

# Cost Center Hierarchies

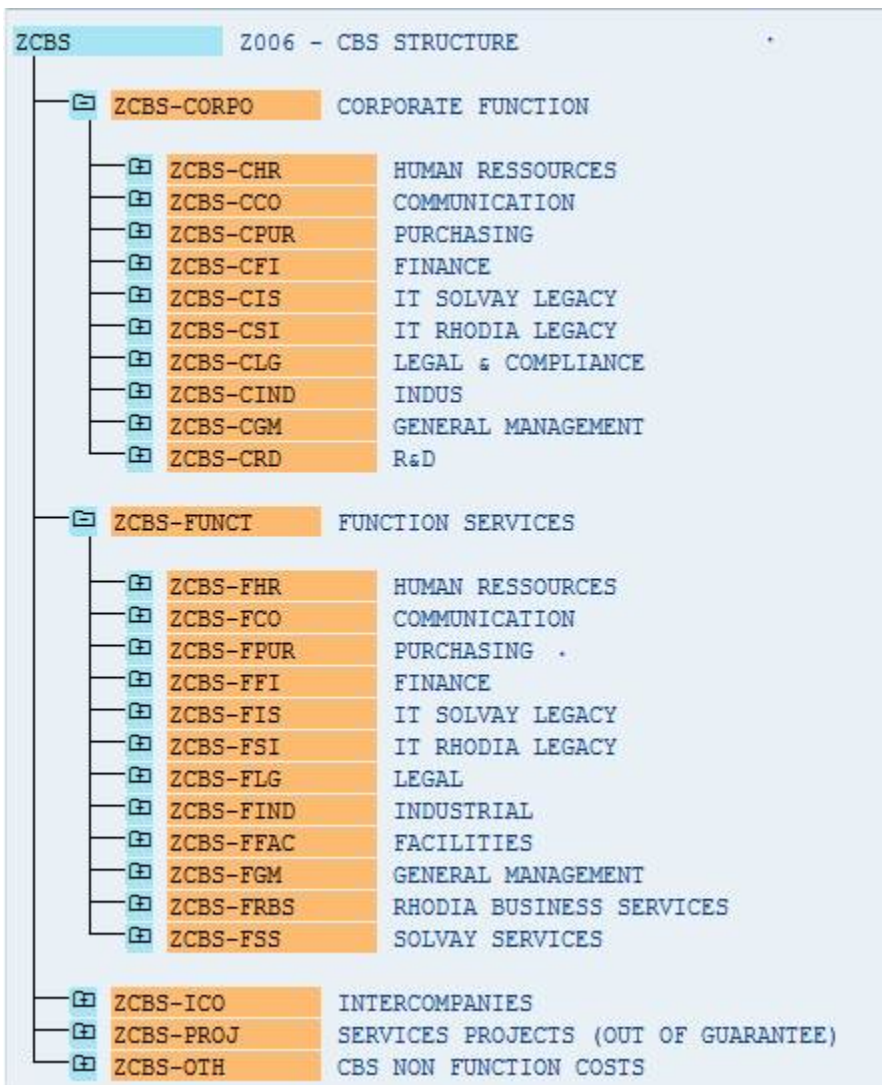
- 1 [Functional informations](#)
- 2 [Actual loading of hierarchy](#)
- 3 [Data Loading](#)
  - 3.1 [Info Providers and objects loaded](#)
- 4 [Data Quality Control](#)
- 5 [Operational Documentation](#)
  - 5.1 [Procedures](#)
  - 5.2 [Scheduling](#)
  - 5.3 [Monitoring](#)
  - 5.4 [Error Handling](#)
  - 5.5 [Known Bugs](#)
  - 5.6 [Roadmap](#)

## Functional informations

Before CBS project, the hierarchies used in BW for Funct0 were the FUNCT0 hierarchies.

Now, we have new hierarchies called ZCBS, existing in the different ERPs.  
The name of the hierarchies is *Controlling area + 'ZCBS'*

Example below for Z006CBS:



# Actual loading of hierarchy

Today, all ZCBS hierarchies are created and managed in the ERP. We don't use C\_FUNCTCC anymore.

To load the attributes, we have one transformation (for RCS controlling areas) **Transformation: 0COSTCENTER\_ATTR -> C\_COSTCTR.**

The start routine in the transformation rule will search for the controlling area-cost center the different level

The start routines start to load the hierarchy of the controlling area into an internal table.

Then we search the cost center in this hierarchy in order to find the lowest level of the hierarchy.

With this level, we can find the levels above until the highest (level 0), which is ZCBS.

Example for Cost center 7666-2025.

We find the hierarchy's code in the hierarchy's table RSHIEDIR.

Then we load into an internal table from the hierarchy cost center's table for this cost center

HIEID	OBJVERS	NODEID	NODENAME	TLEVEL	PARENTID	CHILDID	NEXTID	
<input type="checkbox"/>	125VOOSGSOPOKZ1XJASTBUI50	A	00001329	Z0067666-2025	06	00001319	00000000	00001330

We find the level 6 and we obtain a node's code (PARENTID field: 1319)

Then we search this PARENTID value in the same table but in the NODEID field.

HIEID	OBJVERS	NODEID	NODENAME	TLEVEL	PARENTID	CHILDID	NEXTID	
<input type="checkbox"/>	125VOOSGSOPOKZ1XJASTBUI50	A	00001319	Z006EPRD2X	05	00001318	00001320	00000000

We got now the level 5, the lowest level of our hierarchy (in BW it's the level 4) and also a new PARENTID.

The node name (minus the 4 first characters) is the code of the level.

We search the new PARENTID in order to find the level above and we stop when there's no PARENTID value.

It means we are at the highest level, ZCBS

HIEID	OBJVERS	NODEID	NODENAME	TLEVEL	PARENTID	CHILDID	NEXTID	
<input type="checkbox"/>	125VOOSGSOPOKZ1XJASTBUI50	A	00000001	Z006ZCBS	01	00000000	00000002	00000000

The texts of the levels are managed in a dedicated hierarchy, the Z013ZCBS.

## Data Loading

### Info Providers and objects loaded

Process chain	Description	Start time (CET)	Frequency	Duration	Comments
---------------	-------------	------------------	-----------	----------	----------

FREQUENTLY_LOAD_MD	02 Frequently Load Master Data	6 am 12 pm 6 pm	Daily (not the week-end)	30 min	Load attributes from hierarchy
RSP_HIER_DAILY	Daily Hierarchies process chain	7 pm	Daily (not the week-end)	35 min	Load hierarchies with sub chains CO_DAILY_MASTER_DATA & CO_DAILY_MASTER_DATA_HY + attributes from hierarchy
ZPC_MD_ATTR_RCS	Daily Master Data ATTR - RCS	12 am	Daily	1h 15 min	Load attributes from hierarchy

## Data Quality Control

Data come from SAP system. To compare data between BW and sources systems, check propagation layers.

## Operational Documentation

### Procedures

<Describe the recurring procedures needed to operate the application (eg. start/pause/terminate/restart the app processes, data preparation, data ingestion, ETL, data visualization, data export, other manual activities)>

### Scheduling

<Describe the scheduling in place for the application (eg. existing jobs, trigger time/event based, dependencies)>

### Monitoring

<Describe the monitoring checks to confirm the application is performing well (eg. check the overall status, check performance metrics like runtime/data volume/memory/disk/CPU, maintain and react to alerts/notifications)>

### Error Handling

<Describe how to handle errors (eg. error codes, description and respective resolution, alert users)>

### Known Bugs

<List the existing bugs, its criticity, workarounds and resolution plan.>

### Roadmap