

June 6th, 2024

abap
conf2024

Closing Keynote

```
DATA(ls_struct) = ...  
ASSERT ls_struct IS NOT NULL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_struct->id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding_id = ls_struct->id.  
  ls_attribute = VALUE #( ( id = <ls_struct->id + <ls_property_binding->id + <ls_struct->id > ) ) .  
  name = <ls_property_binding->alias_text .  
  binding = <ls_property_binding->property_binding .  
  structure_id = ls_struct->id .  
  
  IF <ls_property_binding>-is_key = abap_true .  
    ls_attribute-is_key = if_sadl_type-not_is_key_true .  
  ENDIF .  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes .  
ENDLOOP .
```

It's a wrap!

2 live streams

2 keynotes

3 tracks

18 sessions

28 speakers

1 expo

```
DATA(ls_struct)
ASSERT ls_struct IS NOT NULL
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) )
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_struct-id

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT NULL
  ls_attribute = VALUE #(
    name = <ls_struct-name> |<ls_struct-id> |<ls_struct-id> |<ls_struct-id>
    binding = <ls_struct-binding> |<ls_struct-binding>
    structure_id = ls_struct-id )
  IF <ls_struct-binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

09:00 Keynote: ABAP Cloud – Best practices for a clean core strategy
Boris Gebhardt

Audimax

W1/W2

W3

10:15 Custom Code Adaptation for S/4HANA
Łukasz Pęgiel, English

How can a test pyramid be implemented?
Winfried Schwarzmann, English

CDS - Deep dive on new features and capab.
Andrea Schlotthauer | Konrad Gaerdes, English

11:15 Developing with ABAP Cloud
Thomas Fiedler | Ingo Bräuninger, English

Distributed tracing in ABAP
Petr Plenkov, English

Advanced ABAP - A practical Showcase
Marcos del Perto Garcia | Guruaaj Raman, English

13:00 Steampunk bei der Bahn
Sascha Wächter, German

Experiences with CDS entity views in RAP
Bogdan Brzozowski, English

SAP BTP ABAP Environment Roadmap
Frank Jentsch, English

14:00 Clean Core & ABAP Cloud
Lukas Bretschneider | Lukas Käser, German

Deep Dive into RAP Developer Extensibility
Pavan Kumar Reddy, English

Working with code generators
André Fischer, English

15:00 Keynote ABAP Cloud in action
Latest features and capabilities
Marcel Hermanns | Volker Drees, English

16:15 Dieser Mann schrieb alles auf ...
Enno Wulff, German

Future of ABAP: The Power of Open Source
Marc Bernard, English

Embedded Analytics
Andreas Riehl, English

17:15 KI für ABAP - Opensource Projekt
Jörg Müller, German

ABAP Tools for Clean ABAP
Björn Jüliger | Jörg-Michael Grassau, English

ABAP custom code transform. for Clean Core in S/4
Olga Dolinskaja, English

18:15 Closing Keynote: ABAP Cloud
ABAPConf orga team, English

18:30 Evening party

18:30 Evening party

18:30 Evening party

Thank you to our Sponsors!



June 6 2024

abap
conf2024



ASBRUCON



gambit

SCC



Organized by **cadaxo**

Supported by **SAP**

```
DATA(ls_structure) = get_structure1  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA  
DELETE ls_sadl-attributes USING KEY  
  
LOOP AT it_property_binding ASSIGN  
ls_attribute = VALUE #( ( ls  
name  
binding  
structure  
  
IF <ls_property_binding>-is_key =  
ls_attribute-is_key = if_sadl_t  
ENDIF.  
INSERT ls_attribute INTO TABLS ls  
ENDLOOP.
```

We will be back ...

```
DATA(ls_structure) = get_structured( <ls_getting_get> ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.
```

```
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( <ls_structure-name> ( <ls_structure-id> ) )  
    name = <ls_property_binding-name>_field  
    binding = <ls_property_binding-property_binding>  
    structure_id = ls_structure-id ).
```

```
IF <ls_property_binding>-is_key = abap_true.  
  ls_attribute-is_key = if_sadl_types-not_is_key_true.  
ENDIF.  
INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

We will be back ...

```
DATA(ls_structure) = get_structured( <ls_getting_get> ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( <ls_structure-id> ( <ls_structure-id> ) )  
    name = <ls_property_binding->name <ls_structure-id> )  
    binding = <ls_property_binding->property_binding  
    structure_id = ls_structure-id ).  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

We will be back ...

```
DATA(ls_structure) = get_structured( <ls_getting_get > ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.
```

```
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( <ls_structure-name > |<ls_data> | )  
    name = <ls_property_binding-name_prefix>  
    binding = <ls_property_binding-property_binding>  
    structure_id = ls_structure-id ).
```

```
IF <ls_property_binding>-is_key = abap_true.  
  ls_attribute-is_key = if_sadl_types-not_is_key_true.  
ENDIF.  
INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

We will be back ...

```
DATA(ls_structure) = get_structured( <ls_getting_get> ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure-id WHERE structure-id = ls_structure-id.
```

```
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( ( id = ( |{ls_structure-id}|{it_property_binding-id}| ) ) )  
  name = <ls_property_binding>-name.  
  binding = <ls_property_binding>-property_binding.  
  structure_id = ls_structure-id.
```

```
IF <ls_property_binding>-is_key = abap_true.  
  ls_attribute-is_key = if_sadl_types-not_is_key_true.  
ENDIF.  
INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```


Save the date

December 2024

```
DATA(ls_structure) = get_structure( ... )  
ASSERT ls_structure IS NOT INITIALIZED  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.
```

```
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding_id = it_property_binding-id.  
  ls_attribute = VALUE #( ( id = |{ ls_structure-id }| { ls_property_binding-id } )  
    name = <ls_property_binding-name>_field  
    binding = <ls_property_binding-property_binding>  
    structure_id = ls_structure-id ).
```

```
IF <ls_property_binding>-is_key = abap_true.  
  ls_attribute-is_key = if_sadl_type-not_is_key_true.  
ENDIF.  
INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

LIVE | ONLINE | FREE

```
DATA(ls_structure) = get_structured( <ls_getting_get > ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.  
  ls_attribute = VALUE #( <ls_attribute> = [ <ls_structure>= <ls_structure-id> | <ls_property_binding>= <ls_property_binding> | <ls_data>= <ls_data> ].  
  name = <ls_property_binding>-name.  
  binding = <ls_property_binding>-binding.  
  structure_id = ls_structure-id.  
  IF <ls_property_binding>-is_key = <ls_is_key>.  
    ls_attribute-is_key = if_sadl_type-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

Be a

Speaker

Keynote Speaker

Sponsor

```
DATA(ls_structure) = get_structured( ls_getting_get );
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING ls_property_binding.
  ls_attribute = VALUE #(
    name = <ls_property_binding-name>
    binding = <ls_property_binding-property_binding>
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

Innovate

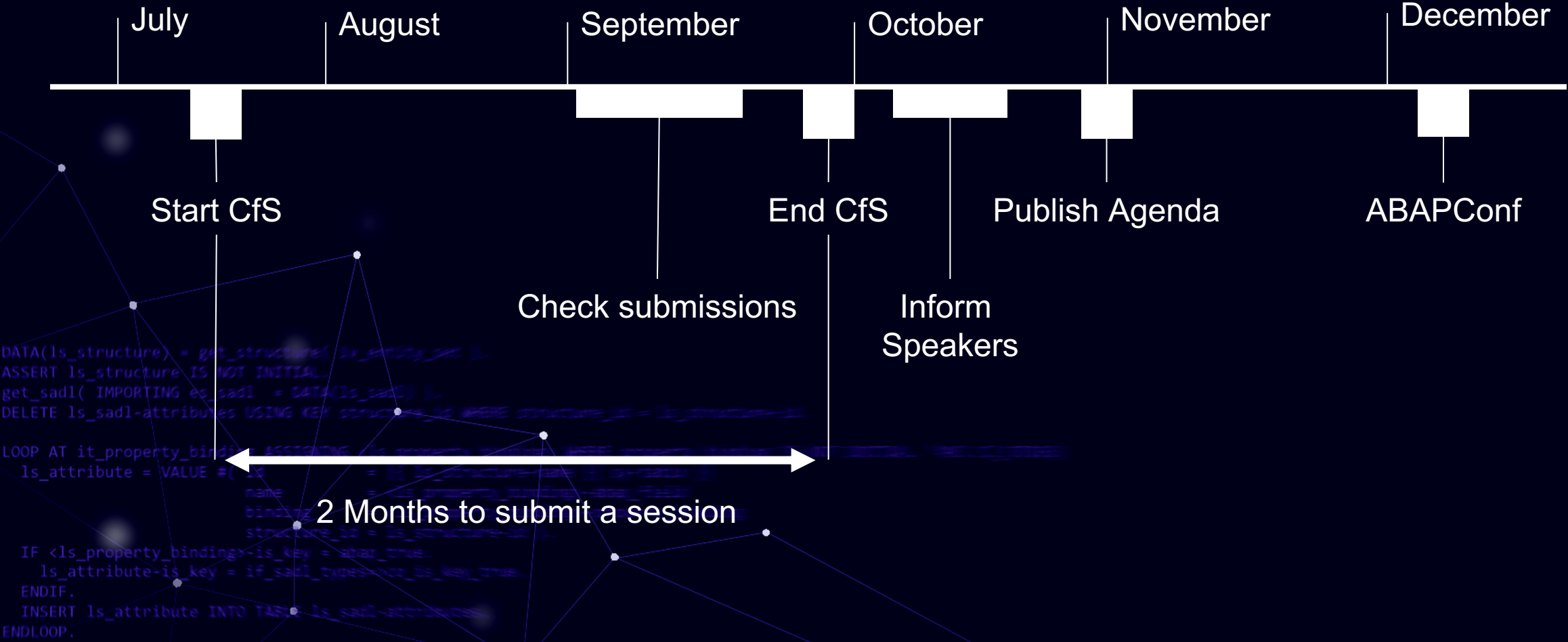
Open Source Track?

Cool Tools Track?

```
DATA(ls_structure) = get_structured( ls_getting_get );
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #(
    name = <ls_property_binding->name |>
    binding = <ls_property_binding->binding |>
    structure_id = ls_structure-id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_types-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

Get ready to submit your session

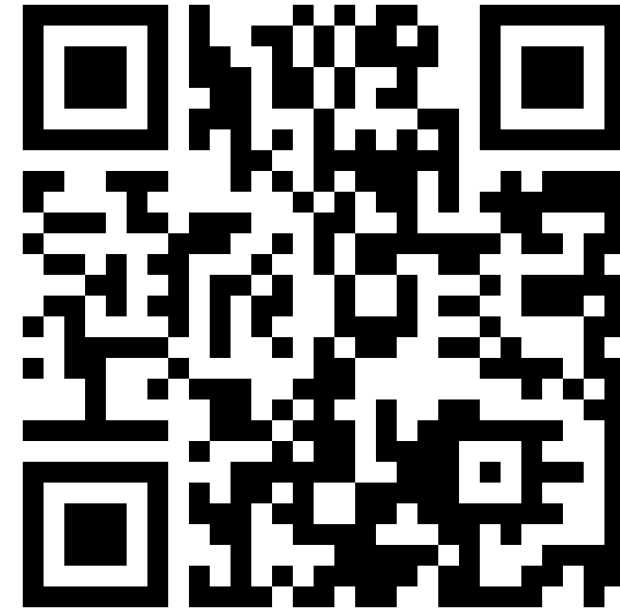


Stay in the loop

Join our LinkedIn group

```
DATA(ls_structure) = get_structured_ls_getting_get().
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure_id.

LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_binding_id = ls_structure-property_binding_id.
  ls_attribute = VALUE #( ( name = <ls_property_binding->name.
                           binding = <ls_property_binding->binding.
                           structure_id = ls_structure_id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```



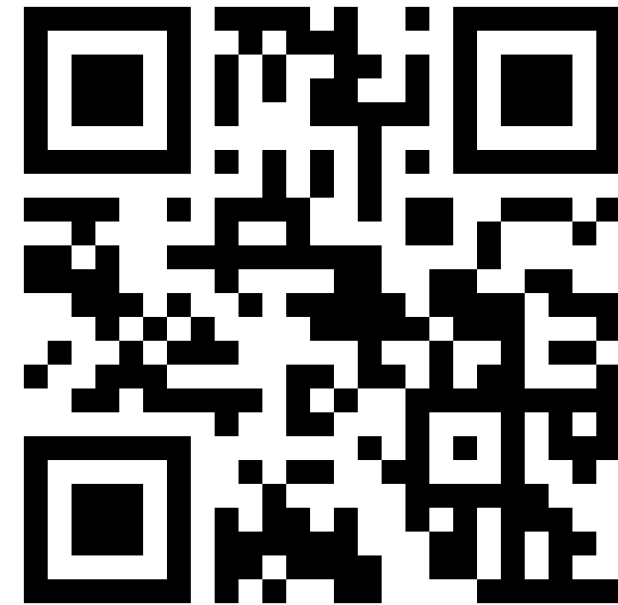
<https://tinyurl.com/ABAPConfLI>

Learn more ...

cadaxo
Webinare zu
ABAP (auf DE)

```
DATA(l_s_attr) TYPE TABLE OF ls_attr.  
ASSERT( s_attr IS NOT INITIAL ).  
get_sadl( IMPORTING es_sadl = s_attr EXPORTING ls_attr = l_s_attr ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure_id.  
  
LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding_id = ls_structure_id.  
  ls_attribute = VALUE #( ( ls_structure_id = ls_structure_id ) ).  
  name = <ls_property_binding-name>_field.  
  binding = <ls_property_binding-property_binding_id>_binding.  
  structure_id = ls_structure_id.  
  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not-is-key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

cadaxo



<https://www.cadaxo.com/webinare/>

Learn more ...

Brandeis Cheat Sheets

```
DATA(ls_structure) = get_structured_ls_getting_get()
ASSERT ls_structure IS NOT INITIAL
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) )
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure_id

LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding_id = ls_structure_id
  ls_attribute = VALUE #( ( name = <ls_property_binding-name>
                          binding = <ls_property_binding-property_binding_id>
                          structure_id = ls_structure_id ) )
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

 Brandeis Consulting



<https://www.brandeis.de/blog/cheat-sheets>

Werde Mitglied

ABAPConf, der Verein

```
DATA(ls_structure) = get_structured( ls_getting_get ).  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).  
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure-  
  
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE property_status  
  ls_attribute = VALUE #( ( name = <ls_structure-name> |<ls_property_binding-name> |<ls_property_binding-property_binding> |<ls_structure-id> ).  
  IF <ls_property_binding>-is_key = abap_true.  
    ls_attribute-is_key = if_sadl_types-not_is_key_true.  
  ENDIF.  
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.  
ENDLOOP.
```

Mehr Infos
dazu
demnächst

Welcome to the ABAPConf Legends club

Sascha Wächter

Łukasz Pełgiel

Thomas Fiedler

Ingo Bräuninger

Lukas Bretschneider

Lukas Käser

Enno Wulff

Winfried Schwarzmann

Petr Plenkov

Bogdan Brozowski

Pavan Kumar Reddy

Marc Bernard

Björn Jülliger

Jörg-Michael Grassau

Andrea Schlotthauer

Marcos del Perto Garcia

Jörg Müller

Konrad Gaerdes

Volker Drees

Frank Jentsch

André Fischer

Boris Gebhardt

Andreas Riehl

Olga Dolinskaja

```
DATA: ls_structured IS TYPE s_structured.
ASSIGN INITIAL VALUE TO ls_structured.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl) ).
LOOP AT it_property_binding ASSIGNING <ls_property_binding> WHERE <ls_property_binding>
  ls_attribute = VALUE #( ( name = <ls_property_binding>-name
                           binding = <ls_property_binding>-binding
                           structure_id = <ls_property_binding>-structure_id ).
  IF <ls_property_binding>-is_key = abap_true.
    ls_attribute-is_key = if_sadl_type-not_is_key_true.
  ENDIF.
  INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

Thank you to our Sponsors!



June 6 2024

abap
conf2024



ASBRUCON



gambit

SCC



Organized by **cadaxo**

Supported by **SAP**

```
DATA(ls_structure) = get_structure1  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA  
DELETE ls_sadl-attributes USING KEY  
  
LOOP AT it_property_binding ASSIGN  
ls_attribute = VALUE #( ( ls  
name  
binding  
structure  
  
IF <ls_property_binding>-is_key =  
ls_attribute-is_key = if_sadl_t  
ENDIF.  
INSERT ls_attribute INTO TABLS ls  
ENDLOOP.
```

Evening celebration



Food

Beer



Drinks

Music



```
DATA(ls_structure) = get_sadl(IMPORTING es_sadl = DATA(ls_sadl-attributes))
ASSERT ls_structure IS NOT INITIAL.
get_sadl( IMPORTING es_sadl = DATA(ls_sadl-attributes))
DELETE ls_sadl-attributes USING KEY structure_id WHERE structure_id = ls_structure_id.
```

```
LOOP AT it_property_binding ASSIGNING ls_property_binding WHERE property_binding IS NOT INITIAL.
  ls_attribute = VALUE #( ( name = ls_property_binding->name
                          binding = ls_property_binding->binding
                          structure_id = ls_structure_id ). )
```

```
IF <ls_property_binding>-is_key = abap_true.
  ls_attribute-is_key = if_sadl_types-not_is_key_true.
ENDIF.
INSERT ls_attribute INTO TABLE ls_sadl-attributes.
ENDLOOP.
```

Thank you to our Sponsors!



June 6 2024

abap
conf2024



ASBRUCON



gambit

SCC



Organized by **cadaxo**

Supported by **SAP**

```
DATA(ls_structure) = get_structure1  
ASSERT ls_structure IS NOT INITIAL.  
get_sadl( IMPORTING es_sadl = DATA  
DELETE ls_sadl-attributes USING KEY  
  
LOOP AT it_property_binding ASSIGN  
ls_attribute = VALUE #( ( ls  
name  
binding  
structure  
  
IF <ls_property_binding>-is_key =  
ls_attribute-is_key = if_sadl_t  
ENDIF.  
INSERT ls_attribute INTO TABLS ls  
ENDLOOP.
```