



# Customer Data Control

- verwalten und steuern -

# Customer Data Control

- Für jede kleine Steuertabelle eine Datenbanktabelle anzulegen ist teuer!
- Jeder Entwickler programmiert den Zugriff auf Z-Tabellen anders!
- Die Pflegedialoge der Z-Tabellen haben jeweils einen anderen Aufbau!

## Besser ist es, wenn ...

- die Pflege im Dialog einheitlich und komfortabel ist
  - Pflege von Schlüsseln mit Wildcards: Customizing
  - Zugriff mit Wildcards: Matchcode
- die Zugriffe im Programm standardisiert sind
- die Datenbank nur eine Tabelle verwalten muss

# Vorgehen

Um die Steuerung über das CDC nutzen zu können, sind folgende Schritte notwendig:

1. **Definition der DDIC-Strukturen für Schlüssel- und Datenbereich.**
2. **Einbinden des Funktionsbausteins für das Lesen und Bereitstellen der Daten im Anwendungsprogramm und Auswertung der zurück gelieferten Werte.**
3. **Pflege der Daten im CDC:**
  1. Einbinden der DDIC-Strukturen und
  2. erfassen eines beschreibenden Kurztextes.
  3. Steuern des Programms durch Daten zum Schlüssel

# Dialog

Zur ausgewählten Struktur werden die Kopfdaten angezeigt und die Positionsdaten in die Tabelle eingelesen.

KeySegment(Typ): /ALOGIS/KEY\_CSDATA ▼ CD-Key: Teilbereich für Beisp...  
 DatenSegment(Typ): /ALOGIS/DAT\_CSDATA ▼ CD-Data: Teilbereich für Beis...

Bezeichnung:

am:

Letzte Änderung von:  um:

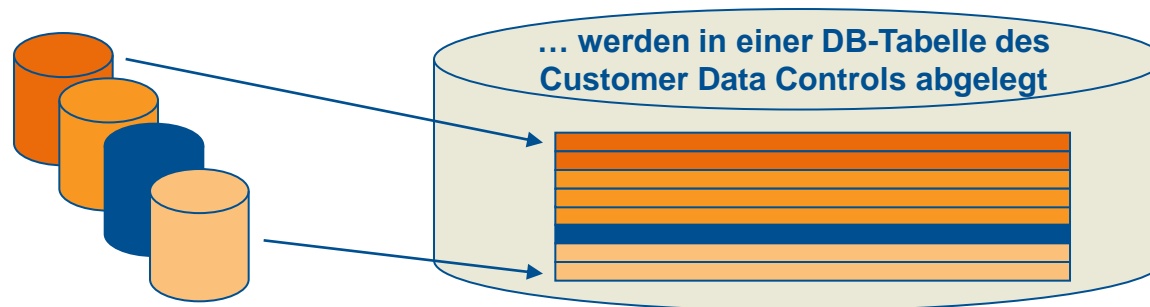
Der Schlüsselbereich kann Wildcards enthalten. Dementsprechend können im Programm bei einem Aufruf auch mehrere Tabelleneinträge zurückgegeben werden.

Der Datenbereich enthält entsprechend der Datenstruktur, die Werte der einzelnen Felder. Diese stehen im Programm zur Auswertung zur Verfügung.

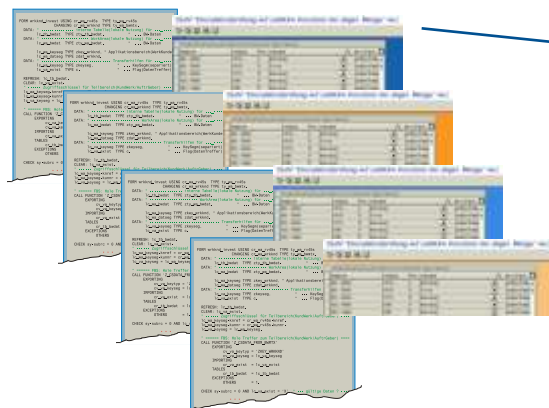
Werk	MArt	NK	A...	ALg	Bearbeiter	am	um
*	*				KÜHNE	28.06.2004	15:53:14
++01	*	Z0			KÜHNE	28.06.2004	15:52:13
0001	*	Z0			KÜHNE	28.06.2004	15:51:44
0001	FERT	Z2	X		KÜHNE	28.06.2004	15:51:24
0001	HALB	Z1	X	X	KÜHNE	28.06.2004	15:51:00

# Vereinheitlichung

„Z-Tabellen“ verschiedener Erweiterungen...



Unterschiedlich implementierte Zugriffe und Pflegedialoge...

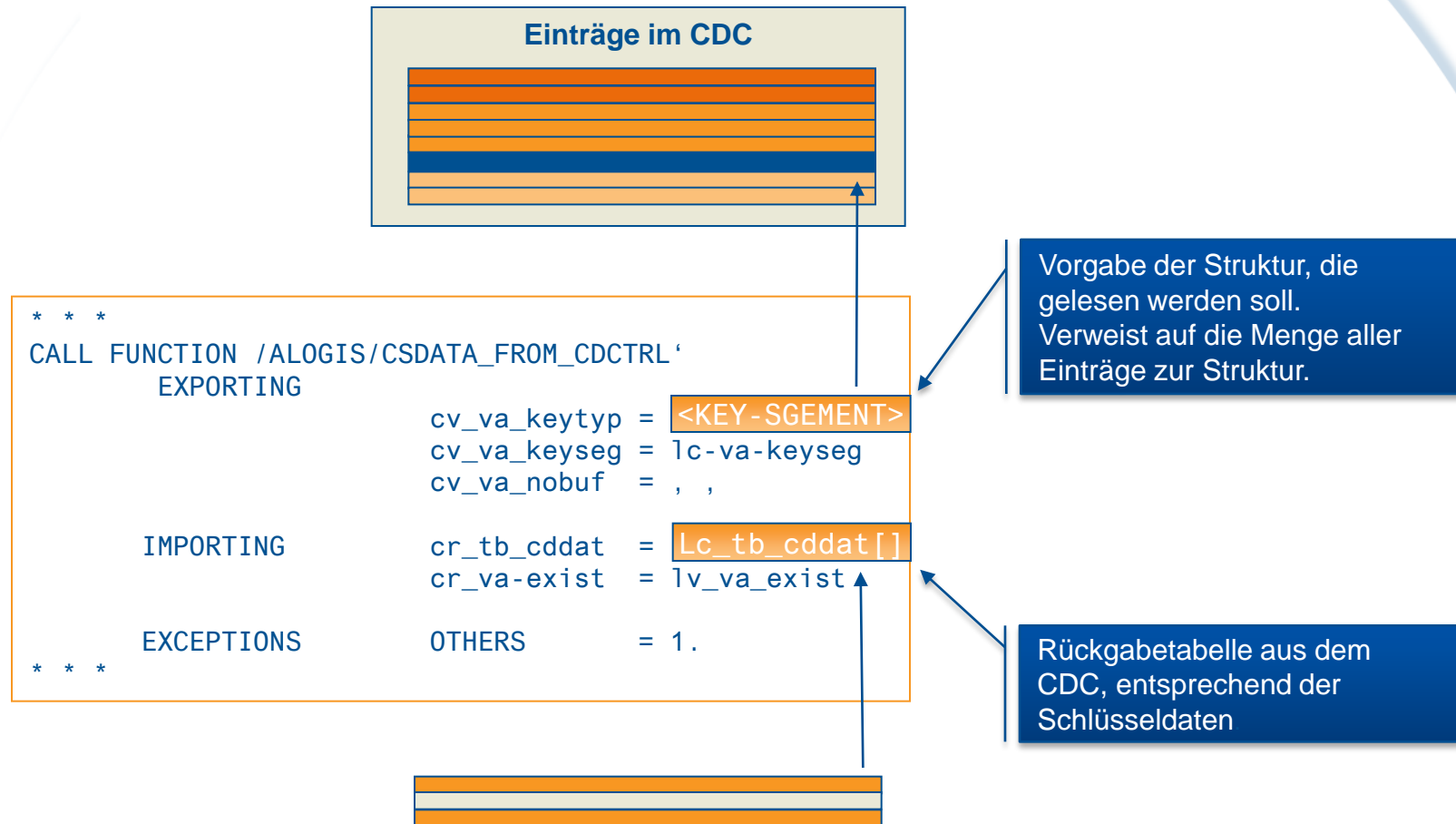


... werden durch eine einheitliche Oberfläche und Funktionsbausteine für alle Strukturen ersetzt

alogis Customer Data Control

Werk/Mart	NK	A..	ALg/Bearbeiter	am	um
*	*		KÜHNE	28.06.2004	15:53:14
++01	*	Z0	KÜHNE	28.06.2004	15:52:13
0001	*	Z0	KÜHNE	28.06.2004	15:51:44
0001	FERT	Z2	X KÜHNE	28.06.2004	15:51:24
0001	HALB	Z1	X X KÜHNE	28.06.2004	15:51:00

# Aufruf im Programm



# Beispiel: Dialog

BW-Matrix


KeySegment(Typ): /ALOGIS/KEY\_CSDATA  CD-Key: Teilbereich für Beisp...

DatenSegment(Typ): /ALOGIS/DAT\_CSDATA  CD-Data: Teilbereich für Beis...

Bezeichnung: Beispiel: CS-Zugriffe auf CustDataCntrl

am: 28.06.2004

Letzte Änderung von: KÜHNE um: 15:50:24



Werk	MARt	NK	A...	ALg	Bearbeiter	am	um
*	*				KÜHNE	28.06.2004	15:53:14
++01	*	Z0			KÜHNE	28.06.2004	15:52:13
0001	*	Z0			KÜHNE	28.06.2004	15:51:44
0001	FERT	Z2	X		KÜHNE	28.06.2004	15:51:24
0001	HALB	Z1	X	X	KÜHNE	28.06.2004	15:51:00

- Werk und MARt (Materialart) sind die Schlüsselfelder der Struktur.
- NK (Nummernkreis) ist ein Wert, der bei der Anlage der Sicht verwendet werden soll.  
„A“ steht für automatische Anlage einer Sicht 1, ebenso „ALg“ (automatische Anlage Sicht 2).
- Danach kommen die vom CDC selbst verwalteten Felder zur Protokollierung.

# Beispiel: Strukturen

**Dictionary: Struktur anzeigen**

Struktur: /ALOGIS/KEY\_CSDATA aktiv

Kurzbeschreibung: CD-Key: Teilbereich für Beispiel(CS-Zugriff)

Eigenschaften | **Komponenten** | Eingabehilfe/-prüfung | Währungs-/Mengenfelder

Komponente	Komponententyp	DTyp	Länge	DezSt...	Kurzbeschreibung
WERKS	WERKS_D	CHAR	4	0	Werk
MTART	MTART	CHAR	4	0	Materialart

Aufbau des Schlüssels im SAP DDIC.

Struktur: /ALOGIS/DAT\_CSDATA aktiv

Kurzbeschreibung: CD-Data: Teilbereich für Beispiel(CS-Zugriff)

Eigenschaften | **Komponenten** | Eingabehilfe/-prüfung | Währungs-/Mengenfelder

Komponente	Komponententyp	DTyp	Länge	DezSt...	Kurzbeschreibung
NUMKR	NUMKR	CHAR	2	0	Nummernkreis
MMDKZ	/ALOGIS/MMDKZ	CHAR	1	0	DispoSicht(automatische Anlage)
MMLKZ	/ALOGIS/MMLKZ	CHAR	1	0	LagerortSicht(automatische Anlage)

Aufbau des Datenbereichs im SAP DDIC.

# Beispiel: Zugriff im Programm

```

-----*
* form csdata_invest      (Wortschöpfung: CuStomizing_INVESTigate) *
-----*
* Ermittlung des relevanten Datensatzes(Teilbereich: CS-Beispiel) *
* aus dem CustomerDataControl. Anhand der Selektionsdaten wird nach *
* dem präzisest zutreffenden Datensatz gesucht: *
-----*
* --> ... : ... *
* <-- C.by R: cr_wa_bwdat - Bw-Arbeitsdaten *
-----*
FORM csdata_invest CHANGING cr_wa_bwdat TYPE ty_wa_bwdat.
FIELD-SYMBOLS: " ----- InformationsSegmente für ... -----
  <lc_va_inf> TYPE ANY,

  <lc_va_key> TYPE ANY,
  <lc_va_dat> TYPE ANY. " Anwendungsdaten
DATA: " ----- interne Tabelle(lokalere Nutzung) für ...
lc_tb_cddat TYPE /alogis/ty_tb_cddat. " ... Bw-Daten
DATA: " ----- WorkArea(lokalere Nutzung) für ...
lc_wa_cddat TYPE /alogis/ty_wa_cddat, " ... Bw-Daten

lc_wa_keyseg TYPE /alogis/key_csdata, " ... CsData(BW-
lc_wa_datseg TYPE /alogis/dat_csdata.
DATA: " ----- Transferhilfen für ...
lc_va_keyseg TYPE /alogis/keyseg, " ... KeySegm(separ
lc_va_exist TYPE c. " ... aktives UserE

REFRESH: lc_tb_cddat.
CLEAR: lc_va_exist.

" ===== Schlüssel für BW-Matrix(UNICODE-fähig)
_assign lc_va_keyseg to <lc_va_key> cast /alogis/key_csdata.
_struct <lc_va_key> with cr_wa_bwdat-werks to 'WERKS'.
_struct <lc_va_key> with cr_wa_bwdat-mtart to 'MTART' .

```

Dieses Coding wird durch Kopieren der Vorlage ins Programm eingefügt. Es müssen nur die Stellen geändert werden, die die gewünschten Datenstrukturen beschreiben.

```

va_subrc = co_mistk. " ===== FBS: Hole Treffer der BW-Matrix ====
CALL FUNCTION '/ALOGIS/CSDATA_FROM_CDCTRL'
  EXPORTING cv_va_keytyp = '/ALOGIS/KEY_CSDATA'
            cv_va_keyseg = lc_va_keyseg
            cv_va_nobuf = ' '
  IMPORTING cr_tb_cddat = lc_tb_cddat[]
            cr_va_exist = lc_va_exist
  EXCEPTIONS OTHERS = 1.

IF sy-subrc <> 0 OR lc_va_exist <> 'X'. " ---- nichts gültiges !! ----
  " Fehlermeldung ...
  EXIT.
ENDIF.

SORT lc_tb_cddat BY keyseg DESCENDING.
" ----- Hole präzisesten Eintrag ----
READ TABLE lc_tb_cddat INTO lc_wa_cddat INDEX 1.
CHECK sy-subrc = 0 OR sy-subrc <> 0. " für erweit. Syntax !!

va_subrc = co_ok. " ===== BW-Daten rückgeben(UNICODE-fähig) !! =====
_assign lc_wa_cddat-datseg to <lc_va_dat> cast /alogis/dat_csdata.
_field 'NUMKR' of <lc_va_dat> to cr_wa_bwdat-numkr.
_field 'MMLKZ' of <lc_va_dat> to cr_wa_bwdat-mmlkz.
_field 'MMDKZ' of <lc_va_dat> to cr_wa_bwdat-mmdkz.
ENDFORM. " ===== Ende des form's 'csdata_invest' =====

```

Ermittlung des am besten zutreffenden Rückgabewertes.

# Auswerten im Programm

```
START-OF-SELECTION.  
  wa_bwdat-werks = '0001'.  
  wa_bwdat-rtart = 'FERT'.  
  PERFORM csdata_invest " ----- daten zum Teilbereich 'csdata' ----  
    CHANGING wa_bwdat.  
  IF wa_bwdat-mmdkz = 'A'.  
  ...
```

Modus CS – Customizing:  
(PERFORM csdate\_invest)

Zu einem vollständig qualifizierten Schlüssel werden auch Sätze mit teilqualifizierten Schlüsselwerten zurück gegeben.  
\* Ersetzt eine Zeichenfolge  
+ ersetzt ein einzelnes Zeichen

Modus MC – Matchcode:  
(PERFORM mcdate\_invest)

Zu einem teilqualifizierten Schlüssel werden alle passenden Sätze ermittelt.

In der Rückgabetable des CDC steht der Schlüssel, mit dem am meisten übereinstimmenden Schlüssel an erster Stelle

Im Aufruf werden die Schlüsselfelder gefüllt. Anschließend kann auf die zurückgegebenen Werte zugegriffen werden.

# CDC Zusammenfassung

- Eine Datenbanktabelle für  $n$  Strukturen
- Eine Logik für den Zugriff
- Einheitlicher Benutzerdialog
- Zentrale Übersicht über kundenspezifische Steuerungsfelder
- Leichte Parametrierung von kundenindividuellen Programmen
- In Kombination mit dem CEC zur Steuerung von Exits nach betriebswirtschaftlichen Kriterien