



ABAP 7.51(7.52) Releaseabhängige Änderungen

Webinar, 25.1.2019, 10:00 Uhr



Johann Föbleitner

Senior Consultant at Cadaxo

eMail: johann.foessleitner@cadaxo.com

Twitter: [@foessleitnerj](https://twitter.com/foessleitnerj)

Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- Clean Code
- S/4HANA Custom Development



Domī Bigl

Senior Consultant at Cadaxo

eMail: dominik.bigl@cadaxo.com

Twitter: [@DomiBiglSAP](https://twitter.com/DomiBiglSAP)

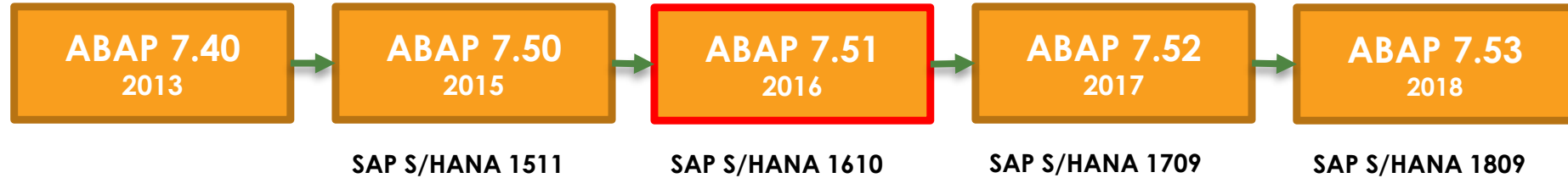
Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- ABAP Units
- SAP Fiori, SAP UI5

• SAP Inside Tracks

- wiki.scn.sap.com
- München, Frankfurt, Berlin, Hannover, Oslo, Madrid, Tokio, Caribbean, ...
- **Wien am 23. November 2019**





ABAP

Umfrage

- Enumerations
- ABAP Open SQL
- ABAP SQL / CDS – Eingebaute Funktionen
- ABAP CDS
- ABAP Development Tools
- ABAP Test Cockpit Checks
- ABAP Channels, ABAP Daemons
- Schnelle Serialisierung für RFC

Enumerations

- Datentype
- Fixe Liste von Werten mit Konstanten
- Zuweisung mit Wertprüfung
 - Default Type Integer
 - 0-based !!!

```
TYPES: BEGIN OF ENUM ty_steps,
    step_0,
    step_1,
    step_2,
    step_3,
END OF ENUM ty_steps.

out->write( step_0 ).      " output: STEP_0
out->write( CONV i( step_0 ) ). " output: 0
```

```
CONSTANTS: BEGIN OF object_state,
    new      TYPE crm_j_status VALUE 'E0001' ##NO_TEXT,
    active   TYPE crm_j_status VALUE 'E0002' ##NO_TEXT,
    closed   TYPE crm_j_status VALUE 'E0003' ##NO_TEXT,
    canceled TYPE crm_j_status VALUE 'E0004' ##NO_TEXT,
END OF object_state.
```

```
TYPES: BEGIN OF ENUM ty_object_state STRUCTURE object_state_2 BASE TYPE crm_j_status,
    undefined VALUE IS INITIAL,
    new        VALUE 'E0001' ##NO_TEXT,
    active     VALUE 'E0002' ##NO_TEXT,
    closed     VALUE 'E0003' ##NO_TEXT,
    canceled   VALUE 'E0004' ##NO_TEXT,
END OF ENUM ty_object_state STRUCTURE object_state_2.
```

```
45 set_state( object_state_2-new ).  
46  
47 set_state( 'E0099' ). The literal "'E0099'" is not type-compatible with the formal parameter "I_STATE". -  
48  
49 set_state( CONV #( 'E0002' ) ).  
50
```

```
TRY.
```

```
    set_state( CONV #( 'E0099' ) ).
```

```
CATCH cx_sy_conversion_no_enum_value.  
ENDTRY.
```

ABAP Open SQL

Common/(General) Table Expressions

- **WITH - CTEs**
 - Ergebnis eines SELECTs als Datenquelle
 - „codeinterner View“
 - Verbesserte Lesbarkeit
 - Performance

```
SELECT main~addrnumber,  
       main~persnumber,  
       m~tel_number AS mobile,  
       p~tel_number AS phone  
FROM   adcp AS main  
LEFT OUTER JOIN adr2 AS m  
    ON  m~addrnumber = main~addrnumber  
    AND m~persnumber = main~persnumber  
    AND m~r3_user    = 3  
  
LEFT OUTER JOIN adr2 AS p  
    ON  p~addrnumber = main~addrnumber  
    AND p~persnumber = main~persnumber  
    AND p~r3_user    = 1  
  
WHERE main~persnumber = @persno  
INTO TABLE @DATA(telnumbers).
```

WITH

```
+mobile_default AS ( SELECT * FROM adr2 WHERE persnumber = @persno AND r3_user = 3 ),  
+phone_default AS ( SELECT * FROM adr2 WHERE persnumber = @persno AND r3_user = 1 )
```

```
SELECT main~addrnumber,  
       main~persnumber,  
       m~tel_number AS mobile,  
       p~tel_number AS phone  
FROM adcp AS main  
LEFT OUTER JOIN +mobile_default AS m  
    ON m~addrnumber = main~addrnumber  
  
LEFT OUTER JOIN +phone_default AS p  
    ON p~addrnumber = main~addrnumber  
  
WHERE main~persnumber = @persno AND  
      ( m~tel_number IS NOT NULL OR  
        p~tel_number IS NOT NULL )  
INTO TABLE @DATA(telnumbers).
```

```
WITH
+phonenumbers AS ( SELECT * FROM adr2 WHERE persnumber = @persno )

SELECT main~addrnumber,
       main~persnumber,
       m~tel_number AS mobile,
       p~tel_number AS phone
FROM   adcp AS main
LEFT OUTER JOIN +phonenumbers AS m
  ON    m~addrnumber = main~addrnumber
  AND  m~r3_user      = 3
LEFT OUTER JOIN +phonenumbers AS p
  ON    p~addrnumber = main~addrnumber
  AND  p~r3_user      = 1
WHERE  main~persnumber = @persno AND
      ( m~tel_number IS NOT NULL OR
        p~tel_number IS NOT NULL )
INTO TABLE @DATA(telnumbers).
```

• ABAP 7.52 – interne Tabelle als Datenquelle

- Daten werden entweder wie gepufferte Tabellen am Application Server behandelt (wenn elementarer Datentyp!) oder die Tabelle wird zum Datenbankserver übertragen
 - Übertragung zum DB Server nicht von allen DBs unterstützt!

```
17
18 * Tabelle bleibt am Applikationsserver
19 DATA lt_partner TYPE TABLE OF bu_partner WITH NON-UNIQUE KEY table_line.
20
21 SELECT partner FROM but000 INTO TABLE @lt_partner.
22
23 SELECT table_line AS partner
24 FROM @lt_partner AS partner
25 INTO TABLE @DATA(result2).
26
27 out->write( result2 ).
```

```
28
29 * Interne Tabelle wird auf die Datenbank übertragen
30 IF NOT cl_abap_dbfeatures=>use_features(
31   EXPORTING
32     requested_features =
33     VALUE #( ( cl_abap_dbfeatures=>itabs_in_from_clause ) ).
34   out->write( 'Sorry, geht leider nicht!' ).
35   RETURN.
36 ENDIF.
37
38 SELECT * FROM but000 INTO TABLE @DATA(lt_but000_2).
39
40 SELECT * FROM @lt_but000_2 AS partner
41 WHERE type = 1
42 ORDER BY partner DESCENDING
43 INTO TABLE @DATA(result).
44
45 out->write( result ).
46
```


ABAP Open SQL

SELECT, DELETE, ...

- **ABAP Open SQL**
 - SELECT ... OFFSET ...
 - SELECT ... CROSS JOIN ...
 - DELETE FROM <table> ...
 - ORDER BY
 - OFFSET
 - UP TO ROWS
 - Eingebaute Funktionen

- **SELECT FROM ... OFFSET** ○

- Es werden aus der Ergebnismenge nur die Zeilen hinter der Zählung zurückgeliefert
- Geht nicht zusammen mit `SELECT SINGLE`, `FOR ALL ENTRIES`, `UNION` und bei Projektions-View
- Verwendung macht nur Sinn, wenn die Reihenfolge nicht undefiniert ist -> `ORDER BY`

• CROSS JOIN

- Ein Cross Join gibt das kartesische Produkt der Tabellenzeilen eines Joins wieder
- Ein neuer Datensatz entsteht durch die Verknüpfung jeder Zeile der einen Tabelle mit jeder Zeile der zweiten Tabelle
- Es kann keine ON Verknüpfung angegeben werden

- **DELETE FROM <table>**
 - Zusätze ORDER BY, OFFSET und UP TO n ROWS können nun verwendet werden

ABAP SQL / CDS

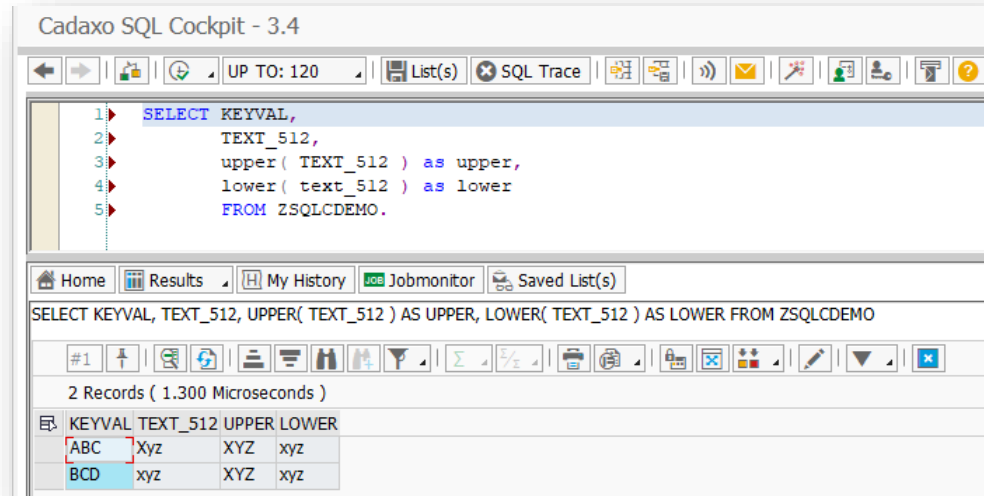
Eingebaute Funktionen

SQL Funktion	Beschreibung	SQL	CDS	Buffer
ABS(arg)	Absolutbetrag von arg.	7.50	7.40	✓
CEIL(arg)	Kleinste ganzzahlige Zahl, die nicht kleiner als der Wert von arg ist.	7.50	7.40	✓
DIV(arg1, arg2)	Ganzzahliger Anteil der Division von arg1 durch arg2.	7.50	7.40	✓
DIVISION(arg1, arg2, dec)	Division von arg1 durch arg2. Das Ergebnis wird auf dec gerundet.	7.51	7.40	
FLOOR(arg)	Größte ganzzahlige Zahl, die nicht größer als der Wert von arg ist.	7.50	7.40	✓
MOD(arg1, arg2)	Positiver oder negativer ganzzahliger Rest der Division von arg1 durch arg2.	7.50	7.40	✓
ROUND(arg, pos)	Gerundeter Wert von arg. Falls pos größer 0 wird zur Position pos gerundet.	7.50	7.40	

SQL Funktion	Beschreibung	SQL	CDS	Buffer
CONCAT(arg1, arg2)	Verkettung der Zeichenketten in arg1 und arg2. Schließende Leerzeichen werden ignoriert.	7.50	7.40	✓
CONCAT_WITH_SPACE(arg1, arg2, spaces)	Verkettung der Zeichenketten in arg1 und arg2 wie mit CONCAT, jedoch getrennt mit spaces Leerzeichen.	7.51	7.50	✓
INSTR(arg, sub)	Position des ersten Vorkommens der Zeichenkette aus sub in arg.	7.51	7.50	
LEFT(arg, len)	Zeichenkette der Länge len mit den len linken Zeichen von arg ohne Berücksichtigung der schließenden Leerzeichen.	7.51	7.50	
LENGTH(arg)	Anzahl der Zeichen in arg ohne Berücksichtigung möglicher schließender Leerzeichen.	7.50	7.50	
LOWER(arg)	Zeichenkette der Länge arg, in der alle Groß- in Kleinbuchstaben konvertiert sind.	7.51	7.51	
LPAD(arg, len, src)	Zeichenkette der Länge len mit dem rechtsbündigen Inhalt von arg ohne schließende Leerzeichen.	7.50	7.50	
LTRIM(arg, char)	Zeichenkette mit dem Inhalt von arg, in dem alle schließenden Leerzeichen und führenden Zeichen entfernt sind, die dem Zeichen in char entsprechen.	7.50	7.50	
REPLACE(arg1, arg2, arg3)	Zeichenkette arg1, in der unter Beachtung der Groß- und Kleinschreibung alle Vorkommen von arg2 durch den Inhalt von arg3 ersetzt sind.	7.50	7.40	
RIGHT(arg, len)	Zeichenkette der Länge len mit den len rechten Zeichen von arg ohne Berücksichtigung der schließenden Leerzeichen.	7.50	7.50	
RPAD(arg, len, src)	Zeichenkette der Länge len mit dem linksbündigen Inhalt von arg ohne schließende Leerzeichen.	7.51	7.50	
RTRIM(arg, char)	Zeichenkette mit dem Inhalt von arg, in dem alle schließenden Leerzeichen und schließenden Zeichen entfernt sind, die dem Zeichen in char entsprechen.	7.50	7.50	
SUBSTRING(arg, pos, len)	Teilfeld von arg ab der Position pos in der Länge len.	7.50	7.50	✓
UPPER(arg)	Zeichenkette der Länge arg, in der alle Klein- in Großbuchstaben konvertiert sind.	7.51	7.51	

• ABAP SQL Funktion: LOWER & UPPER

- LOWER(arg) & UPPER(arg)
- Umwandlung der Zeichenkette arg in Groß bzw. Kleinbuchstaben



SQL Funktion	Beschreibung	SQL	CDS
FLTP_TO_DEC(arg AS dtype)	Konvertierung von arg (Gleitpunktzahl) in eine gepackte Zahl.	-	7.51
BINTOHEX(arg)	Konvertierung von arg (RAW) in eine Zeichenkette (CHAR).	7.52	7.50
HEXTOBIN(arg)	Konvertierung von arg (CHAR) in eine Bytekette (RAW).	7.52	7.50
UNIT_CONVERSION(p1 => a1, ...)	Konvertierung von Mengeneinheiten.	-	7.40
CURRENCY_CONVERSION(p1 => a1, ...)	Konvertierung von Währungen.	-	7.40
DECIMAL_SHIFT(p1 => a1, ...)	Dezimaltrennzeichen setzen.	-	7.40

SQL Funktion	Beschreibung	SQL	CDS	Buffer
COALESCE(arg1, arg2, ...)	Wert des ersten Arguments, das nicht den Null-Wert hat.	✓	7.40	✓

• ABAP CDS Funktion: FLTP_TO_DEC

- fltp_to_dec(arg as dtype)
- Konvertiert eine Gleitpunktzahl in eine gepackte Zahl
 - Über eingebauten Datentyp oder ein Datenelement
 - Mögliche Zieldatentypen: DEC, CURR oder QUAN
 - abap.dec, abap.curr, abap.quan

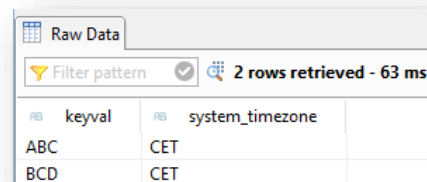
```
6 define view ZSQLCCDSDEMO as select from zsqlcdemo {
7   keyval,
8   float_a,
9   fltp_to_dec( float_a as abap.dec(10,2) ) as fltp_to_dec_dec,
10  fltp_to_dec( float_a as abap.curr(10,2) ) as fltp_to_dec_curr,
11  fltp_to_dec( float_a as abap.quan(10,2) ) as fltp_to_dec_quan,
12  fltp_to_dec( float_a as /cadaxo/sqlcaggrcount ) as fltp_to_dec_dataelement
13 }
```

Raw Data						
Filter pattern 2 rows retrieved - 2 ms						
keyval	float_a	fltp_to_dec_dec	fltp_to_dec_curr	fltp_to_dec_quan	fltp_to_dec_dataelement	
ABC	1.2300000000000000E+02	123.00	123.00	123.00	123	
BCD	1.0000000000000000E+00	1.00	1.00	1.00	1	

SQL Funktion	Beschreibung	SQL	CDS
DATS_IS_VALID	Stellt fest, ob ein Datum gültig ist	7.52	7.50
DATS_DAYS_BETWEEN	Differenz zwischen zwei Datumsangaben	7.52	7.50
DATS_ADD_DAYS	Addiert eine Anzahl von Tagen zu einem Datum	7.52	7.50
DATS_ADD_MONTHS	Addiert eine Anzahl von Monaten zu einem Datum	7.52	7.50
TIMS_IS_VALID	Stellt fest, ob eine Zeitangabe gültig ist	7.53	7.50
TSTMP_IS_VALID	Stellt fest, ob ein Zeitstempel gültig ist (YYYYMMDDHHMMSS)	7.53	7.50
TSTMP_CURRENT_UTCTIMESTAMP	Liefert einen aktuellen Zeitstempel	7.53	7.50
TSTMP_SECONDS_BETWEEN	Berechnet die Differenz von zwei Zeitstempel	7.53	7.50
TSTMP_ADD_SECONDS	Addiert eine Anzahl von Sekunden zu einem Zeitstempel	7.53	7.50
ABAP_SYSTEM_TIMEZONE	Gibt die Systemzeitzone des ABAP Servers zurück	7.53	7.51
ABAP_USER_TIMEZONE	Gibt die Benutzerzeitzone des Users vom ABAP Server zurück	7.53	7.51
TSTMP_TO_DATS	Extrahiert aus einem Zeitstempel das Datum	7.53	7.51
TSTMP_TO_TIMS	Extrahiert aus einem Zeitstempel die Uhrzeit	7.53	7.51
TSTMP_TO_DST	Extrahiert aus einem Zeitstempel die Sommerzeitmarkierung	7.53	7.51
DATS_TIMS_TO_TSTMP	Erzeugt aus einem Datum und Uhrzeit einen Zeitstempel	7.53	7.51

- **ABAP CDS Funktion: ABAP_SYSTEM_TIMEZONE**
 - ABAP_SYSTEM_TIMEZONE(cInt, on_error)
 - Liefert die Systemzeitzone des angegebenen Mandant cInt
 - on_error: 'FAIL', 'NULL' oder 'INITIAL'
 - Systemzeitzone aus TTZCU-TZONESYS

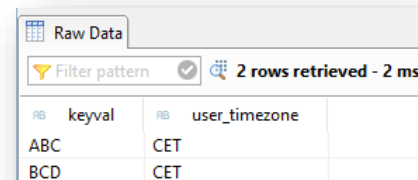
```
1@AbapCatalog.sqlViewName: 'ZSQLCDEMOV'  
2@AbapCatalog.compiler.compareFilter: true  
3@AbapCatalog.preserveKey: true  
4@AccessControl.authorizationCheck: #CHECK  
5@EndUserText.label: 'ZSQLCCDSDEMO'  
6define view ZSQLCCDSDEMO as select from zsqlcdemo {  
7  keyval,  
8  ABAP_SYSTEM_TIMEZONE( $session.client, 'NULL' ) as system_timezone  
9}
```



Raw Data	
Filter pattern <input checked="" type="checkbox"/> 2 rows retrieved - 63 ms	
keyval	system_timezone
ABC	CET
BCD	CET

- **ABAP CDS Funktion: ABAP_USER_TIMEZONE**
 - ABAP_USER_TIMEZONE(user, cInt, on_error)
 - Liefert die Zeitzone des Users user aus Mandant cInt
 - on_error: 'FAIL', 'NULL' oder 'INITIAL'
 - Entspricht SY-ZONLO

```
1 @AbapCatalog.sqlViewName: 'ZSQLCDEMOV'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'ZSQLCCDSDEMO'
6 define view ZSQLCCDSDEMO as select from zsqlcdemo {
7   keyval,
8   ABAP_USER_TIMEZONE( $session.user, $session.client, 'NULL' ) as user_timezone
9 }
10
```



Raw Data

Filter pattern ☒ 2 rows retrieved - 2 ms

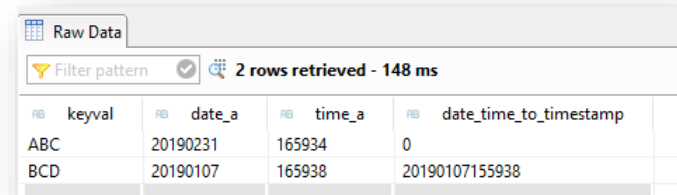
keyval	user_timezone
ABC	CET
BCD	CET

- **ABAP CDS Funktionen: TSTMP_TO_DATS, TSTMP_TO_TIMS, TSTMP_TO_DST**
 - `tstmp_to_...(tstmp, time_zone, client, on_error)`
 - Konvertiert einen Zeitstempel in Datum, Uhrzeit bzw. Sommerzeitflag
 - `on_error`: 'FAIL', 'NULL' oder 'INITIAL'
 - Sommerzeitflag: ,X' = true, space = false

```
1@AbapCatalog.sqlViewName: 'ZSQLCDEMOV'
2@AbapCatalog.compiler.compareFilter: true
3@AbapCatalog.preserveKey: true
4@AccessControl.authorizationCheck: #CHECK
5@EndUserText.label: 'ZSQLCCDSDEMO'
6define view ZSQLCCDSDEMO as select from zsqlcdemo {
7  keyval,
8  timestamp_a,
9  tstmp_to_dats(timestamp_a, 'CET', $session.client, 'INITIAL' ) as timestamp_to_dats,
10 tstmp_to_tims(timestamp_a, 'CET', $session.client, 'INITIAL' ) as timestamp_to_tims,
11 tstmp_to_dst(timestamp_a, 'CET', $session.client, 'INITIAL' ) as timestamp_to_dst
12 }
13
```

Raw Data					
Filter pattern					
2 rows retrieved - 104 ms					
keyval	timestamp_a	timestamp_to_dats	timestamp_to_tims	timestamp_to_dst	
ABC	20190104110021	2019-01-04	120021		
BCD	0	00000000	000000	0	

- **ABAP CDS Funktion: DATS_TIMS_TO_TSTMP**
 - `datst_tims_to_tstmp(date, time, time_zone, client, on_error)`
 - Konvertiert Datum/Uhrzeit in einen Zeitstempel
 - `on_error`: 'FAIL', 'NULL' oder 'INITIAL'

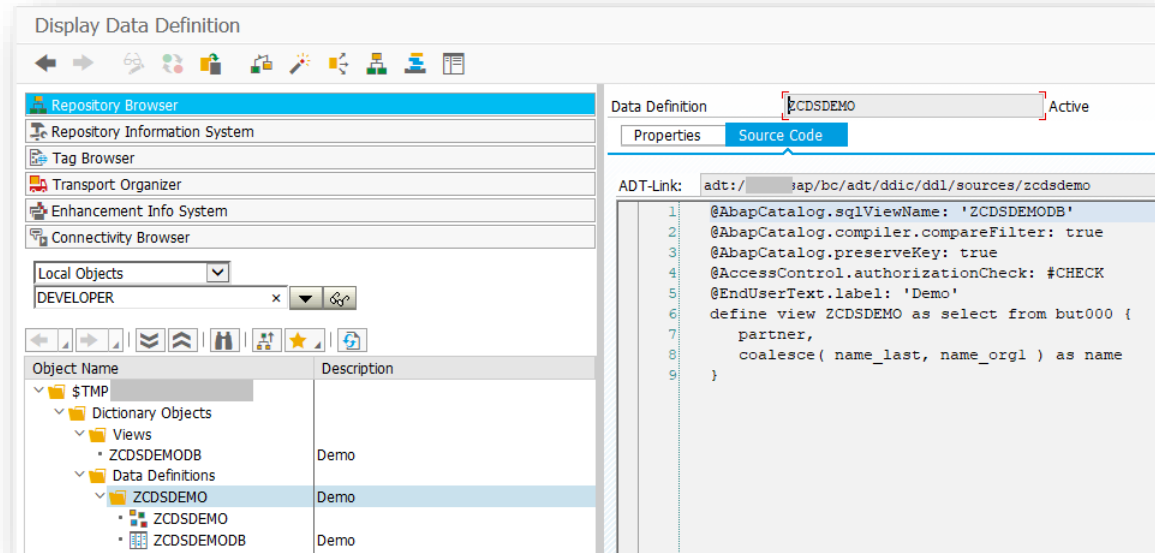


The screenshot shows a 'Raw Data' table with 4 columns: keyval, date_a, time_a, and date_time_to_timestamp. It contains two rows of data. The first row has keyval 'ABC', date_a '20190231', time_a '165934', and date_time_to_timestamp '0'. The second row has keyval 'BCD', date_a '20190107', time_a '165938', and date_time_to_timestamp '20190107155938'. The interface also shows a filter pattern and a status bar indicating '2 rows retrieved - 148 ms'.

keyval	date_a	time_a	date_time_to_timestamp
ABC	20190231	165934	0
BCD	20190107	165938	20190107155938

ABAP CDS

- **Anzeige von CDS DDL in SAP Gui**
 - SE80 neu mit 7.51
 - Report DEMO_SHOW_DDL_SOURCE für ältere Releases
 - SQL Cockpit



- **ABAP CDS 7.51**

- Neue Session Variable `$session.system_date` (sy-datum)
- Erweiterte Möglichkeiten für View Enhancements
- Neue Annotations
- (Einführung von CDS Variants (noch nicht generell freigegeben))

- **ABAP CDS 7.52**

- CDS Annotations können als eigene CDS Objekte definiert werden
- Kardinalität bei `LEFT OUTER JOIN` mit `TO ONE` oder `TO MANY`
- CDS Rollen nun auch für CDS Table Functions möglich
- CDS Extensions sind nun Switch Framework integriert

ABAP Development Tools

- **ADT 2.68 (Backend 7.51 SP00)**
 - ABAP Repository Trees (Nur mit SAP HANA Datenbank)
 - Open Elements aus der Elementinfo
 - CDS Data Definitionen kopieren
 - HTTP Links für CDS Objekte
 - DDL Formatieren
 - CDS Data Definition - Active Annotations, Dependency Analyzer
 - Debugging CDS Table Functions
 - Erstellen von ABAP Push Channel Anwendungen

- **ADT 2.73 (Backend 7.51 SP01)**
 - Source Code Breadcrumbs
 - Anlegen von Domänen in ADT
 - Ausgaben in Console mit `if_oo_adt_classrun`
 - FAQ für SE80 Entwickler ♥

The screenshot shows the SAP ADT IDE interface. On the left, the 'Contents' view displays a hierarchical tree of topics. The 'FAQs for SE80 Experts Using ADT' document is open on the right. The document title is 'FAQs for SE80 Experts Using ADT'. Below the title, it states: 'The following chapters answer frequently asked questions (FAQs) about the ABAP Development Tools (ADT) – that is SAP's Eclipse-based IDE. T in ADT.' The document is organized into sections: 'Scope', 'Contents', 'Example', 'Note', and 'Target Group'. The 'Example' section describes use cases like comparing versions and editing objects. The 'Note' section mentions that the document will be continuously supplemented. The 'Target Group' section lists users who are used to working with the ABAP Workbench or are new to ADT.

Contents

- Adding Code Templates to an ABAP Exception Class
- Working with ABAP Classes Assigned to Multiple Transport Requests
- Launching an ABAP Application (Console)
- Working with ABAP Programs
- Working with ABAP Function Groups and Modules
- Working with Source Code Plug-Ins
- Copying and Duplicating Source Code Based Objects
- Comparing Source Code
- Using Code Folding
- Working with Classic Objects in ABAP Dictionary
- Working with Transformations for XML
- Working with Messages and Message Classes
- Working with ABAP Channels
- Searching in ABAP Projects
- Editing ABAP Source Code
- Applying Quick Assists
- Working with Transport Organizer
- Accessing ABAP Keyword Documentation
- Working with Released APIs
- Working with Data Preview
- Getting Feeds
- Working with Bookmarks
- Ensuring Quality of ABAP Code
- Using Troubleshooting Tools
- Enabling Accessibility Features in ADT
- Reference
- Keyboard Shortcuts for ABAP Development
- List of Development Objects with an Eclipse-based Editor
- Syntax of ABAP Dictionary Objects
- Logpoint Syntax
- Syntax for Breakpoint Conditions
- FAQs for SE80 Experts Using ADT
- ABAP Core Development
- ABAP Dictionary Tools
- Testing and Quality Checking
- Troubleshooting Tools
- Security Guide
- What's New in ABAP Core Development

FAQs for SE80 Experts Using ADT

The following chapters answer frequently asked questions (FAQs) about the ABAP Development Tools (ADT) – that is SAP's Eclipse-based IDE. T in ADT.

Scope

Focus is on SAP GUI-based functionality and its integration within the IDE for easy use and fast transition.

Contents

Each FAQ starts with a header containing the relevant keywords. Below, you will find a table, split into how to information for SE80 and ADT. Each link to the relevant content in the official ADT help documentation to get further details.

Example

The FAQs describe use cases from your daily work, such as:

- Comparing versions and editing objects across several systems
- Arranging the object tree in accordance to your current need
- Working with quick fixes to get support or the relevant code snippets for the appropriate position/activity
- And so on

Note

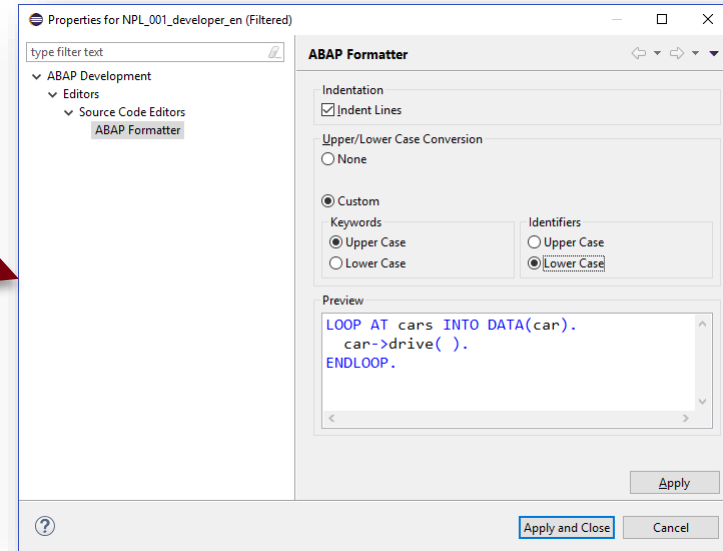
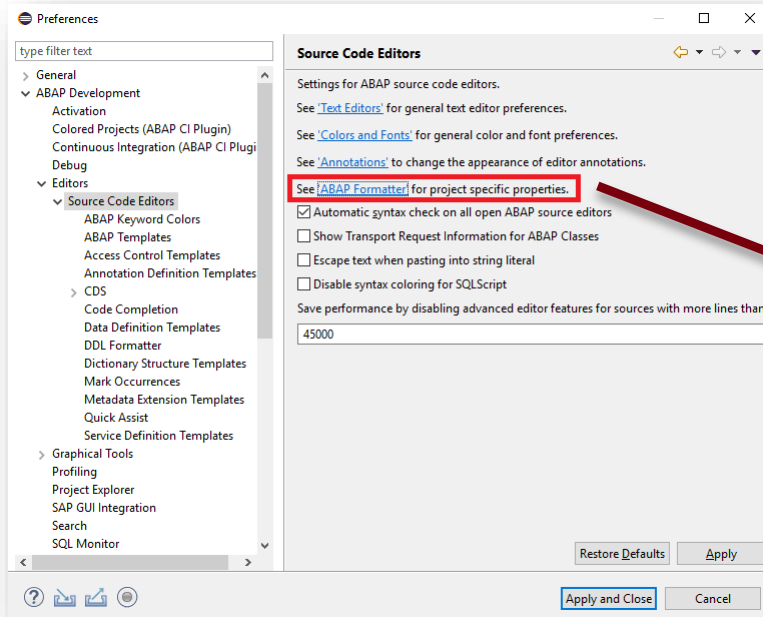
This document will be continuously supplemented in future. If you want to propose an FAQ that needs to be added to this documentation, comment

Target Group

This reference guide is for all ABAP developers who:

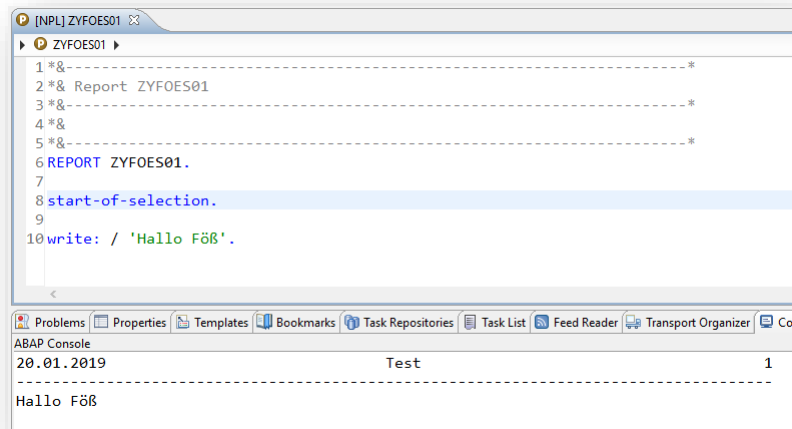
- Are used to work with the ABAP Workbench so far and are now starting to work with ADT
- Already work with ADT and want to find out how a specific feature has been integrated

- **ADT 2.77 (Backend 7.51 SP02) bzw. 2.80 (Backend 7.51 SP03)**
 - Überarbeiteter ABAP Source Code Formatierer
 - Umfangreiche Erweiterungen für ABAP Unit Tests



• ADT 2.83 (Backend 7.52 SP00)

- Ausgaben in Console mit WRITE & CL_DEMO_OUTPUT (für Reports)
- Erstellung und Pflege von ABAP Paketen
- Erstellung und Pflege von Datenbanktabellen
- Erweiterungen des Debuggers

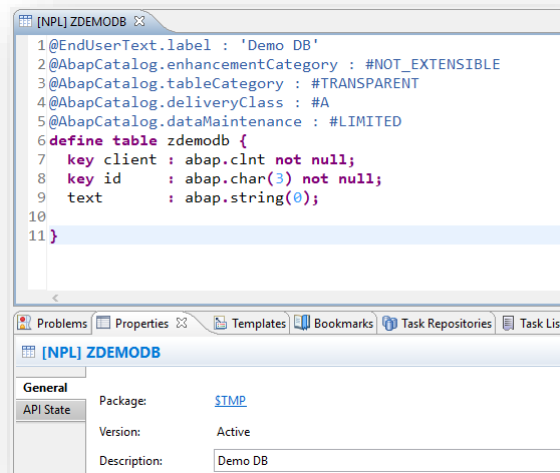


```

1 *&-----*
2 *& Report ZYFOES01
3 *&-----*
4 *&
5 *&-----*
6 REPORT ZYFOES01.
7
8 start-of-selection.
9
10 write: / 'Hallo Föß'.
  
```

ABAP Console
20.01.2019 Test 1

Hallo Föß



```

1 @EndUserText.label : 'Demo DB'
2 @AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #LIMITED
6 define table zdemodb {
7   key client : abap.clnt not null;
8   key id : abap.char(3) not null;
9   text : abap.string(0);
10
11 }
  
```

Problems Properties Templates Bookmarks Task Repositories Task List

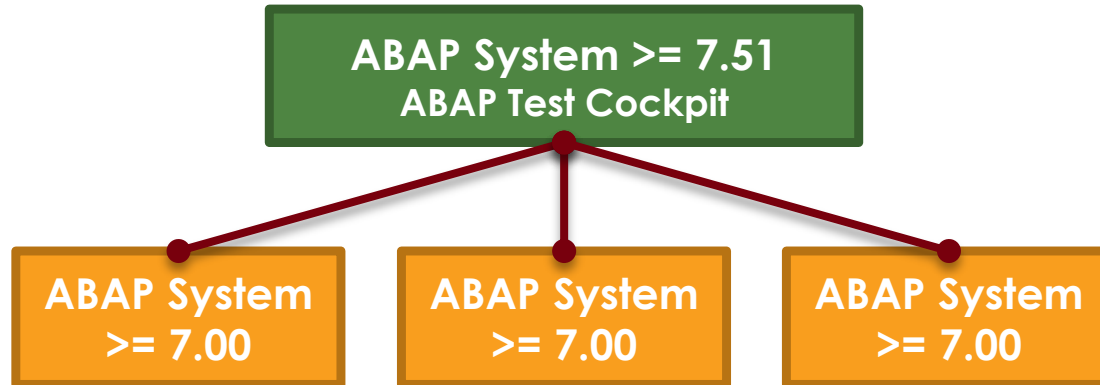
[NPL] ZDEMODB

General	Package:	STMP
API State	Version:	Active
	Description:	Demo DB

ABAP Code Analysen

- **Remote Code Analysen**

- Ein zentrales ABAP Test Cockpit im System
- Zugriff per RFC
- Checkvarianten brauchen nur mehr im zentralen System eingestellt werden



- **Remote Code Analysen - Vorteile**

- Einheitliche Checks innerhalb der Systemlandschaft
- Neueste Checks können vorgenommen werden
 - SAP liefert neue Checks nur mit neuen Releases aus
- Administration nur im zentralen System
- Zentrale Behandlung von Ausnahmen
- Baseline Concept (Legacy Code)

ABAP Channels, ABAP Daemons

- **ABAP Channels, ABAP Daemons**
 - **ABAP 7.51**
 - Large Messages in ABAP Channels
 - Verbessertes Browser Handling für WebSockets
 - Pflege von ABAP Push Channels in ADT
 - **ABAP 7.52**
 - Pflege von ABAP Message Channels in ADT
 - ABAP Daemons

- **ABAP Daemon Framework**

- Permanent verfügbare ABAP Sessions um real-time Events zu Verarbeiten
- Ähnlich wie Windows Services oder Linux cron jobs
- Langläufer – Einmal gestartet, läuft ein Daemon endlos
- Monitoring, Debugging, Laufzeitanalysen, Kernel Statistiken

```
CALL FUNCTION 'ZAPM_WL_DAEMON'
  IN BACKGROUND TASK
  AS SEPARATE UNIT
  EXPORTING
    i_master          = i_master
  EXCEPTIONS
    system_failure    = 1
    communication_failure = 2.
PERFORM set_state USING c_running.
COMMIT WORK.
```

```
DO.
  PERFORM get_state CHANGING l_state.
  IF l_state = c_stopping.
    PERFORM set_state USING c_stopped.
    COMMIT WORK.

    EXIT. "DO - end Daemon
  ENDIF.

  PERFORM do_work.

  PERFORM set_state USING c_running.
  COMMIT WORK.

ENDDO.
```

- **ABAP Daemon Framework**

- CL_ABAP_DAEMON_EXT_BASE
- CL_ABAP_DAEMON_CLIENT_MANAGER

- **Interface IF_T100_DYN_MSG für Exception Klassen (7.50)**
 - Direkte Angabe von Meldungen
 - Verknüpfung von klassischen Ausnahmen mit klassenbasierten Ausnahmen
 - Reduzierung unnötiger Ausnahmentexte
 - Ab. 7.51 - Kurzform MESSAGE <exception_ref>

```
18  
19 RAISE EXCEPTION TYPE zcx_exception  
20 MESSAGE ID '/CADAXO/SQLC'  
21 TYPE 'E'  
22 NUMBER '026'  
23 WITH 'Will Smith'.  
24
```

```
25 CATCH zcx_exception INTO DATA(exception).  
26  
27 * output the message with "MESSAGE"  
28 message exception type exception->if_t100_dyn_msg~msgty.  
29  
30 * new with 7.51 - short message form  
31 message exception.  
32
```


- <https://help.sap.com/viewer/825e9222e7ad4fe1988c6cc600bda779/7.51.3/de-DE/4f20776734874d48abe3aa4234903b8a.html>

[< Previous](#)

RFC: Neue schnelle Serialisierung (neu)

RFC wurde um folgende Funktion erweitert:

Transaktion SM59 (RFC-Destinationen) bietet eine neue Serialisierung für RFC-Kommunikation an. Sie sollten die neue schnelle RFC-Serialisierung wählen, **wenn Sie eine schnellere RFC-Verbindung benötigen**. Dies ist nur möglich, wenn beide Kommunikationspartner den neuen Serialisierungstyp unterstützen.

Die neue schnelle RFC-Serialisierung ist um ein Vielfaches schneller als xRFC/klassisches RFC oder basXML. Außerdem ist sie wesentlich leichter zu verwalten. Die Verwendung der bestehenden Serialisierungen könnte die RFC-Kommunikation erheblich verlangsamen, beispielsweise wenn ein übertragener Parameter um ein Feld erweitert wird. Die neue schnelle Serialisierung hingegen behandelt alle Datentypen gleich und erlaubt Erweiterungen. Außerdem ist sie vollständig kompatibel mit den bereits existierenden RFC-Serialisierungstypen.



a new serialization for RFC communication. **You should choose the new fast RFC serialization, if you need a faster RFC connection.** This is only support the new serialization type.

is faster than xRFC/classic RFC or basXML, and considerably easier to manage. Using the existing serializations, RFC communication may, for

- **ABAP 7.51 – Sonstige News**

- Neue Struktur für Export/Import-Tabellen (Daten Cluster)
- Transformationen – Neue Domänen, Formate, ...
- Volltextsuche in SAP-GUI-Version der ABAP Schlüsselwortdoku
- Statischer ABAP mit eingeschränkter Objektverwendung

- **ABAP 7.52 – Sonstige News**
 - Virtuelle Sortierung von internen Tabellen
 - Implizite Nachrichtenangabe bei `RAISE EXCEPTION MESSAGE`
 - Verweis von ABAP-Doc Kommentaren auf andere Dokumentationen von Repository Objekten
 - Standard-ABAP mit eingeschränkter Objektverwendung

ABAP 7.53

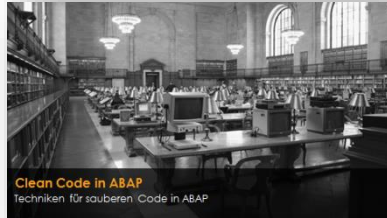
- **Änderungen zu Release 7.53**
 - Pool / Clustertabellen endgültig abgeschafft
 - Open SQL -> ABAP SQL
 - ABAP SQL: Unterstützung neuer Datums/Zeitfunktionen
 - ABAP SQL: Neue Grouping Möglichkeiten
 - Subquery als Datenquelle von MODIFY
 - ATC: Automatisierte Code Anpassungen (Quick Fixes via ADT)
 - Automatische Codenpassung für S/4HANA mit SimDB

- **Blogs**

- [ABAP News for Release 7.51](#) by Horst Keller
- [SAP NetWeaver AS for ABAP 7.52](#) by Olga Dolinskaja

- **Links**

- [Release Notes für ABAP 7.51](#)
- [Release Notes für ABAP 7.52](#)



<http://www.cadaxo.com/blog/>



Modern ABAP

Webinar, 29.3.2019, 10:00 Uhr

<https://www.eventbrite.com/e/modern-abap-tickets-55120171918>

See you again!

Thank you for participating!



<https://twitter.com/foessleitnerj>



<https://www.linkedin.com/in/johann-fößleitner-a9851b2a>



https://www.xing.com/profile/johann_foessleitner



johann.foessleitner@cadaxo.com



<https://twitter.com/domibiglsap>



<https://www.linkedin.com/in/dominik-bigl-9b98b68b>



https://www.xing.com/profile/dominik_bigl



dominik.bigl@cadaxo.com



cadaxo

Cadaxo GmbH
Stubenring 18/5a | 1010 Vienna, Austria

office@cadaxo.com
www.cadaxo.com



<https://www.linkedin.com/company/cadaxo-gmbh>



<https://www.xing.com/companies/cadaxogmbh>



<https://www.facebook.com/CadaxoGmbH>



<https://twitter.com/cadaxo>



<http://com.slideshare.net/cadaxogmbh>



<http://www.youtube.com/CadaxoGmbH>