

SAP Community > Products and Technology > Technology > Technology Blogs by SAP
> CDS - One Concept, Two Flavors

CDS - One Concept, Two Flavors



 horst_keller
Product and Topic Expert

07-20-2015 10:40 AM



115 Kudos

If you search the web or SCN for CDS, SAP's Core Data Services, you can easily find statements like "*Core data services (CDS) is an infrastructure for defining and consuming semantically rich data mode....*". On the other hand, there also seems to be something called ABAP CDS in ABAP Dictionary. How are they connected? Let's have a look from the ABAP (and even ABAP CDS) docu writer's worm's eye view.

ABAP Dictionary

In order to develop (business) applications, you need something to model your data. In the ABAP world, meaning on the ABAP application server, for this purpose you used the ABAP Dictionary or tools like the data modeler for years. The ABAP Dictionary was and is the platform independent metadata repository for database tables and database views that can be accessed in ABAP using Open SQL. The definition of the database entities is done in the form based ABAP Dictionary tool on the application server and the according DDL is passed to the database via the DBI (Database Interface). You can examine that by looking at the "Database Object" in SE11. E.g., for a view, you see the according DDL statement CREATE VIEW. For an ABAP program, the entities defined in the ABAP Dictionary act like global types, making it simple to declare ABAP data objects as targets or sources for DML statements in Open SQL.

HANA CDS

With the dawn of SAP HANA and the possibility to develop applications directly on the database, not using an application server any more, the need arised, to create a meta model repository directly on the database. As with the ABAP Dictionary on the application server, there should be more in the box for you than using native SQL's CREATE TABLE or CREATE VIEW. Especially regarding the need of enriching pure technical definitions with semantics. And that's one of the main reasons for SAP's Core Data Services.

Core Data Services provide a specification for an SQL-based DDL that is enriched with further possibilities like annotations or associations that generally can be implemented on different platforms. On SAP HANA, CDS provides the possibility "to define the artifacts that make up the data-persistence model." The DDL of CDS allows you to define database tables, database views, data types by wrapping the according native HANA SQL statements and enriching them with semantical properties. From an ABAP programmer's point of view, one might say: Well, OK, there's a source code based dictionary tool directly on the HANA database. But that would be too shirt-sighted.

ABAP CDS

From the beginning, CDS was not designed for HANA alone. Also the ABAP application server should be able to benefit from the enhanced capabilities that are offered by the DDL of CDS compared to the form based ABAP Dictionary tool. Since the ABAP Dictionary with its capability of defining tables, views and data types was already there, the natural way of introducing CDS on the ABAP application server was to add it to the ABAP Dictionary. An ADT based source code editor allows you to create DDL sources. On activation, the CDS entities defined in such a DDL source become full citizen ABAP Dictionary objects. They work as ABAP types that can be named after a TYPE addition and they can be accessed in Open SQL. As a first step, the advanced view building capabilities of CDS have been implemented in ABAP CDS. With many, many tables already defined in the ABAP Dictionary, the DEFINE VIEW statement of ABAP CDS makes the full wealth of the CDS universe readily available to existing ABAP data models. You can either define sophisticated new views or simply wrap an existing table in a CDS view in order to enrich it semantically. For example, ABAP CDS offers a new authorization concept based on roles defined with DEFINE ROLE (partly released with 7.40, SP10 and fully released with ABAP 7.50) in a DCL source code. In order to let an existing table participate in the new authorization concept, you can simply create a CDS view for that table that is connected to a role. Other examples are advanced joining of tables with associations or using specific annotations in order to connect existing tables to new technologies like OData or UI5. Even in ABAP itself, CDS views are handled in an advanced way compared to classical dictionary

objects - automatic client handling is an example. So, with introducing CDS views in the ABAP Dictionary one big step is already taken. Capabilities to create database tables, database functions (available with ABAP 7.50), and data types in ABAP CDS might follow. This would ultimately allow you to create data models in the ABAP Dictionary from scratch using ABAP CDS only.

ABAP CDS vs. HANA CDS

With HANA CDS and ABAP CDS we have two implementations based on the same specification. The implementations are very similar but not 100 % equal. If you know the DDL of CDS, you should be able to understand definitions of CDS entities in both flavors. But as a rule you will not be able to copy DDL sources from HANA to ABAP and vice versa without modifications. The implementation of HANA CDS and ABAP CDS is ongoing and with different priorities. That's why a blog like [New Core Data Services Features in SAP HANA 1.0 SPS 10](#) is mainly interesting for developers working directly on SAP HANA. ABAP developers have to look for blogs like [news in ABAP CDS](#).

While HANA CDS has to function on SAP HANA only, ABAP CDS is open. Consequently there are some restrictions for ABAP CDS that do not exist for HANA CDS (in the same way as Open SQL is more restricted than Native SQL). A good example are built-in functions. A built-in function like [CURRENCY_CONVERSION](#) offered in ABAP CDS must be available on any database platform and - very important - must behave in the same way on all platforms. Same for expressions like arithmetic expressions, aggregates or the CAST expression. Before releasing such a functionality in ABAP CDS, all platforms have to participate. Quiet a task! And that's why ABAP CDS cannot offer all [SQL Functions](#) of SAP HANA yet (but [working on it](#)...). On the other hand, it is open! And there are also ABAP specialties like client handling or table buffering that are not available in HANA CDS, but supported in ABAP CDS.

Conclusion

SAP's Core Data Services provide a single concept for an infrastructure for data modelling that is implemented in SAP HANA as well as on the ABAP application server. The design principles are the same for both implementations but due to differences in the respective environments, there are naturally some differences between these flavors.

Note

To my knowledge, there is no native implementation of the CDS concept on other database platforms than on SAP HANA up to now. In order to model in CDS for databases other than HANA, you can of course use ABAP CDS in the ABAP Dictionary, and let it push down the model for you.

Tags:

abap

CDS

sap hana

Comments

abdul_hakim Active Contributor

07-20-2015

4:37 PM

Hi Horst, Another great blog from you. Thanks for sharing it with the community.

Thanks

Hakim



Florian Active Contributor

07-20-2015

7:32 PM

Hi Horst,

well written and thanks for pointing out the difference again.

~Florian



UweFetzer_se38 Active Contributor

07-22-2015

7:00 PM

Define role: in the documentation we still can read "**This functionality is not yet released and must not be used**"

Is the CDS authorisation concept really officially released in SP10?



 horst_keller Product and Topic Expert

07-23-2015

7:13 AM

The DCL syntax DEFINE ROLE was created and delivered with 7.40, SP08. It allows you to create a role for CDS entity (view). In a first go, only mapping roles are supported (annotation MappingRole must be true), that binds the CDS entity to classical authorization objects.

But defining roles is not enough. You also need a framework that handles the role. The idea is, that the ABAP programmer does not have to use AUTHORITY-CHECK any more, but that he uses a framework that implicitly checks the authorization.