ABAP Managed Database Procedures

ABAP Managed Database Procedures: Demo

ABAP Managed Database Procedures

- Catchable Exceptions



ABAP Managed Database Procedures

AMDP Exception Hierarchy



Additional
Material

CX_AMDP_ERROR

— CX_AMDP_VERSION_ERROR

— CX_AMDP_VERSION_MISMATCH

Version conflict; database procedure has been changed during program execution

— CX_AMDP_CREATION_ERROR

— CX_AMDP_DBPROC_CREATE_FAILED

Database procedure could not be created because a called database procedure does not exist on the database (any more).

— CX_AMDP_NATIVE_DBCALL_FAILED

SQL error at creation or update of a database procedure before it is called.

— CX_AMDP_WRONG_DBSYS

Database procedure not defined for the current database system.

CX_AMDP_ERROR

— CX_AMDP_EXECUTION_ERROR

— CX_AMDP_EXECUTION_FAILED

Database error during execution of a database procedure.

— CX_AMDP_IMPORT_TABLE_ERROR

Table parameter error during execution of a database procedure.

— CX_AMDP_RESULT_TABLE_ERROR

Error at passing of result table.

CX_AMDP_ERROR

— CX_AMDP_CONNECTION_ERROR

— CX_AMDP_NO_CONNECTION

No database service connection available.

— CX_AMDP_NO_CONNECTION_FOR_CALL

No database connection available for calling a database procedure.

— CX_AMDP_WRONG_CONNECTION

Reserved database connection must not be used for calling a database procedure.

ABAP Managed Database Procedures

Additional Information

Extension Concept

- Extensibility of ABAP Managed Database Procedures supported via AMDP BADls
→ More information on the following slides and on SCN in the ([ABAP for SAP HANA Community](#))

Debugging of AMDPs

- Fully integrated support in the ABAP Development Tools in Eclipse not yet available
- Workaround: External session debugging.
For more information see:
 - SCN document:
<http://scn.sap.com/docs/DOC-51790>
 - Video tutorial:
http://www.youtube.com/watch?v=5OEuyI_s5ME



ABAP Managed Database Procedures

Extensibility: AMDP Business Add-Ins (BADIs)



Additional
Material

Modification-free enhancement of AMDP methods using BADIs and calling points

Special BADIs provided (AMDP BADIs)

- AMDP BAdI interface has to include the tag interfaces
IF_AMDP_MARKER_HDB
- Only AMDP methods allowed
- Fallback method is mandatory

Integrated in the Switch & Enhancement Framework

Filters are currently not supported

Enhancement Spot ZA4H1_AMDP_BADI Change

Enhancement Spot: ZA4H1_AMDP_BADI Inactive

Attributes | Enhancem. Implementations | Technical Details | **Enh. Spot Element Definitions**

BAdI Definitions

- ZA4H1_AMDP_BADI
 - Interface
 - Implementations

BAdI Definition: ZA4H1_AMDP_BADI

Description: BAdI Definition

Interface: ZIF_A4H1_AMDP_BADI

Usability

- ☐ Multiple Use
- ☐ Can only be implemented SAP-internally
- ☐ Limited Filter Use
- ☒ **AMDP BAdI**

Instance Creation Mode

- ☒ Newly Creating Instantiation
- ☐ Reusing Instantiation
- ☐ Context-Specific Instantiation

☒ Call fallback if no implementation is executed

Fallback Class: ZCL_A4H1_AMDP_BADI

Implementation Example Classes

Example Classes

Example Classes	Description
ZCL_A4H1_AMDP_BADI	Class for BAdI: ZA4H1_AMDP_BADI

ABAP Managed Database Procedures

Extensibility: Consumption of an AMDP Business Add-In



Additional
Material

Consumption

- Call of AMDP BAdI like any other AMDP call
 - Inclusion in USING list
 - Consumption via CALL <AMDP_NAME>

```
CLASS zcl_a4h1_call_amdp_badi IMPLEMENTATION.
```

```
  METHOD call_badi
```

```
    by DATABASE PROCEDURE
```

```
    FOR HDB
```

```
    LANGUAGE SQLSCRIPT
```

```
    options read-only
```

```
    using ZA4H1_AMDP_BADI=>get_customer_infos.
```

```
    call "ZA4H1_AMDP_BADI=>GET_CUSTOMER_INFOS"( et_customer_info => et_customer_info );
```

```
  ENDMETHOD.
```

BAdI Caller

Enhancement Spot ZA4H1_AMDP_BADI Display

Implementations		
BAdI Definition		
ZA4H1_AMDP_BADI		
2 implementations found		
Active	Enhancement Implementation	Badi Implementation
	ZCL_A4H1_BADI_IMPL_01	ZCL_A4H1_BADI_IMPL_01
	ZCL_A4H1_BADI_IMPL_02	ZCL_A4H1_BADI_IMPL_02

BAdI Implementation

ABAP Managed Database Procedures

What's Next?

Week 4 Unit 3

Domain-Specific Usage of Core Data Services





Thank you

Contact information:

open@sap.com

open**SAP**

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP AG or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.