

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 1 von 56

SAP-Programmierrichtlinie zur Erstellung von Objekten im SAP-Netweaver

Stand: 01.02.2025
Status: ABGESTIMMT
Verfasser: IT.BA.SAP

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 2 von 56

Inhaltsverzeichnis

1	Einleitung	6
1.1	Zweck des Dokuments / Zielgruppe	6
1.2	Berechtigung zur Ausführung von Workbenchaktivitäten	6
1.3	Grundsätze der Programmierung bei der TK für SAP-Objekte	6
1.4	Fortschreibung des Dokuments / Ausnahmen / Abstimmung	6
1.5	Weiterführende Literatur	7
2	Strukturierte Programmierung	8
2.1	Trennung von Logik und Ablaufsteuerung	8
2.2	Vermeidung tiefer Verschachtelungen	8
2.3	Bildung kleiner überschaubarer Einheiten	8
2.4	Verwendung von Literalen	8
2.5	Verwendung von Nachrichten	9
2.6	Verwendung von ungenutztem Coding	9
2.7	Gültigkeitsbereich von Variablen / Inline-Definition	9
2.8	Verwendung von Macros	9
2.9	Verwendung von Booleans	9
2.10	Verwendung von Includes	10
3	Objektorientierte Entwicklung in ABAP	11
3.1	ABAP Unit Testklassen	11
4	Entwicklung von WebDynpro	12
4.1	Anwendungscoding	12
4.1.1	Allgemeine Regeln	12
4.1.2	Verwendung von Assistenz-Klassen	12
4.2	Kontext	14
4.3	Gui	14
4.4	Komponenten	15
4.5	Kontroller	15
4.6	Datenbeschaffung	15
5	Verwendung des XFT-Frameworks	16
5.1	Allgemeines	16
5.2	Namenskonventionen für Prozessframework von xft	16
5.3	Customizing	16
5.4	Datenhaltung der Prozessakten	16
5.5	Documentcomposer	16
5.6	Queuemanager	17
5.7	WebDynpro innerhalb XFT	17
5.8	Plausibilitätsprüfungen (Validierungsmethoden zum Personalmanager)	18
5.9	Allgemeine Services	18
5.10	Konstanten	18
6	Entwicklung mit SAPUI5	19
6.1	Übergreifendes	19
6.1.1	Entwicklungsumgebung	19
6.1.2	Grundlegende Prinzipien	19

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 3 von 56

6.1.3	Regelungen zur Kompatibilität	20
6.2	Anwendungsgrundstruktur	20
6.2.1	SAPUI5-App	20
6.2.2	BSP-Applikation	21
6.2.3	Ordnerstruktur	21
6.2.4	Besonderheiten bzgl. Git-Versionierung und Deployment	21
6.3	Anwendungsbestandteile	21
6.3.1	Applikationseinstieg	21
6.3.2	Views / XML-Views	22
6.3.3	Controller	22
6.3.4	Component Configuration / App Descriptor	23
6.3.5	JavaScript	23
6.3.6	Darstellung Texte, Internationalisierung, Barrierefreiheit, Geräteunabhängigkeit 24	
6.3.7	Mocking und Maschinelle Tests	24
6.4	OData Backend / Frontend	25
6.5	GitHub	26
6.6	Erweiterung von SAP-Standard-Apps	26
7	Einzelaspekte bei der Entwicklung	27
7.1	Verwendung von Log-Points und Assertions	27
7.2	Erweiterungskategorien bei Strukturen und Tabellen	27
7.3	Verwendung von Validierungen und Substitutionen	27
7.3.1	Validierungen	27
7.3.2	Substitutionen	27
7.4	Verwendung von logischen Dateinamen	27
7.4.1	Logischer Pfadname (Präfix)	27
7.4.2	Bildung des Pfad- / Dateinamen im Programm	27
7.5	Konventionen bei der Kommunikation via EAI-Schicht	28
7.6	Sonderregeln für die Funktionalität in Verbindung mit TKeasy-Wrapper-Gui	29
7.7	Sprachkompatibilität	30
7.8	UNICODE-Kompatibilität	30
7.9	Löschverfahren	30
8	Wiederverwendbarkeit	31
8.1	Funktionsbausteine	31
8.2	Forms	31
9	Datenbankrelevantes Coding (SAP HANA)	32
10	Sicherheitsanforderungen	33
10.1	Grundsatz	33
10.2	Maschinelle Prüfung	33
10.3	Datenstrukturen	33
10.3.1	Tabellenberechtigungsgruppe	33
10.3.2	Tabellenprotokollierung	33
10.3.3	Erstellung von Pflegeviews für Tabellen	34
10.4	Programmstrukturen	34
10.4.1	Berechtigungsprüfung bei Transaktionen	34
10.4.2	Hart codierte Berechtigungen / Hintertüren / fest codierte Benutzerkennungen	34
10.4.3	Berechtigungsprüfungen in RFC-fähigen Funktionen	34
10.4.4	Berechtigungsprüfungen in Reports	34

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 4 von 56

10.4.5	Debug-Code (z.B. in ASSERT-Statements)	34
10.4.6	Generischer und dynamischer ABAP-Code	34
10.4.7	Generische Funktionsaufrufe	35
10.4.8	Generische Reports / Programme (ABAP Command Injection)	35
10.4.9	Generische Berechtigungen	35
10.4.10	Generische RFC Destination	35
10.4.11	SQL-Injection / Forceful querying / Generic Table query	35
10.4.12	Directory Traversal	35
10.4.13	Aufrufe in den Kernel.....	35
10.4.14	System Command Injection / Execution	35
10.4.15	Fest codierte Mandantenprüfung / Systemprüfung	36
10.4.16	Mandantenübergreifender Datenzugriff.....	36
10.4.17	Direkte Datenbankänderungen	36
10.4.18	Datenbankzugriff mit EXEC SQL	36
10.4.19	Dateidownload /-upload.....	36
11	Namenskonventionen	37
11.1	Grundlegende Regeln.....	37
11.2	Namenskonvention für Entwicklungsklassen	37
11.3	Namenskonvention für Domänen	37
11.4	Namenskonvention für Codepages	37
11.5	Namenskonvention für Typdefinitionen	37
11.6	Namenskonvention für Nachrichtenklassen und Nachrichten / Syslog-Meldungen.....	37
11.7	Namenskonvention für Funktionsgruppen	37
11.8	Namenskonvention für Funktionsbausteine	38
11.8.1	Allgemeine Funktionsbausteine	38
11.8.2	Funktionsbausteine für RFC von SAP-System zu SAP-System.....	38
11.8.3	Funktionsbausteine zur Ansteuerung von TKeasy via MQ.....	38
11.8.4	Funktionsbausteine zur Ansteuerung von SAP via MQ	38
11.8.5	Funktionsbausteine zur Ansteuerung von Business-Connector-Services.....	38
11.9	Namenskonvention für Includes	38
11.10	Namenskonvention für Reports.....	38
11.11	Namenskonvention für FORM-Routinen.....	38
11.12	Namenskonvention für ENHANCEMENT-Points	39
11.13	Namenskonvention für programminterne Datendeklarationen	39
11.13.1	Grundsatz	39
11.13.2	Variablen.....	39
11.13.3	Konstanten.....	39
11.13.4	Variablen aus dem Selektionsbild	39
11.14	Namenskonvention für Parameter von Funktionsbausteinen / Methoden	39
11.15	Namenskonventionen für Funktionsbausteine zu Zeitpunkten im FS-CD	40
11.16	Namenskonventionen für Funktionsbausteine zu allgemeinen Exits im FS-CD	40
11.17	Namenskonventionen für objektorientierte Entwicklung	40
11.17.1	Interfaces	40
11.17.2	Klassen	40
11.17.3	Exception-Klassen.....	40
11.17.4	Instanz-Variablen	41
11.17.5	Klassen-Variablen	41
11.17.6	Methoden.....	41
11.18	Namenskonventionen für das XFT-Framework	43
11.19	Namenskonventionen für Web Dynpro ABAP.....	43

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 5 von 56

11.19.1	Einleitung	43
11.19.2	Modell	44
11.19.3	View / Window.....	47
11.19.4	UI-Elemente	47
11.19.5	Plugs.....	48
11.19.6	Controller	48
11.19.7	Controllermethoden.....	49
11.19.8	Assistance-Klassen	49
11.19.9	Konfigurationen	49
11.19.10	Floorplan-Manager	49
12	Einheitliches Layout.....	50
12.1	Programmaufbau.....	50
12.2	maximale Zeilenlänge	50
12.3	Kommentierung	50
12.4	Maximale Befehlsanzahl pro Zeile.....	51
12.5	Hervorhebung von Verarbeitungsblöcken durch Einrückungen.....	51
12.6	Verständlichkeit von Abfragen.....	51
12.7	Einsatz maschineller Unterstützung	51
12.8	Verwendung von Templates	52
13	Kennzeichnung der Sourcen.....	53
14	Dokumentationsdokumente "technische Entwürfe".....	53
15	Maschinelle Prüfung	53
16	Anhang.....	54
16.1	Checklisten für die Programmprüfung	54
16.2	Beispielprogramm.....	55
16.3	Gültige Nachrichtenklassen / Syslog-Meldungsklassen	56
16.4	Templates	56
16.5	Hauptgebiete / Untergebiete	56
16.6	Liste der Programmberechtigungsgruppen.....	56

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 6 von 56

1 Einleitung

1.1 Zweck des Dokuments / Zielgruppe

In diesem Dokument werden die bei der TK einzuhaltenden Standards für die Programmierung auf Basis SAP Netweaver (ABAP / SAPUI5 / Fiori / WebDynpro for ABAP / XFT-Framework) beschrieben. Für Implementierungen zum SAP Business Connector sind keine TK-eigenen Standards festgelegt.

Andere Entwicklungstechnologien der SAP werden bei der TK nicht für Eigenentwicklungen verwendet.

Dieses Dokument ist ein Arbeitsmittel für die einschlägigen ArBa des GB IT.

Zielgruppe des Dokuments sind alle internen und externen Mitarbeitenden, die an der Eigenentwicklung bezüglich SAP beteiligt sind.

Sofern im Dokument abstim- oder genehmigungspflichtige Sachverhalte aufgeführt sind, ist die Abstim- oder Genehmigungsinstanz das Team IT.BA.SAP.

Nicht enthalten in diesem Dokument sind

- Richtlinien für das Customizing,
- Richtlinien für die Anwenderdokumentation,
- Richtlinien für die Beschreibung der maschinellen Unterstützung von Prozessen.

1.2 Berechtigung zur Ausführung von Workbenchaktivitäten

Die Berechtigung zur Ausführung von Workbenchaktivitäten ist vor Aufnahme solcher Tätigkeiten bei der Teamleitung IT.BA.SAP zu beantragen. Für die Eingabe des Entwicklerschlüssels in den SAP-Systemen der TK ist zu beachten, dass hier nur zugeteilte Entwicklerschlüssel benutzt werden dürfen, deren Zuteilungszeitpunkt nicht länger als einen Monat in der Vergangenheit liegt.

1.3 Grundsätze der Programmierung bei der TK für SAP-Objekte

Die Programmierung ist so auszuführen, dass die drei Hauptziele

- Wiederverwendbarkeit
- einfache Wartbarkeit
- Korrektheit

erreicht werden. Grundsätzlich ist ebenfalls der anerkannte Stand der Technik einzuhalten.

Um die Hauptziele zu erreichen, hat sich die ABAP-Programmierung an den folgenden Prinzipien auszurichten:

- strukturiert
- wiederverwendbar
- zukunftsicher
- einheitliches Layout
- einheitliche Namenskonventionen
- Sicherstellung der Vollständigkeit

Die zu den Prinzipien gehörigen Regeln werden im Folgenden beschrieben.

1.4 Fortschreibung des Dokuments / Ausnahmen / Abstimmung

Dieses Dokument wird in regelmäßigen Abständen fortgeschrieben.

Fortschreibungen:

2025/02: Anpassungen SAPUI5

2024/02: Ergänzungen zu ABAP Unit Tests, zur Entwicklung mit SAPUI5,
zur Implementierung in der Cloud

2021/12: Entwicklung mit UI5 aufgenommen

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 7 von 56

2019/12: Org.-Änderungen, diverse Anpassungen wg. neuer Konstrukte
 2017/07: Organisatorische Änderungen, Security-Ergänzungen / neue SQL-Konstrukte
 2015/12: Unicode-Standard, Verzicht auf Macros
 2013/08: Neue Kapitel für Typdefinition, Booleans, Änderung Includes, Erweiterung EAI/MQ
 2012/04: Ergänzung Sicherheitsaspekte / WebDynpro / Entfall Internet-Services
 2011/12: redaktionelle Änderungen
 2010/05: Ergänzungen OO, WebDynpro, Release ECC6.0
 2004/11: Aufnahme Namenskonvention für Funktionsbausteine zur MQ-Ansteuerung
 2004/12: Erweiterung für Business-HTML / Internetservices

Abweichungen von den nachfolgend beschriebenen Standards sind vor der Arbeitsausführung mit IT.BA.SAP einvernehmlich abzustimmen. Ebenfalls sind ggf. entstehende Unklarheiten oder Verständnisprobleme vor der Arbeitsausführung mit IT.BA.SAP zu klären.
 Verstöße gegen die Richtlinie gehen zu Lasten der jeweiligen Hersteller der entwickelten Objekte und sind von diesen zu beheben.

Implementierungen im Umfeld SAP Cloud bedürfen einer vorherigen und begleitenden Abstimmung mit IT.BA.SAP.

1.5 Weiterführende Literatur

- Vorgehensmodell der TK (Intranet der TK)
- Empfohlene Lektüre: Buch "ABAP-Programmerrichtlinien" (Keller / Thümmel), Verlag Galileo Press
- Empfohlene Lektüre: Buch "Sichere ABAP-Programmierung" (Wiegenstein et al.), Verlag Galileo Press
- Die bindenden Festlegungen für die Entwicklung bezogen auf die verschiedenen Firmen auf den SAP-Systeme der TK sind dokumentiert in X:\TK SAP R3\01 R3 (Schnittstellen)\9 TK Dokumentation\Konzepte Soll\BC-KO-Softwarelogistik TK-TK GmbH.doc.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 8 von 56

2 Strukturierte Programmierung

2.1 Trennung von Logik und Ablaufsteuerung

ABAP-Programme des Typs Report oder Dialogverarbeitung werden anhand der Ereignisse strukturiert. Unter jedem Ereignis werden 1- max. 10 Forms angeordnet. Auf dieser Ebene ist nur der PERFORM-Befehl zulässig sowie einfache Abfragen. Das Coding wird in den Forms angeordnet.

Eine Form hat immer nur eine von beiden nachfolgenden Funktionen:

Steuerung des Programmablaufs

Durchführung eines zusammenhängenden Verarbeitungsblocks

Für Funktionsbausteine gilt entsprechendes wobei die Gliederung wie folgt vorgenommen wird:

Eine FORM darf nur von genau einem Funktionsbaustein benutzt werden

Die Schnittstelle erfolgt via USING-Klausel

Es erfolgt in der Form kein Zugriff auf Daten über den USING-Umfang hinaus.

Verwendung von FORMs aus Top-Includes von Funktionsgruppen

- Diese Forms dürfen von Funktionsbausteinen aus der Funktionsgruppe benutzt werden, in der sie definiert sind

2.2 Vermeidung tiefer Verschachtelungen

Verzweigungsstrukturen haben einfach zu sein. Komplizierte Fälle sind z.B. über Forms aufzuteilen.

Als Faustregel gilt: Vermeidung von mehr als 3 Verschachtelungstiefen.

2.3 Bildung kleiner überschaubarer Einheiten

Es wird aufgeteilt in Forms, Funktionen bzw. Methoden. Pro Einheit sollen ca. 30 Zeilen ausführbarer Code vorhanden sein.

2.4 Verwendung von Literalen

Literale sind grundsätzlich als Konstanten zu deklarieren (Ausnahme: Ziffern, die als Ziffern verwendet werden / Report- und Spaltenüberschriften / Eingabefeldbeschreibungen z.B. im Selektionsbildschirm, Funktionsnamen, Nachrichtenklassen)

Für die Definition übergreifender Konstanten (globale Gültigkeit) werden globale Klassen (ZVG_) verwendet. Konstanten mit lokaler Gültigkeit werden ohne Includes gebildet.

Für die Prüfung gegen Literale / Literalgruppen werden Vergleichsklassen erstellt. Die Klasse beginnen mit ZVG_ gefolgt vom Wert der geprüft wird. Vergleichsklassen werden für Vergleiche mit global gültigen Parametern erstellt.

Konstanten, die sowohl innerhalb als auch außerhalb von Vergleichsklassen verwendet werden, sind in der Vergleichsklasse als Konstanten mit der Sichtbarkeit Public zu definieren.

```

*           Klassendefinition laden
CLASS zvg_bmagr      DEFINITION LOAD.
...
*           Konstanten
CONSTANTS: k_true    TYPE i VALUE 1
...
*           handelt es sich um eine Familienpause ?
IF zvg_bmagr=>is_fam_pause( zpernr-bmagr ) = k_true.
  WRITE 'TRUE'.
ELSE.
  WRITE 'FALSE'.
ENDIF.

```


SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 9 von 56

2.5 Verwendung von Nachrichten

Nachrichten sind zu verwenden, um Programmmzustände an die Umgebung zu signalisieren. Generische Nachrichten dürfen nur in begründbaren Ausnahmefällen verwendet werden. Die eindeutige Nachrichtenverwendung ermöglicht per Nachrichten-Nummer eindeutig Programmmzustände zu signalisieren.

2.6 Verwendung von ungenutztem Coding

Auskommentierter Code ist nur zulässig in modifizierten SAP-Programmen, um das SAP-Originalcoding zu sehen. In TK-eigenen Programmen ist auskommentierter Code zu entfernen. Ausnahmefälle sind abzustimmen.

In dem Programm deklarierte Variablen, Konstanten, Strukturen, Tabellen etc. müssen eine Verwendung besitzen. Über die erweiterte Syntaxprüfung werden ungenutzte Deklarationen erkannt.

Für Programme im Kundennamensraum wird der Modifikationsassistent nicht verwendet. Ausgenommen sind Modifikationen durch Anpassungen außerhalb des Originalsystems (z.B. im System E22 bei Originalsystem E21).

2.7 Gültigkeitsbereich von Variablen / Inline-Definition

Variablen sind immer so lokal wie möglich und so global wie nötig zu definieren.

Sie sind immer am Anfang des Gültigkeitsbereichs zu definieren. Die lokalen Deklarationen dürfen nicht über die Implementierung der Prozedur verteilt werden. Abweichende Regelung für Inline-Deklarationen unten.

Werden globale Variablen verwendet und diese an eine FORM als Argument in der FORM-Aufrufleiste übergeben, dürfen in der gerufenen FORM diese globalen Variablen nicht direkt angesprochen werden, sondern nur über den als Argument übergebenen Namen.

Verwendung von Inline-Deklarationen: Sie können verwendet werden zur Vereinfachung und Vermeidung von Laufzeitfehlern wegen falscher Datentypen.

Regeln Inline-Deklaration für DATA- und ASSIGN-Deklarationen

- Inline-Deklarationen dürfen nicht für globale Verwendung definiert werden
- Deklaration so lokal wie möglich,
- wenig Verwendungen der Deklaration, wenn sie nicht am Anfang definiert wird
- Verwendung in logischen Blöcken (loop, perform, if, case)
- keine Vermischung mit dynamischer Variablenverwendung

2.8 Verwendung von Macros

Die Verwendung von Macros ist nicht zulässig.

2.9 Verwendung von Booleans

Folgende Typen und Konstanten sind erlaubt:

- abap_bool mit den Konstanten abap_true und abap_false
- x_or_space mit den Konstanten der Klasse ZVG_BOOL K_TRUE, K_FALSE
- zbool mit den Konstanten der Klasse ZVG_BOOL K_TRUE_1, K_FALSE_0
- wdy_boolean x oder space
- boole_d im OData-Kontext
- ZEAI_BOOL_0_1 für die EAI-Kommunikation
- ZEAI_BOOL_SPACE_X für die EAI-Kommunikation

Es werden keine eigenen Konstanten für TRUE / FALSE verwendet.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 10 von 56

2.10 Verwendung von Includes

Includes werden verwendet, um bei sehr großen Programmen Codingteile aufzunehmen und hierdurch die Bearbeitbarkeit der Source zu erleichtern. Die in diesem Zusammenhang gebildeten Includes dürfen nur aus genau einer Source heraus verwendet werden.

Includes werden nicht für die Auslagerung von FORMS innerhalb von Reports eingesetzt.

Verwendung von TOP-Includes bei Funktionsgruppen:

TOP-Includes dürfen für die Definition globaler Variablen verwendet werden, die dem Datenaustausch zwischen Funktionsbausteinen der gleichen Funktionsgruppe dienen, falls dieser Datenaustausch nicht über Schnittstellenversorgung realisiert werden kann. Vorgehensweise ist nur als Ausnahme definiert, die abgestimmt werden muss.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 11 von 56

3 Objektorientierte Entwicklung in ABAP

Die Regeln für die objektorientiert Entwicklung sollen folgende Grundsätze sollten berücksichtigen:

- Persistenzklassen werden separat angelegt
- Es gelten die für Funktionsbausteine geltenden Prinzipien der Strukturierung
- Vererbung ist auf < 5 Ebenen beschränkt

Für das Coding, das in der Ausimplementierung der Methoden einer Klasse geschrieben wird, gelten die Programmierrichtlinien aus den vorhergehenden Kapiteln.

Für weitere Regelungen ist eine Abstimmung mit IT.BA.SAP über die Erstellungsregeln auszuführen.

3.1 ABAP Unit Testklassen

Testklassen, welche **Testmethoden für eine „normale“ Klasse** bereitstellen, sind üblicherweise **lokal** innerhalb der zu testenden Klasse zu definieren.

Bei kritischen Prozessen können Testklassen als globale Klasse angelegt werden, um die Trennung zwischen Ausführungs- und Testcoding zu erhöhen. In diesem Fall sind die Klassennamen so zu wählen, dass sie sich nur im Präfix unterscheiden.

Testklassen, welche **Testmethoden für Funktionsbausteine** bereitstellen, sind als **globale** Testklasse zu definieren und im selben Paket wie die Funktionsgruppe zu hinterlegen.

Die Namenskonventionen sind in Kapitel 11.17.2 beschrieben.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 12 von 56

4 Entwicklung von WebDynpro

Web Dynpro Anwendungen basieren auf den Model-View-Controller-Ansatz (MVC). Dies bedeutet, u.a. dass in den Views möglichst keine Business Logik implementiert wird sondern an den Controller delegiert werden soll.

WebDynpro-Entwicklungen können (müssen aber nicht) auf einem Standard-Floorplan der SAP vorgenommen werden, wenn die Aufgabenstellung dies als sinnvoll erscheinen lässt. Dies ist vor Beginn einer Entwicklung mit IT.BA.SAP abzustimmen.

Per Parameter wird in der Anwendung keine Personalisierung durch den Benutzer zugelassen (Attribut WDDISABLEUSERPERSONALIZATION). Ausnahmen hiervon sind vorher abzustimmen.

Die Empfehlungen und Richtlinien der SAP für die Entwicklung anhand der Beschreibungen in

- Einstieg in Web Dynpro ABAP (Ofenloch, Schwaiger)
- Web Dynpro ABAP, das umfassende Handbuch (Ofenloch, Schwaiger)
- Floorplan Manager für Web Dynpro ABAP (Frambach, Hoeg)

sind als Grundlage zu nehmen. Namenskonventionen s. 11.17

Für die Erzielung einer optimalen Performance sind folgende Regeln zu beachten:

- das Anwendungsdesign sollte möglichst simpel und kommunikationsarm sein
- nicht mehr benötigte Instanzen sollten zeitnah gelöscht werden
- die Antwortzeit der Serververarbeitung sollte < 300ms sein
- die Antwortzeit der Clientverarbeitung sollte < 700ms sein

4.1 Anwendungscoding

4.1.1 Allgemeine Regeln

- das Anwendungscoding sollte in eigenen ABAP-Klassen sein
- Assistenzcontroller sind performanter als Komponentencontroller bei Aufruf von Anwendungscoding

4.1.2 Verwendung von Assistenz-Klassen

4.1.2.1 Vorteil

Die Methoden der Assistenz-Klasse können von jedem View, Window und Controller via `wd_assist->[Aliasname]` aufgerufen werden.

Die Verwendung des Wizards zur Codegenerierung entfällt weitgehend.

Das Coding bleibt leicht wartbar und ist zentral erweiterbar.

4.1.2.2 Verwendung

Jede WDA-Komponente hat die Basis Assistenz-Klasse `ZCL_WD_BASIS_ASSIST` direkt oder indirekt zu verwenden.

Die zentrale Assistenzklasse `ZCL_WD_BASIS_ASSIST` stellt dem Anwender eine Reihe von Methoden zur Verfügung:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx		Seite 13 von 56
Nur für den Dienstgebrauch		

Klassenschnittstelle

ZCL_WD_BASIS_ASSIST

realisiert / aktiv

Eigenschaften

Interfaces

Friends

Attribute

Methoden

Ereignisse

Typen

Aliases



☐ Filter

Interface	Abstrakt	Final	Nur mo...	Beschreibung
<u>IF_WD_COMPONENT_ASSISTANCE</u>	☒	☐	☐	Model-Interface für Web Dynpro
ZIF_WD_CONTEXT_HELPER	☐	☐	☐	Interface für Context Zugriffe
ZIF_WD_MESSAGE_HELPER	☐	☐	☐	Interface für den Message-Manager

Derzeit werden Methoden zu folgenden Themen bereitgestellt:

- Textermittlung
- Kontext-Bearbeitung
- Fehlermeldung-Bearbeitung

Werden weitere anwendungsspezifische Funktionalitäten benötigt, so bietet sich die Verwendung einer speziellen Assistenz-Klasse an, die von ZCL_WD_BASIS_ASSIST erbt.

Web-Dynpro-Component	ZWDC_BC_CHANGE_CONFIG	aktiv	
Beschreibung	Pflege Konfigurationsdaten im BC		
Assistance-Klasse	ZCL_BC_CHANGE_CONFIG_ASSIST		
Anleger	209323	Erstellungsdatum	30.03.2012
letzter Änderer	159775	Änderungsdatum	02.04.2012
Originalsprache	DE	Paket	Z01_11_30_BC
☒ Barrierefreiheit-Prüfungen aktiv			

Klassenschnittstelle	ZCL_BC_CHANGE_CONFIG_ASSIST	realisiert / a		
Eigenschaften	Interfaces	Friends	Attribute	Methoden
 Superklasse  Vererbung aufheben  Vererbung ändern				
Erbt von	ZCL_WD_BASIS_ASSIST	☐ Nur Modell		
Beschreibung	Assistance-Klasse für ZWDC_BC_CHANGE_CONFIG...			
Inst.-Erzeugung	Public			
☐ Final				

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 14 von 56

4.1.2.3 Initialisierung

Damit die geerbten Methoden der Assistenzklasse verwendet werden können, ist die Klasse zum Zeitpunkt WDDOINIT des COMPONENTCONTROLLER zu initialisieren.

Dies kann durch das folgende Coding vorgenommen werden:

```
METHOD wddoinit .
    DATA: lv_wd_controller          TYPE REF TO if_wd_controller.

    CHECK wd_assist IS BOUND.
    lv_wd_controller = wd_this->wd_get_api( ).
    wd_assist->init( lv_wd_controller ).
ENDMETHOD.
```

4.1.2.4 Aufrufbeispiele

```
DATA: lv_selection          TYPE wd_this->element_selektion
      ,lt_return            TYPE STANDARD TABLE OF bapiret1
      ,lv_return            LIKE LINE OF lt_return.

*                               Kontext auslesen
wd_assist->get_attributes( EXPORTING path = 'SELEKTION'
                          IMPORTING values = lv_selection ).

*                               Kontext füllen
wd_assist->bind_table( path = 'ET_RETURN'
                     new_items = lt_return ).

*                               Meldungen ausgeben
LOOP AT lt_return          INTO lv_return.
    wd_assist->report_t100_message( lv_return ).
ENDLOOP.
```

4.2 Kontext

- zu große Kontexte sollen vermieden werden
- tief geschachtelte Kontexte sollen vermieden werden
- dynamische Attribute sollen vermieden werden
- der Kontext soll jeweils nur die Daten enthalten, die für Kommunikation zwischen Window und View benötigt wird
- häufiges Aktualisieren der Kontextdaten soll vermieden werden

4.3 Gui

- der Aufbau der GUI soll möglichst simpel sein
- tiefe Verschachtelungen im Layout sollten vermieden werden < 5 Ebenen
- die GUI sollte auf eine typische Seite des Browserfensters passen (Vermeidung von Scrollen)
- GUI Elemente, die viel Client Rendering Zeit benötigen sollen nicht in Anwendungen mit hoher Nutzer-Last eingesetzt werden (siehe Tabelle der SAP)

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 15 von 56

UI Price Tags in Web Dynpro ABAP

Overview of Client Rendering CPU Time for Web Dynpro UI Elements



UI screen or UI element's technical name2	Rating category	Client Rendering CPU time
↑ LinkToAction	Very Fast	0.5 ms
↑ Button	Very Fast	1.2 ms
↑ InputField (default)	Very Fast	1.5 ms
↑ InputField with value help active	Very Fast	3.0 ms
👉 Empty Web Dynpro page (Container Layout: GridLayout)	Fast	30 ms
👉 Enable Scrolling, Adding a scrollbar to a container (e.g. TransparentContainer)	Medium	200-300 ms
👉 WD Table Control (default – the short description field is used header)	Medium	180 ms
👉 ALV (default)	Medium	250 ms
👉 ALV (editable cells by changing default configuration model)	Medium	340 ms
👉 Open new Internet Explorer window (about:blank)	Medium	250 ms
👉 Startup FlashIsland	Medium	400 ms
↓ Open simple flash application in FlashIsland	Slow	700 ms
↓ Open medium complex flash application in FlashIsland	Slow	1250 ms
↓ Open complex flash application in FlashIsland	Slow	4300 ms

Source: Fabian Schlarmann, SAP TD CoE Performance, "UI Price Tags for Web Dynpro ABAP", 10/2009, SAP Internal

© 2010 SAP AG. All rights reserved. / CD203 / Page 67

4.4 Komponenten

- Komponenten mit nur einer View sollt vermieden werden
- eine Komponente soll nicht mehr als 15 Views haben
- generische Komponenten sollen vermieden werden

4.5 Kontroller

- ein Kontroller soll nicht mehr als 8 Kontroller verwenden

4.6 Datenbeschaffung

Es sollen immer nur die Daten beschafft werden, die für die aktuelle Anzeige benötigt werden (späte Datenbeschaffung).

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 16 von 56

5 Verwendung des XFT-Frameworks

Mit XFT werden alle in einem Prozess anfallenden Informationen, wie Dokumente, E-Mails oder elektronische Belege an einer zentralen Stelle zugänglich gemacht. Anwendungen sind:

- › xft task manager
- › xft personnel file
- › xft contract manager
- › xft queue manager
- › xft document composer

5.1 Allgemeines

Eine ausführliche Einführung und Beschreibung der XFT-Prozesse ist zu finden unter:
X:\TK SAP R3\01 R3 (Schnittstellen)\9 TK Dokumentation\Konzepte Soll\XFT Dokumentation

5.2 Namenskonventionen für Prozessframework von xft

s. 11.18

5.3 Customizing

Die bindenden Festlegungen für das XFT-Customizing der TK sind dokumentiert INFOBRIEF "X:\TK SAP R3\01 R3 (Schnittstellen)\9 TK Dokumentation\Konzepte Soll\XFT Dokumentation\PROC HR-WF Systemdokumentation v1.docx"

5.4 Datenhaltung der Prozessakten

Die Datenhaltung erfolgt in den Standard-XFT-Tabellen und den Erweiterungs-Tabellen:

Kopfdaten: ZHR_<Prozess>FORM_001

Positionsdaten: ZHR_<Prozess>FORM_010

Modell

Die Tabellen werden über die Strukturen gefüllt:

ZHR_<Prozess>_MOD_001_EXT

ZHR_<Prozess>_MOD_010_EXT

Connection

Klassen, von denen geerbt werden muss

Standardklasse:

/XFT/CL_HR_PEM_FORM_CON

Beispiel:

ZCL_HR_UEB_PEM_FORM_CON

Standardklasse:

/XFT/CL_HR_PEM_FORM_MOD

Beispiel:

ZCL_HR_UEB_PEM_FORM_MOD

5.5 Documentcomposer

Für Schriftguterzeugung ist der Documentcomposer zu verwenden. Es werden Worddokumente erstellt und zu pdf-Dokumenten konvertiert.

Klassen, von denen geerbt werden muss

für Metainformationen:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 17 von 56

Standardklasse:
/XFT/CL_MSML_DP_META_INFO_ROOT
Beispiel:
ZCL_HR_UEB_FORM_001_DP_MI

für die Datenermittlung:
Standardklasse:
/XFT/CL_MSML_DP_RUNTIME_ROOT
Beispiel:
ZCL_HR_UEB_FORM_001_DP_RT

für den Context-Service:
Standardklasse:
/XFT/CL_SRM_DYN_REC_MS_DP_SE
Beispiel:
ZCL_HR_UEB_FORM_001_DP_SE

5.6 Queuemanager

Bei der Erzeugung von Queuemanager-Prozessen muss dem Parameter `im_description` als erstes die RecordID in eckigen Klammern übergeben werden, damit später im Queuemanager einheitlich die Vorgangsnummer angezeigt werden kann:

```

29 * Auftrag erzeugen
30 <AKTION>ZHR_UEB_DIO_MAIL_MA_ALL;do Queuemanager;ZHR_UEB_DIO_MAIL_MA<AKTION>
31 lo_request = lo_dio_public->/xft/if_dio_public_input-create_request_s(
32     im_pro_type_id = 'ZHR_UEB_DIO_MAIL_MA'
33     im_description = '({ i_rec_id }) UEB: - Mail an Mitarbeiter { I_PERNR }'
34     im_properties_use = lt_prop_use ).

```

XFT Queue Manager

Selektion

Prozesstyp ID	Zeitraum	Status	Agent
	Angelegt	☒ Bereit	☒ Prozess
von 24.01.2020 14:32:28		☒ Fehler	☒ Archive
bis 24.01.2020 15:00:00		☒ Asynchron	
		☒ Wiederholen	
		☐ Wartet auf Ereignis	

Selektion ausführen Statistik berechnen

Maximale Anzahl 5000 Maximale Anzahl 10000

2 selektiert (Filter aus)

St...	Status	Agent	Dauer (...	Dauer (H...	Beschreibung	Ausf.Info.	Anlegedatum	Angelegt	Angelegt	Se...	Geändert
	Asynchron	Prozess	4	19:22:46	Cycle Bearbeiterprozess: Mitbestimmung: MIT-000...	Wartet	24.01.2020	14:33:02	WF-BAT...	Se...	24.01.2020
	Asynchron	Prozess	4	19:23:19	[MIT-0000037] Mitbestimmung: MIT-0000037 - V...	Wartet	24.01.2020	14:32:28	200760	Se...	24.01.2020

Zeit	Status	Agent	Beschreibung	Anlegedatum	Angelegt	Angelegt	Frühestens	Frühestens	Netto	Dauer (Sek.)
	Asynchron	Prozess	Bearbeiter Prozesse für Cycle gestartet.	24.01.2020	14:33:01	WF-BAT...		00:00:00		0,6897120
	Bereit	Prozess	[FEB11C7545071EDA8FD451222A0DBFA] Mitbe...	24.01.2020	14:32:28	200760		00:00:00		0,0000000

5.7 WebDynpro innerhalb XFT

Es gelten die allgemeinen Programmierrichtlinien zu WebDynpro.

- Die zu entwickelnde Klasse muss erben von

/XFT/CL_HR_PEM_FORM_WDY

Beispiel:

ZCL_HR_UEB_PEM_FORM_WDY

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 18 von 56

- Die zu entwickelnde Komponente muss das Interface implementieren:

/XFT/CI_HR_PEM_FORM_STEP

Beispiel:

ZWDC_UEB_PEM_FORM_ALV

5.8 Plausibilitätsprüfungen (Validierungsmethoden zum Personalmanager)

Plausibilitätsprüfungen sind nach folgendem Beispiel anzulegen.

Beispiel:

ZCL_HR_<Prozess>_SERVICE_PLAUSI

5.9 Allgemeine Services

Allgemeine Services sind nach folgendem Beispiel anzulegen.

Beispiel:

ZCL_HR_<Prozess>_SERVICE

5.10 Konstanten

Konstanten sind grundsätzlich in Vergleichsklassen anzulegen. Konstanten die nicht Vergleichsklassen zugeordnet werden können, sind in der jeweiligen Konstantenklasse des Prozesses zu hinterlegen.

Beispiel:

ZCL_HR_UEB_CONSTANT

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 19 von 56

6 Entwicklung mit SAPUI5

In diesem Kapitel werden die bei der TK einzuhaltenden Standards für die SAPUI5- / SAP-Fiori-Programmierung beschrieben. Bezogen auf die Backend-Anbindung von SAPUI5-Anwendungen sind die in diesem Kapitel enthaltenen Ausführungen als Ergänzung der übrigen Ausführungen der Programmierrichtlinien zu verstehen.

Insbesondere folgende Kontexte sind noch nicht final beschrieben und bedürfen daher stets einer Abstimmung mit IT.BA.SAP:

- CDS-Views / Fiori Elements
- Erweiterungsimplementierungen von SAP- oder kundeneigenen Apps
- Implementierungen mit kundeneigenem HTML-Code
- Implementierungen in der SAP Cloud

Als weiterführende Literatur wird empfohlen:

- SAP Fiori: Implementierung und Entwicklung (978-3-8362-7268-1)
- Das neue SAPUI5-Handbuch (978-3-8362-7523-1)

6.1 Übergreifendes

6.1.1 Entwicklungsumgebung

IDE Zur Entwicklung und zum Deployment von SAPUI5-Anwendungen wird das „SAP Business Application Studio“ (BAS) eingesetzt.

Weitere Entwicklungstools können verwendet werden, sofern die grundsätzliche Kompatibilität der UI5-Projekte innerhalb des BAS sichergestellt bleibt.

GitHub Kundeneigene SAPUI5-Codeartefakte müssen in der TK-GitHub-Organisation „sap“ hinterlegt werden. (Details siehe Kapitel 6.5)

6.1.2 Grundlegende Prinzipien

Strukturierung	Die App-Implementierung erfolgt unter Berücksichtigung einer wiedererkennbaren Anwendungsgrundstruktur. Diese folgt im Üblichen dem SAP-Standard.
MVC-Pattern	Die Apps sind unter Verwendung des Model-View-Controller-Konzepts zu implementieren.
Modularisierung	Modularisierungsmöglichkeiten werden zwecks Wiederverwendbarkeit und Komplexitätsreduktion genutzt.
Sichtbarkeit	Anwendung der Sichtbarkeitsprinzipien (public vs. private). Globale Variablen sind zu vermeiden.
Security	Einhaltung der Content Security Policy zur Verhinderung von Cross-Site Scripting und Code Injection.
Benennung	Entwicklungsobjekte (Projekte, Dateien, Methoden, Variablen, etc.) sind sprechend zu benennen, um eine hohe natürliche Lesbarkeit zu erzielen.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 20 von 56

Formatierung	Die Formatierung des Quellcodes wird anhand der in der Entwicklungsumgebung vorhandenen Standardfunktionalitäten einheitlich vorgenommen.
Kommentare	Wiederverwendbare sowie erklärungsbedürftige Quellcodebestandteile sind nachvollziehbar zu kommentieren.
Fiori Design Guidelines	Die Verwendung der SAPUI5-Komponenten erfolgt im Sinne der Fiori Design Guidelines, um diesbezüglich übergreifend konsistente Anwendungen zu entwickeln.
Performance	Bei der Implementierung ist auf eine Optimierung des Ladeverhaltens zu achten, bspw. durch korrekte Definition von Dependencies sowie der Nutzung von Lazy und Asynchronous Loading.
IDs	Der Zugriff auf Implementierungsbestandteile erfolgt durch die Vergabe und Verwendung von stabilen IDs, um die Robustheit der Anwendungen sicherzustellen.

6.1.3 Regelungen zur Kompatibilität

Vorrangig sind im SAPUI5-Framework enthaltene Standardfunktionalitäten zu verwenden. Die Umsetzung von Custom Controls ist mit IT.BA.SAP gesondert abzustimmen und muss gemäß den entsprechenden SAPUI5-Vorgaben erfolgen. Fremdbibliotheken werden nicht eingesetzt.

Die SAPUI5-Bibliotheken für die Laufzeitumgebung sind vom Fiori Frontend-Server (FES) zu beziehen. Ein Verweis auf ein Content Delivery Network (CDN) oder andere Quellen ist nicht zulässig. Demzufolge ist die auf dem FES eingesetzte Version Maßgabe für die Entwicklung.

Grundsätzlich *nicht* erlaubt sind:

- Verwendung oder Überschreiben von SAPUI5-Funktionalität, welche nicht Teil der Public API ist
- Verwendung von als „experimental“ gekennzeichnete Funktionalität in Produktivumgebungen
- Verwendung von als „deprecated“ gekennzeichnete Funktionalität für Neuimplementierungen
- Manipulation von HTML/CSS via JavaScript oder direkt via CSS
- Verwendung von Custom CSS, welches Einfluss auf Theming- oder Standard-Eigenschaften hat.

6.2 Anwendungsgrundstruktur

6.2.1 SAPUI5-App

Projektname:	<hauptgebiet><untergebiet><titel> bspw. <i>hrmsszeitnachweisinbox</i> (max. 40 Zeichen)
Namespace:	de.tk

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 21 von 56

6.2.2 BSP-Applikation

Name: Z<HAUPTGEBIET><UNTERGEBIET><KURZTITEL>
bspw. ZHRMSS_ZNINBOX
(max. 15 Zeichen)

Paket: Zuordnung auf fachlicher Basis
bspw. ZHR_MSS

6.2.3 Ordnerstruktur

Mindestens:

- webapp
- controller
- i18n
- model
- view

6.2.4 Besonderheiten bzgl. Git-Versionierung und Deployment

Von der **Git-Versionierung** auszuschließen sind

- Temporäre Laufzeitobjekte
- Buildergebnisse (dist-Ordner) und IDE-Artefakte (bspw. Node-Modules)
- Persönliche Benutzereinstellungen

Vom **Deployment** auszuschließen sind

- index.html
(Ausnahme: Fiori-Apps, die ausschließlich als Component eingebettet werden)
- Mock- und Testdaten

Auszug aus ui5-deploy.yaml

```
exclude:
  - /test/
  - /localService/
  - index.html
```

6.3 Anwendungsbestandteile

6.3.1 Applikationseinstieg

Root: *App.view.xml*
App.controller.js

DataSources: *mainService*
als Bezeichnung des einzigen bzw. hauptsächlichen OData-Services

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 22 von 56

6.3.2 Views / XML-Views

View-Art	XML-Views sind den anderen View-Arten vorzuziehen
Benennung	View-Namen beginnen mit einem Großbuchstaben und enden mit *.view.xml
Ablage	Alle Views sind im View-Ordner abgelegt.
Namespaces	sap.m ist als Default Namespace zu verwenden; für weitere wird der letzte Part des SAP Namespaces als Alias verwendet
Modularisierung	Die App.view.xml wird ausschließlich als Hülle verwendet. Fachliche Anwendungsbestandteile sind in eigenen Views zu implementieren und per Routing aufzurufen. Fragmente können zur weiteren Übersichtlichkeit beitragen.
IDs	Hauptelemente einer View sind mit einer eindeutigen ID zu versehen.
Bindings / Formatter	Expression Bindings können für einfache Operationen eingesetzt werden. Komplexere Logiken sind bspw. in Formatter auszulagern.
Lazy Loading	Inhalte, die nicht direkt angezeigt werden, sind per lazy-loading zu implementieren. Asynchrones Laden: Angabe "async": true in der manifest.json

6.3.3 Controller

Benennung	Controller-Namen beginnen mit einem Großbuchstaben und enden mit *.controller.js Controller-Namen werden mit der gleichen Bezeichnung wie von der zugehörigen View versehen, sofern eine 1:1-Beziehung besteht.
Strict Mode	Für die Implementierung ist der Strict Mode zu aktivieren („use strict“;)
Modularisierung	Einzelne Module werden per sap.ui.define im globalen Namespace eingebunden und mit ihrer tatsächlichen Bezeichnung im Constructor übergeben. Wiederverwendbarer Code ist in eigene Controller auszulagern
Lazy Loading	Abhängigkeiten, die ausschließlich innerhalb der Controllerlogik bestehen, können per sap.ui.require asynchron zum relevanten Zeitpunkt nachgeladen werden.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 23 von 56

6.3.4 Component Configuration / App Descriptor

Konfiguration	Die Komponentenkonfiguration ist als Datei Component.js benannt und im wie alle anderen relevanten UI-Bestandteile der App im webapp-Ordner hinterlegt.
Anwendungszugriff	Die index.html Datei ist nur für Testzwecke zu verwenden und vom Deployment auszuschließen, da der Anwendungszugriff ausschließlich über das Launchpad erfolgt. (Ausnahme: Fiori-Apps, die ausschließlich als Component in andere Apps eingebettet werden.)
Manifest	Die Anwendungsbeschreibung ist als Datei manifest.json benannt und im webapp-Ordner hinterlegt.
i18n	Titel und Beschreibung der App sind in der i18n.properties zu hinterlegen.

6.3.5 JavaScript

„use strict“	Einsatz für hochwertigeren Code bspw. Pflicht Variablendeklaration
ESLint 6	ES6-Funktionalität darf nur nach Abstimmung mit IT.BA.SAP eingesetzt werden, da es seitens SAPUI5 noch nicht vollständig unterstützt ist. Dies gilt für die entsprechende Einstellung in der statischen Syntaxprüfung entsprechend.
Sichtbarkeit	Es ist das Prinzip „use locals, avoid globals“ einzuhalten.
Namenskonventionen	Verwendung der Hungarian Notation für Variablen und Objektbezeichnungen:

<u>Beispiel</u>	<u>Typ</u>
s Id	string
o DomRef	object
\$ DomRef	jQuery object
i Count	int
m Parameters	map / assoc. array
a Entries	array
d Today	date
f Decimal	float
b Enabled	boolean
r Pattern	RegExp
fn Function	function
v Variant	variant types

Verwendung von CamelCase.

Klassennamen beginnen mit einem Großbuchstaben.

Private Funktionen und Variablen beginnen mit einem Unterstrich.

Event-Handler-Methoden beginnen mit „on“.

Instanzvariablen werden im Konstruktor initialisiert.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmrichtlinie.docx Nur für den Dienstgebrauch		Seite 24 von 56

Kompatibilität Vermeidung von Vererbung von SAPUI5-Objekten.

Implementierung von Default-Verhalten für die Verarbeitung von Enum-Werten.

6.3.6 **Darstellung Texte, Internationalisierung, Barrierefreiheit, Geräteunabhängigkeit**

Anwendungssprache Die Anwendungstextsprache folgt der Unternehmenssprache Deutsch.

i18n Textuelle Bestandteile werden in der Datei i18n.properties hinterlegt. Es werden keine weiteren sprachspezifischen Properties-Dateien erstellt.

Die i18n-Schlüssel sind in lower camelCase definiert.

I18n-Schlüssel werden nicht verkettet eingesetzt. Bei Bedarf sind Platzhalter zu verwenden.

Spezialzeichen sind per Unicode Escape Sequence einzubinden.

Labels Oberflächenbestandteile sind mit verständlichen Labels zu versehen.

Barrierefreiheit Für visuelle Elemente sind Alternativtexte zu hinterlegen.

Geräteunabhängigkeit Anwendungen werden für die unterschiedlichen Bildschirmgrößen Desktop, Tablet, Phone optimiert.

Tablet und Phone verfügen sinnvollerweise über gleichartige Anpassungen.

Geräteabhängige Einstellungen werden anhand der sap.ui.Device API vorgenommen

Für Geräte mit Touch-Bedienung ist die Content Density „sapUISizeCozy“ in der Component.js zu setzen. (vgl. <https://experience.sap.com/fiori-design-web/v1-70/cozy-compact/>)

6.3.7 **Mocking und Maschinelle Tests**

Strukturierung Der webapp/test Ordner enthält ausschließlich nicht-produktiven Code.

MockServer-Seiten sind im Ordner webapp/test abzulegen.
JUnit Tests sind im Ordner webapp/test/unit abzulegen.
OPA5 Tests sind im Ordner webapp/test/integration abzulegen.

Mockdaten / Mockskript MockServer-Seiten enden auf MockServer.html

Mockdaten sowie das Mockscript mockserver.js sind im Ordner webapp/localService abzulegen.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 25 von 56

Zu beachten: Die metadata.xml wird bei Änderungen des OData-Services nicht automatisch aktualisiert.

QUnit

Die Implementierung erfolgt für eigene JavaScript-Funktionalitäten mit wiederverwendbarer Fachlogik

QUnit-Dateien enden auf *qunit.html

Die Testsuite wird durch die Datei unitTests.qunit.html ausgeführt.

Die Ordnerstruktur ist analog der Anwendungsstruktur zwecks Wiedererkennung aufzubauen.

Dependencies sind durch Stubs zu ersetzen, um ausschließlich die eigene Funktionalität zu testen.

OPA5

Die Implementierung erfolgt für eine Anwendung je nach Bedarf.

Zur Strukturierung werden Page Objects und Journeys mit entsprechender Ordnerstruktur eingesetzt.

6.4 OData Backend / Frontend

Modularisierung

Grundsätzlich ist eine 1:1-Beziehung zwischen UI5-App und OData-Service anzustreben.

Im OData-Service sind ausschließlich OData-spezifische Operationen sowie Datenmapping zwischen Backend und Frontend zu implementieren. Tiefergehende Fachlogik ist auszulagern zur Reduzierung der Abhängigkeiten sowie Erhöhung der Wiederverwendbarkeit.

Servicename

Z<HAUPTGEBIET>_<THEMA>_<TITEL>
bspw. ZHR_PT_ZEITNACHWEIS

bei Fremdprodukten:

Z<HAUPTGEBIET>_<FIRMA>_<ORIGINAL>
bspw. ZHR_SCD_HDY_UI5_2_SRV

Pakethierarchie

OData-Services werden in Paketen mit Suffix „_ODATA“ abgelegt

auf Backendseite:

auf Anwendungspaketkomponentenebene

z.B. Z01_11_30 ... ZHR ... ZHR_PT ... ZHR_PT_ODATA

auf Frontendseite:

im gleichen Paketpfad wie auf Backendseite

bei Fremdprodukten:

Zuordnung in fachliches Fremdprodukt-Paket

bspw. ZFI_HOAG_ODATA, ZHR_SCD_ODATA

für x42-Entwicklungen – aktuell nur Transportwesen:

ZBC_TW_ODATA

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 26 von 56

Grundsatz „Never trust the client“

Sofern notwendig sind in den Serviceimplementierungen zusätzliche Berechtigungsprüfungen einzubauen.

Daten für Änderungen sind im Backend erneut zu validieren.

6.5 GitHub

Repositories

Der Repository-Name entspricht dem SAPUI5-Anwendungs-Projektnamen, enthält ausschließlich Kleinbuchstaben, keine Sonderzeichen und max. 40 Zeichen.
bspw. *hrmsszeitnachweisinbox*

Ausnahme: Mit der Entwicklungsumgebung generierte Adaptation-/Extension-Projekte haben einen abweichenden Namensaufbau.

Branches

Jedes Repository besitzt einen Standardbranch. Dieser repräsentiert den jeweils aktuellen Deploymentstand und ist somit Ausgangsbasis für neue Entwicklungen.

Entwicklungstätigkeiten werden auf temporären Featurebranches durchgeführt.

Der Merge von einem Featurebranch auf den Standardbranch erfolgt per Pull-Request unter Berücksichtigung des Vier-Augen-Prinzips und ist spätestens vorzunehmen vor der Freigabe des zugehörigen Transportauftrags im Entwicklungssystem.

Nach Abschluss der Entwicklungstätigkeiten sind die Remote-Featurebranches zu löschen.

Pull-Requests

Nach einem Deployment ist zeitnah ein Pull-Request zu eröffnen.

Der Pull-Request-Bezeichnung ist die zugehörige SAP-Transportauftragsnummer voranzustellen.
bspw. *E12K902135 Feature Grundansicht*

Commits

Frühzeitiges und regelmäßiges Committen fördert die Nachvollziehbarkeit von Änderungen sowie möglicher Parallelentwicklungen.

Die Commit-Message enthält mind. eine Zusammenfassung der durchgeführten Anpassungen.

Durch regelmäßige Pushes auf Remote-Banches werden lokale Entwicklungsstände zentral gesichert.

6.6 Erweiterung von SAP-Standard-Apps

Extension Points

Erweiterungen von SAP-Standard-Apps werden nur in den zur Verfügung gestellten Extension Points vorgenommen.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 27 von 56

7 Einzelaspekte bei der Entwicklung

7.1 Verwendung von Log-Points und Assertions

Funktionsbausteine sind über die Benutzung von Log-Points einem Trace zuzuführen. Einzelheiten in Abstimmung mit IT.BA.SAP
Assertions sind zulässig in protokollierender Form.

7.2 Erweiterungskategorien bei Strukturen und Tabellen

TK-eigene Strukturen und Tabellen sind bezüglich der Erweiterungskategorie immer mit "nicht erweiterbar" zu kennzeichnen.

7.3 Verwendung von Validierungen und Substitutionen

7.3.1 Validierungen

In Validierungen dürfen KEINE Datenänderungen und KEINE Datenbankänderungen vorgenommen werden.

7.3.2 Substitutionen

Für Substitutionen gilt:

- die durch Substitution änderbaren Felder werden über die Dokumentation zur Substitution festgelegt. Darüberhinausgehende Änderungen oder Datenbankänderungen sind im Einzelfall abstimmungspflichtig
- Es darf KEINERLEI Coding vorgenommen werden, was COMMIT-Befehle auslöst.

7.4 Verwendung von logischen Dateinamen

Die Vergabe von Dateinamen erfolgt zweistufig.

Ein einheitlicher Präfix wird über die Transaktion FILE im SAP gepflegt
Der weitere Aufbau kann im Programm hinterlegt werden

7.4.1 Logischer Pfadname (Präfix)

Die Logischen Pfadnamen haben folgenden Aufbau:

Z_	Art	Bereich	Suffix für Firma
1-2	n Zeichen	1 n Zeichen	0 oder 4 Zeichen

Aufbau des Suffix für eine Firma:

F	Nummer der Firma (siehe Domäne Z_TW_FIRMA_NR)
1-2	3 -4

Beispiele:

Z_FT_SYS	Verzeichnis für ein SAP-System
Z_FT_SAP_F01	Verzeichnis für alle SAP-Systeme der Firma TK
Z_FT_SAP_F02	Verzeichnis für alle SAP-Systeme der Firma TK-GmbH

7.4.2 Bildung des Pfad- / Dateinamen im Programm

Bei der Ermittlung von Namen anhand von logischen Pfadnamen ist wie folgt vorzugehen:

*	+-----
*	! Klassen
*	+-----

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 28 von 56

```

CLASS: zcl_file      DEFINITION LOAD.
*                   +-----+
*                   ! Variablen
*                   +-----+
DATA: gv_path        TYPE char255.
DATA: go_file        TYPE REF TO zcl_file.

```

...

```

*                   +-----+
*                   ! Ereignisse
*                   +-----+
LOAD-OF-PROGRAM.
CREATE OBJECT go_file EXPORTING i_logical_path = 'logischer_Pfadname'.

gv_path          = go_file->get_name('Text').

```

Anstelle von '**logischer_Pfadname**' ist der jeweilige logische Pfadname anzugeben (ohne Suffix für eine Firma)

Anstelle von '**Text**' ist die jeweilige Ergänzung anzugeben (kann auch ganz entfallen)
folgende Symbole werden bei 'Text' ersetzt

<Sysid>	wird durch das aktuelle System ersetzt	z.B. E11
<Mandt>	wird durch den aktuellen Mandanten ersetzt	z.B. 010
<Firma>	wird durch die aktuelle Firma ersetzt	z.B. 01 für TK
<Date>	wird durch das aktuelle Datum ersetzt	z.B. 20030501
<Time>	wird durch die aktuelle Uhrzeit ersetzt	z.B. 151255

gv_path enthält danach den erzeugten Pfadnamen

7.5 Konventionen bei der Kommunikation via EAI-Schicht

Die Kommunikation erfolgt über remotefähige Funktionsbausteine. Diese haben in einer Funktionsgruppe zu liegen,

- die mit ZMQ_TKEASY_ beginnt (Richtung SAP nach TKeasy)
- die mit ZMQ_SAP_ beginnt (Richtung TKeasy nach SAP).

Die verwendeten Bezugstypen dürfen sich nur auf Strukturen / Datentypen beziehen, die mit ZMQ_ beginnen. Bei der Definition dieser Datentypen dürfen nur primitive Datentypen bzw. wieder Datentypen mit dem Präfix ZMQ_ verwendet werden.

Die Verwendung der Datentypen ZMQ_ und ZEAI_ darf nur in Z_MQ_ Bausteinen oder direkt vor oder nach dem Aufruf von Z_MQ_ Bausteinen (für das Mapping internes, externes Format) erfolgen.

Änderungen an den Schnittstellen der mit ZMQ_ beginnenden Funktionsbausteinen sind immer mit IT.BA.SAP abzustimmen.

Spezielle Datentypen:

- für Datum ist immer der primitive Datentyp DATS zu verwenden,
- für Boolean sind die Datentypen ZEAI_BOOL_0_1 oder ZEAI_BOOL_SPACE_X zu verwenden,
- für den Rückgabewert E_RETURN ist immer der Datentyp ZMQ_RETURN bzw. ZMQ_RETURN_SAP zu verwenden.

Die in den Templates (s. 12.8) vorhandenen Vorgaben für den Aufbau und die Strukturierung der Funktionsbausteine sind einzuhalten.

Funktionsbausteine, die Logik im SAP aufrufen:

- beginnen mit Z_MQ_SAP_

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 29 von 56

- das Mapping zwischen interner Struktur(SAP) und externer Struktur(EAI-Schicht) wird von genau einem TK-eigenen Funktionsbaustein durchgeführt. Dieser Funktionsbaustein enthält keine weitere Verarbeitungsfunktionalität
- für die Verarbeitung wird von diesem Funktionsbaustein genau ein TK-eigener Funktionsbaustein, der die Verarbeitung durchführt, aufgerufen

Funktionsbausteine, die Logik im TKeasy aufrufen:

- beginnen mit Z_MQ_TKEASY_
- besitzen den Rückgabeparameter E_RETURN vom Typ ZMQ_RETURN (bei synchroner Verarbeitung)
- ein Verarbeitungstyp im TKeasy (= 1 Funktionsbaustein beginnend mit Z_MQ_TKEASY) wird von genau einem TK-eigenen Funktionsbaustein aufgerufen, der neben einer Verarbeitung das Mapping zwischen interner Struktur(SAP) und externer Struktur (EAI-Schicht) ausführt.

Die lokale Implementierung enthält den Aufruf der Funktion Z_CA_LIST_PARAMETER, die bei lokaler Ausführung die Daten zu Kontrollzwecken in den Spool stellt

Bei Funktionsbausteinen wird die Lokale Implementierung über die Transaktion ZCA_GEN_01 generiert. (Die Option Generierung erzwingen darf nur zu Testzwecken verwendet werden).

Teilauftrags-Abarbeiter:

Es ist in jeden Z_AB_* Bausteinen zu prüfen, ob die Struktur unterstützt wird. Bei einer unbekannten Struktur ist ein Verarbeitungsfehler zu raisen.

Wenn aus Z_AB_* Bausteinen Exceptions per Raise gesetzt werden, so ist auch immer die Return-tabelle T_RETURN mit einer Meldung zu füllen, damit man im Auftragsmonitor den Grund für den Fehler nachvollziehen kann.

Dies gilt insbesondere für *VERARBEITUNGS_FEHLER*, *WARTET_AUF_EREIGNIS*.

Meldungskonvertierung mit Referenz für Verwendungsnachweis

In Bausteinen, die über den Auftragsmonitor abgearbeitet werden, sollen generalisiert bei Messages folgendes Vorgehen gewählt werden:

```
MESSAGE e004 (zcd) WITH 'Z_MQ_SAP_CD_NIE_ERL_ANZUORDNEN'
                        'lv_betrw-anord'
                        INTO lv_return-message.

CALL FUNCTION 'Z_CA_CONVERT_MSG'
EXPORTING
  i_msg_syst      = syst
IMPORTING
  e_msg_bapiret2 = lv_return
EXCEPTIONS
  OTHERS         = 0.
```

So wird dann die t_return Tabelle mit den Fehlertexten korrekt gefüllt.

7.6 Sonderregeln für die Funktionalität in Verbindung mit TKeasy-Wrapper-Gui

Funktionsbausteine und Klassen, die über ein TKeasy-Wrapper-Gui eingesetzt werden, müssen besonderen Anforderungen genügen. Dies gilt auch für die Entwicklung von User-Exits.

Im Dokument X:\TK SAP R3\01 R3 (Schnittstellen)\9 TK Dokumentation\Richtlinien\FAQ SAP-Services in der TK.doc sind Fragestellungen hierzu aufgeführt.

Besondere Beachtung liegt in dem Einsatz von Konstrukten PERFORM ON-COMMIT, PERFORM ON_ROLLBACK, lokale Verbuchung. Grundsätzlich nicht zulässig ist ein eigener COMMIT in solchen Konstrukten.

Wegen der erhöhten Komplexität sind solche Implementierungen nur in enger Abstimmung mit IT.BA.SAP auszuführen.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 30 von 56

7.7 Sprachkompatibilität

Die Kompatibilität zu ABAP OBJECTS muss beachtet werden. Zu beachten sind die unter ABAP OBJECTS unzulässigen Sprachelemente (s. SAP-Doku ABAP - ABAP Objects - Anweisungen - Ablösung obsoleter Sprachelemente). Diese dürfen nicht verwendet werden.

Alte Sourcen werden angepasst bei Wartungs- oder Weiterentwicklungsaktivitäten.

7.8 UNICODE-Kompatibilität

Alle Entwicklungen müssen UNICODE-kompatibel sein. Die Unicode-Prüfung in den Moduleigenschaften muss aktiv sein.

7.9 Lösungsverfahren

Für TK-eigene Tabellen sind Lösungsverfahren zu implementieren, wenn diese Anwendungsdaten enthalten. Anwendungsdaten sind Daten, die nicht der reinen Ablaufsteuerung dienen (z.B. Customizing-tabellen). Lösungsverfahren stellen sicher, dass die Daten aus der Tabelle gelöscht werden, wenn sie nicht mehr benötigt werden. Die TK ist verpflichtet, nicht mehr benötigte Daten zu löschen. Vorgaben für Löschfristen und Löschumfang sind von der die Anwendung nutzenden Fachabteilung einzuholen und zu dokumentieren.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 31 von 56

8 Wiederverwendbarkeit

8.1 Funktionsbausteine

Es werden Funktionsbausteine verwendet, wenn ein Codingteil mehr als zweimal von unterschiedlichen Programmen aufgerufen wird.

Funktionsgruppen unterliegen einer gemeinsamen Syntaxprüfung und besitzen gemeinsamen Datenbereich. Daher sollten nur Funktionsbausteine in einer Funktionsgruppe liegen, die inhaltlich zusammengehören. Es wird in kleine Funktionsgruppen aufgeteilt (≤ 10 Funktionsbausteine).

Für die Generierung von Pflege-Views (Tabellenpflegegenerator) ist eine eigene Funktionsgruppe zu verwenden.

8.2 Forms

Ein Aufruf externer Forms in der Form "PERFORM FORMXXX(YYY) / PERFORM from IN PROGRAM prog" ist nicht zulässig. Folgende Ausnahmen sind zugelassen:

- Benötigte Forms aus SAP-Standardprogrammen, die nicht als Funktionsbaustein implementiert sind. Jede Benutzung ist im Vorhinein abzustimmen.
- Struktur-Check für Auftragsmonitor- Abarbeiter.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 32 von 56

9 Datenbankrelevantes Coding (SAP HANA)

Die TK verwendet für die SAP-Systeme die Datenbank SAP HANA. Jegliches Coding muss die HANA-spezifischen Vorgaben beachten. Für spezielle HANA-Features dürfen alle Funktionsteile verwendet werden, die über die ABAP-Workbench erreicht werden können. Jegliche Erstellung solchen Codings muss vorab mit IT.BA.SAP abgestimmt werden.

Bei TK werden eingesetzt:

- Open SQL
- Core Data Services (CDS)
- ABAP managed Database Procedures (AMDP)

Bei TK ist nicht zulässig;

- ADBC mit Native SQL. Nach vorheriger Abstimmung kann ADBC für den Aufruf von HANA-Prozeduren eingesetzt werden.

Verwendung Open SQL ab ABAP 7.4 SP2:

Zu beachten:

- Damit die neuen Funktionen verwendet werden können, muss bei diesen Statements
- Hostvariablen das Fluchtsymbol @ vorangestellt werden
- Die Selektionsliste muss Komma separiert angegeben werden

Die neue SQL-Syntax sollte nur Verwendung finden, wenn die "erweiterte SQL-Funktionalität" verwendet wird. Alte SQL-Statement bzw. die Verwendung der "normalen SQL-Funktionalität" erfolgt nach wie vor in der vereinfachten SQL Syntax (ohne Fluchtsymbole und Kommaseparierungen).

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 33 von 56

10 Sicherheitsanforderungen

10.1 Grundsatz

Jede Entwicklungstätigkeit für die SAP-Produktionssysteme der TK hat einschlägige Sicherheitsanforderungen gemäß dem Stand der Technik zu erfüllen. Alle nachfolgenden Gliederungspunkte gehen auf potentielle Sicherheitsrisiken ein und sind zu behandeln. Die Punkte sind abgeleitet aus den Sicherheitsempfehlungen der SAP sowie aus Empfehlungen einschlägiger Leitfäden (z.B. DSAG / BSI). In den folgenden Gliederungspunkten sind TK-spezifische Vorgehensweisen beschrieben und Handlungsvorgaben dokumentiert.

Immer untersagt sind jegliche Implementierungen in den SAP-Produktionssystemen, mit deren Hilfe Zugang zu Daten aus den Umfeldern SGB, Personaldaten, Rechnungslegung, Zahlungsverkehr sowie Daten, die den Bestimmungen des BDSG bzw. der DSGVO unterliegen, erlangt werden kann, ohne dass dies in der Entwicklungsvorgabe spezifiziert und durch eine fachliche Freigabe bestätigt ist. Ebenfalls sind Implementierungen untersagt, durch die bei der Ausführung höhere Berechtigungen im SAP-Produktionssystem erlangt werden können, als durch die zugewiesenen Rollen des ausführenden Users eingeräumt wird. Ausnahme, z.B. für die Realisierung von Vertretungsregelungen, müssen einzeln vor Umsetzung genehmigt werden.

10.2 Maschinelle Prüfung

TK verwendet zur Prüfung der Einhaltung von Sicherheitsanforderungen das Werkzeug Code Profiler der Firma Virtual Forge. Werden von diesem Werkzeug sicherheitsrelevante Findings ausgeworfen entsprechend der im zur Coding-Prüfung gehörigen Dokument festgelegten Schwere, ist dies als ein Programmfehler zu werten. Das Objekt darf nicht in ein Produktionssystem transportiert werden und das Coding ist zu ändern oder entsprechend der in einem bei IT.BA.SAP vorgehaltenem Dokument genannten Vorgehensweise zu mitigieren.

10.3 Datenstrukturen

10.3.1 Tabellenberechtigungsgruppe

Für jede eigenerstellte Tabelle und jeden View ist explizit beim Auftraggeber der Entwicklung der Schutzbedarf für die Anbringung einer Tabellenberechtigungsgruppe abzufragen und zu dokumentieren. Wird vom Auftraggeber keine Tabellenberechtigungsgruppe gefordert, ist das Objekt mit ZTK als Berechtigungsgruppe zu versehen.

Für alle intern bei IT.BA.SAP benötigten Objekte wird die Berechtigungsgruppe auf ZFI1 gesetzt, sofern keine anderen Anforderungen vorgegeben sind.

10.3.2 Tabellenprotokollierung

Für jede eigenerstellte Tabelle ist die Protokollierung von Datenänderungen zu aktivieren (Dictionary - Technische Einstellungen - Datenänderungen protokollieren).

Ausgenommen hiervon sind folgende Tabellentypen:

- ergänzende Tabellen zu Standard-SAP-Tabellen, wenn an den SAP-Tabellen selbst keine Datenänderungen protokolliert werden (Beispiel: ZCD_BUT000 zu BUT000),
- Tabellen, die von ihrer Natur her selbst Protokolltabellen sind (Beispiel ZPBAVPROT_PA0202),
- Tabellen, die von ihrer Natur her nur kurzfristig verwendete Datenscheiben insbesondere zur technischen Kommunikation zwischen verschiedenen EDV-Systemen dienen (Beispiel ZCDDWHGP).

Eine Einzelfallbewertung bezüglich der Aktivierung der Protokollierung hat zu erfolgen, wenn die Tabelle auf mehr als 100.000 Einträge ausgelegt und eine Updaterate von mehr als 10%/Jahr zu erwarten ist.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 34 von 56

Wird keine Protokollierung von Tabellenänderungen nach obiger Maßgabe vorgenommen, ist in Ausnahmefällen bei kritischen Datenfeldern bezüglich Rechnungslegung, Personaldaten oder SGB-Daten eine Protokollierung auf Datenelementebene vorzusehen.

Der Sachstand, dass für eine Tabelle keine Änderungsprotokollierung angeschaltet werden soll, wird mit &KP& in der Kurzbeschreibung angezeigt.

10.3.3 Erstellung von Pflegeviews für Tabellen

Für Pflegeviews ist pro Pflegeview eine individuelle Funktionsgruppe zur Aufnahme der viewspezifischen Funktionsbausteine zu definieren. Der Name der Funktionsgruppe soll den Namen der Tabelle/des Views, auf den sich der Pflegeview bezieht, widerspiegeln.

10.4 Programmstrukturen

10.4.1 Berechtigungsprüfung bei Transaktionen

Bei Verwendung der Aufrufart CALL TRANSACTION ist vorher die Berechtigung via Funktionsbaustein AUTHORITY_CHECK_TCODE zu prüfen. Ergibt diese Prüfung fehlende Berechtigung, darf der Aufruf CALL TRANSACTION nicht ausgeführt werden.

10.4.2 Hart codierte Berechtigungen / Hintertüren / fest codierte Benutzerkennungen

Die Ausführung von Coding in Abhängigkeit einer fest codierten Benutzerkennung ist grundsätzlich untersagt incl. Auslesen von Benutzerkennungen aus Tabellen oder Usereingaben einer Benutzerkennung zum Ausführen von davon abhängigem Coding. Für jegliche betriebsnotwendige Ausnahmen ist für den Einzelfall eine Genehmigung einzuholen.

Die Verwendung des Funktionsbausteins RH_AUTHORITY_CHECK_OFF ist untersagt. Die Verwendung des Schaltelements AUTHORITY_CHECK_ACTIVE = OFF ist verboten.

10.4.3 Berechtigungsprüfungen in RFC-fähigen Funktionen

Grundsätzlich sind Funktionsbausteine nur bei festgestelltem RFC-Bedarf als RFC-fähig in den Eigenschaften zu kennzeichnen. Die Berechtigungsprüfung erfolgt dann im Regelfall über den Namen der Funktionsgruppe im Berechtigungsobjekt S_RFC.

Bei erhöhtem Schutzbedarf (z.B. Zugriff auf Sozialdaten/Personaldaten, Ausführung einer Anordnung, Ausführung einer Zahlungsfunktion) ist zusätzlich am Anfang via fachlichem Berechtigungsobjekt und AUTHORITY-CHECK die Berechtigung des Users zur Ausführung explizit zu prüfen. Wird fehlende Berechtigung ermittelt, hat die weitere Ausführung zu unterbleiben.

10.4.4 Berechtigungsprüfungen in Reports

Der Reportschutz erfolgt im Minimum durch die Zuordnung einer Berechtigungsgruppe in den Eigenschaften des Programms. Vorgaben für die Werte s. 16.6.

Bei erhöhtem Schutzbedarf (z.B. Zugriff auf Sozialdaten/Personaldaten, Ausführung einer Anordnung, Ausführung einer Zahlungsfunktion) ist zusätzlich am Anfang via fachlichem Berechtigungsobjekt und AUTHORITY-CHECK die Berechtigung des Users zur Ausführung explizit zu prüfen. Wird fehlende Berechtigung ermittelt, hat die weitere Ausführung zu unterbleiben.

10.4.5 Debug-Code (z.B. in ASSERT-Statements)

Die Verwendung einer ASSERT- oder ASSERT ID-Anweisung, deren logischer Ausdruck immer falsch ist, ist untersagt.

Die Ausführung von Berechtigungsprüfungen oder sicherheitsrelevanten Werteprüfungen abhängig von ASSERT-Bedingungen ist untersagt.

10.4.6 Generischer und dynamischer ABAP-Code

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 35 von 56

Bei der Verwendung von dynamischen ASSIGN- oder WRITE-Befehlen dürfen keine Benutzereingaben direkt verwendet werden. Sie müssen immer auf im Programm fest codierte Zeichenketten übersetzt werden.

Die Verwendung generischen und dynamischen Codings ist grundsätzlich zu vermeiden. Ausnahmen sind zu begründen und zu genehmigen.

10.4.7 Generische Funktionsaufrufe

Es ist untersagt, generische oder dynamische Aufrufe einer Funktion über eine direkte Benutzereingabe auszuführen. Auf die Ausführung von generischen oder dynamischen Funktionsaufrufen ist im Regelfall zu verzichten. Im Ausnahmefall muss der generische Aufruf sich aus fest codierten Zeichenketten im rufenden Modul heraus bilden. Genehmigung ist erforderlich.

10.4.8 Generische Reports / Programme (ABAP Command Injection)

Die Verwendung der dynamischen Generierung von Reports oder Programmen ist grundsätzlich untersagt. Jegliche einzeln zu begründende Ausnahme ist im Rahmen einer technischen Spezifikation genehmigungspflichtig.

10.4.9 Generische Berechtigungen

Die Verwendung des Parameters USER in den Befehlen SUBMIT und CALL TRANSACTION ist untersagt.

10.4.10 Generische RFC Destination

Generische Destinationen dürfen nicht direkt aus einer Usereingabe übernommen werden. Die Werte sind immer auf Gültigkeit gegen Zeichenketten zu prüfen oder auf zugelassene Destinations zu übersetzen. Die Default-Destination kann durch Aufruf der Methode ZCL_TW_MANDT->GET_RFC_DESTermittelt werden.

10.4.11 SQL-Injection / Forceful querying / Generic Table query

Benutzereingaben, die zu Aktionen auf Datenbanktabellen führen, sind immer gegen zulässige Zeichenketten zu prüfen bezogen auf dynamische Elemente eines SQL-Statements und Inhalte von Variablen. Übergebene Parameter an Funktionsbausteine oder Klassen, die zu Aktionen auf Datenbanktabellen führen, sind bezüglich dynamischer Elemente eines SQL-Statements gegen zulässige Zeichenketten zu prüfen. Im Fehlerfall darf die Aktion nicht ausgeführt werden.

10.4.12 Directory Traversal

Zur Verhinderung von Directory Traversals sind folgende Maßnahmen zu treffen:

1. Ein an einen Report übergebener Parameter zu einem Filenamen und/oder Pfad ist vor Verwendung mit der Methode `zcl_file->get_name` zu prüfen. Erfolgt eine Fehlermeldung, ist die Eingabe/Übergabe mit Fehlermeldung abzuweisen.
2. Ein an einen Funktionsbaustein / Methodenklasse übergebener Filename und/oder Pfad ist vor Verwendung mit der Methode `zcl_file->get_name` zu prüfen. Erfolgt eine Fehlermeldung, ist der aufgerufene Baustein mit einer entsprechenden Rückmeldung an den Aufrufer zu beenden.

10.4.13 Aufrufe in den Kernel

Direkte Aufrufe von SAP-Kernel-Funktionen sind grundsätzlich untersagt. Zu begründende Ausnahmen sind genehmigungspflichtig.

10.4.14 System Command Injection / Execution

Die Ausführung von Systemkommandos soll nur in Ausnahmefällen verwendet werden, die zu genehmigen sind. Eine Ausführung ist nur über die Benutzung der logischen Systemkommandos zulässig.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 36 von 56

sig, die über die Klasse ZCL_SYSTEM erreichbar sind. Benutzereingaben und via Parameter übergebene Werte zu den Kommandos sind vor der Ausführung auf Zulässigkeit zu prüfen.

10.4.15 Fest codierte Mandantenprüfung / Systemprüfung

Direkte Prüfungen auf Mandanten und Systeme sind untersagt.

Prüfungen auf Systeme und Mandanten dürfen nur indirekt über logische Bedingungen mittels der Methoden der Klasse ZVG_SYSTEMINFO ausgeführt werden.

Die Geschäftslogik darf nicht so konstruiert werden, dass abhängig von System- oder Mandantenkennungen Teile der Geschäftslogik nicht ausgeführt werden. Die Prüfung auf System- und Mandantenkennung ist üblicherweise auf den Zweck begrenzt, eine Ausführung der gesamten Funktionalität in einem System oder Mandant zu verhindern.

Parametrierbar aufrufbare Bausteine (z.B. Methoden oder Funktionsbausteine) dürfen solche Konstrukte nicht enthalten, es sei denn, sie sind genehmigt.

10.4.16 Mandantenübergreifender Datenzugriff

Ein mandantenübergreifender Datenzugriff ist grundsätzlich untersagt. Zu begründende Ausnahmen sind genehmigungspflichtig.

10.4.17 Direkte Datenbankänderungen

Eine unmittelbar zum Datenbankzugriff zugehörige Berechtigungsprüfung ist im Regelfall nicht vorgesehen.

In jedem Coding mit änderndem Zugriff auf Datenbanken sind allerdings Berechtigungsprüfungen vorzusehen, die die Prüfung gegen fachliche Berechtigungsobjekte vorsehen, die in den Kontext der verwendeten Datenbanktabellen gehören (Prüfung gegen Fachrollen).

10.4.18 Datenbankzugriff mit EXEC SQL

Verwendung von EXEC SQL ist eine genehmigungspflichtige Ausnahme. Zugriffe dieser Art sind in eigene Funktionsbausteine zu kapseln.

10.4.19 Dateidownload /-upload

Alle verwendeten Pfade und Dateinamen sind vor Ausführung des Download auf zugelassene Zeichenketten zu überprüfen mit der Methode `zcl_file->get_name`. Schlägt die Prüfung fehl, hat der Download oder Upload zu unterbleiben

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 37 von 56

11 Namenskonventionen

11.1 Grundlegende Regeln

Bei allen Namenskonventionen sind die SAP-Vorgaben nach Hinweis 16466 zu beachten. Werden im Rahmen von Modifikation eigene Objekte eingebracht, müssen diese im Kundennamensraum liegen.

11.2 Namenskonvention für Entwicklungsklassen

Die Systematik und die Ausprägungen für Entwicklungsklassen sind in folgendem Dokument abgelegt: [X:\TK SAP R3\01 R3 \(Schnittstellen\)\9 TK Dokumentation\Konzepte Soll\CA-KO-Entwicklungsklassen.doc](X:\TK SAP R3\01 R3 (Schnittstellen)\9 TK Dokumentation\Konzepte Soll\CA-KO-Entwicklungsklassen.doc)

11.3 Namenskonvention für Domänen

Domänennamen dürfen nicht modul- oder produktspezifisch sein.

11.4 Namenskonvention für Codepages

TK-eigene Codepages werden systemübergreifend nummeriert. Sie heißen mit 91nn.

Werteliste:

9100 : Codepage zur Abbildung des im SGB zulässigen Zeichensatzes.

11.5 Namenskonvention für Typdefinitionen

Typen haben mit t_ zu beginnen und mit _tab zu enden, wenn es sich um Tabellentypen handelt.

Beispiel:

```

TYPES: BEGIN OF t_neuer_typ
           ,satz_art           TYPE c LENGTH 2
           ...
           ,anzahl            TYPE i
       ,END OF t_neuer_typ
       ,t_neuer_typ_tab      TYPE STANDARD TABLE OF t_neuer_typ.

```

11.6 Namenskonvention für Nachrichtenklassen und Nachrichten / Syslog-Meldungen

Z	Kennung
1	2

gültige Klassen s. 16.3

Die Nachrichten werden innerhalb der Klasse fortlaufend durchnummeriert.

Syslog-Meldungen:

Z	Kennung
1	1

gültige Klassen s. 16.3

11.7 Namenskonvention für Funktionsgruppen

Z	Hauptgebiet	sprechender Name
1	2	1 max. 24 Zeichen

Hauptgebiet s. 16.5

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 38 von 56

11.8 Namenskonvention für Funktionsbausteine

Anmerkung: Im Umfeld der RFC-fähigen Funktionsbausteine gibt es aus historischen Gründen eine Reihe von Bausteinen, die die Konvention nicht erfüllen.

11.8.1 Allgemeine Funktionsbausteine

Z		Hauptgebiet		sprechender Name
1	1	2	1	max. 24 Zeichen

Hauptgebiet s. 16.5

11.8.2 Funktionsbausteine für RFC von SAP-System zu SAP-System

Z		RFC		Hauptgebiet		sprechender Name
1	1	RFC	1	HR	1	Text

Hauptgebiet s. 16.5

11.8.3 Funktionsbausteine zur Ansteuerung von TKeasy via MQ

Z		MQ		TKEASY_sprechender Name
1	1	MQ	1	TKEASY_Text

Hauptgebiet s. 16.5

11.8.4 Funktionsbausteine zur Ansteuerung von SAP via MQ

Z		MQ		TKEASY_sprechender Name
1	1	MQ	1	TKEASY_Text

11.8.5 Funktionsbausteine zur Ansteuerung von Business-Connector-Services

Für Funktionsbausteine, welche mit dem SAP Business-Connector kommunizieren gilt folgende spezielle Konvention:

Z		BX		Funktion d. Service
1	1	2	1	max. 11 Zeichen

11.9 Namenskonvention für Includes

Z	IK	Hauptgebiet		sprechender Name
1	2-3	2	1	max. 24 Zeichen

Ausnahmen: Für Includes, die in User-Exits aus den SAP-Bausteinen heraus angesprochen werden, liegen die Namen fest. Sie beginnen immer mit ZX.

11.10 Namenskonvention für Reports

Aufbau der Reportnamen

Z	Hauptgebiet	Untergebiet		sprechender Name
1	2	2	1	max. 24 Zeichen

Stelle 1: Z

Hauptgebiet / ggf. Untergebiet s. 16.5

Rest frei vergebbar, sprechender Name

11.11 Namenskonvention für FORM-Routinen

Der Name der FORM-Routine soll seine Funktion verdeutlichen.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 39 von 56

11.12 Namenskonvention für ENHANCEMENT-Points

Der Name des ENHANCEMENT-Points wird vorzugsweise nach dem Inhalt des Objekts, alternativ nach dem Name des Objekt, in dem der Enhancement Point befindet, gebildet.

11.13 Namenskonvention für programminterne Datendeklarationen

11.13.1 Grundsatz

Der Name der Datendeklaration muss die fachliche Bedeutung erklären.
Die Einhaltung der Präfixkonvention ist über den Source-Code-Inspector(SCI) prüfbar.

11.13.2 Variablen

Gültigkeitsbereiche:

Durch Angabe eines entsprechenden Präfixes soll der Gültigkeitsbereich und der Typ einer Variable gekennzeichnet werden.

g = Gültigkeit global
l = Gültigkeit lokal
u = Gültigkeit gemäß Übergabeparameter

Variablentypen:

t = Tabellen
v = Variablen
o = Objekte
if = Interface
cl = Klasse

Beispiel.:

```
FORM pernr-relevant      USING uv_zpernr uv_relevant.
data: lv_alter           type i.

uv_relevant              = k_nein.
lv_alter                  = sy-datum - uv_zpernr-gebdat.

IF lv_alter > gv_mindest_alter.
  uv_relevant             = k_ja.
endif.
ENDFORM.
```

11.13.3 Konstanten

Konstanten erhalten den Präfix k_.

Beispiel.:

```
CONSTANTS: k_woknz_west   LIKE zpernr-woknz VALUE 'W',
           k_woknz_ost     LIKE zpernr-woknz VALUE 'O'.
```

11.13.4 Variablen aus dem Selektionsbild

Präfix p = Parameter
Präfix s = Select-Options

11.14 Namenskonvention für Parameter von Funktionsbausteinen / Methoden

Art der Übergabe:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 40 von 56

Durch Angabe eines entsprechenden Präfixes ist die Art der Übergabe eines Funktionsbausteins bzw. einer Methode kenntlich zu machen.

i_ = Importing
it_ = Importing Tabellen
e_ = Exporting
et_ = Exporting Tabellen
c_ = Changing
r_ = Returning

Beispiel:

```
CLASS zvg_bmag DEFINITION.

    METHODS is_azubi IMPORTING value(i_bmagr) TYPE zbmagr
               RETURNING value(r_bool) TYPE zbool.

ENDCLASS.
```

11.15 Namenskonventionen für Funktionsbausteine zu Zeitpunkten im FS-CD

Funktionsbausteine zu Zeitpunkten im FS-CD tragen in Ihrem Namen das Anwendungsgebiet, in dem der Zeitpunkt vorkommt sowie die Zeitpunktsnummer. Sie beginnen mit Z_CD_EXIT_.

Beispiel:

Z_CD_EXIT_ABSTIMMSCHL_1113: Zeitpunkt 1113, Festlegen Vorschlagswert Abstimmungsschlüssel

Z_CD_EXIT_ZAHLLAUF_0670: Zeitpunkt 0670 im Zahllauf

11.16 Namenskonventionen für Funktionsbausteine zu allgemeinen Exits im FS-CD

Funktionsbausteine zu allgemeinen Exits im FS-CD beginnen mit Z_CD_EXIT_ und bestehen aus Teilen des Namens des SAMPLE-Funktionsbaustein, ggf. ergänzt um das Anwendungsgebiet

Beispiel:

Name des Sample-Bausteins: FKK_SAMPLE_CHECK_DOC_PAYED

Name des eigenen FuBa: Z_CD_EXIT_FKK_CHECK_DOC_PAYED

11.17 Namenskonventionen für objektorientierte Entwicklung

11.17.1 Interfaces

Die Interfacenamen haben den Präfix **ZIF_**.

11.17.2 Klassen

Die Klassennamen haben den Präfix **ZCL_**.

Lokale Klassen

Die Namen der lokalen Klassen haben den Präfix **LCL_**.

Besonderheit für Testklassen:

Mit dem Zusatz „for testing“ versehene Klassen

haben den Präfix **ZCT_** (globale Testklassen) bzw. **LCT_** (lokale Testklassen).

11.17.3 Exception-Klassen

Namen der Exception-Klassen haben den Präfix **ZCX_**.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 41 von 56

11.17.4 Instanz-Variablen

Die Instanz-Variablen beginnen immer mit einem **m** oder mit einem **o** gefolgt vom zweiten Buchstaben, der für in Abhängigkeit vom Typ der Variabel wie folgt lautet:

Präfix	Typ
mv_, ov_	ABAP-Datentypen und Strukturen
mt_, ot_	Tabellentypen
mo , oo	Objektreferenzen

11.17.5 Klassen-Variablen

Die Klassen-Variablen beginnen immer mit einem **c** gefolgt vom zweiten Buchstaben, der für in Abhängigkeit vom Typ der Variabel wie folgt lautet:

Präfix	Typ
cv_	ABAP-Datentypen und Strukturen
ct_	Tabellentypen
co_	Objektreferenzen

11.17.6 Methoden

Für das Coding, das in der Ausimplementierung der Methoden einer Klasse geschrieben wird, gelten die Namenskonventionen aus den vorhergehenden Kapiteln.

11.17.6.1 Importing-Parameter

Importing-Parameter haben den Präfix **i_** für alle Typen außer Tabellentypen. Die Tabellentypen haben den Präfix **it_**.

11.17.6.2 Exporting-Parameter

Exporting-Parameter haben den Präfix **e_** für alle Typen außer Tabellentypen. Die Tabellentypen haben den Präfix **et_**.

11.17.6.3 Changing-Parameter

Changing-Parameter haben den Präfix **c_** für alle Typen außer Tabellentypen. Die Tabellentypen haben den Präfix **ct_**.

11.17.6.4 Returning-Parameter

Returning-Parameter haben den Präfix **r_** für alle Typen außer Tabellentypen. Die Tabellentypen haben den Präfix **rt_**.

11.17.6.5 Getter, Setter

Für den ggf. gewünschten Zugriff auf die Membervariablen sind Getter- und Settermethoden zu implementieren. Die Methodennamen sind wie folgt zu benennen:

Membervariable	Getter-Methode	Setter-Methode
----------------	----------------	----------------

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 42 von 56

mv_name	get_name	set_name
---------	----------	----------

11.17.6.6 Status-Abfrage

Für die Abfrage von boolschen Membervariablen sind Statusabfrage-Methoden zu implementieren.
Die Methodennamen sind wie folgt zu benennen:

Membervariable	Methode
mv_persistiert	is_persitiert

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 43 von 56

11.18 Namenskonventionen für das XFT-Framework

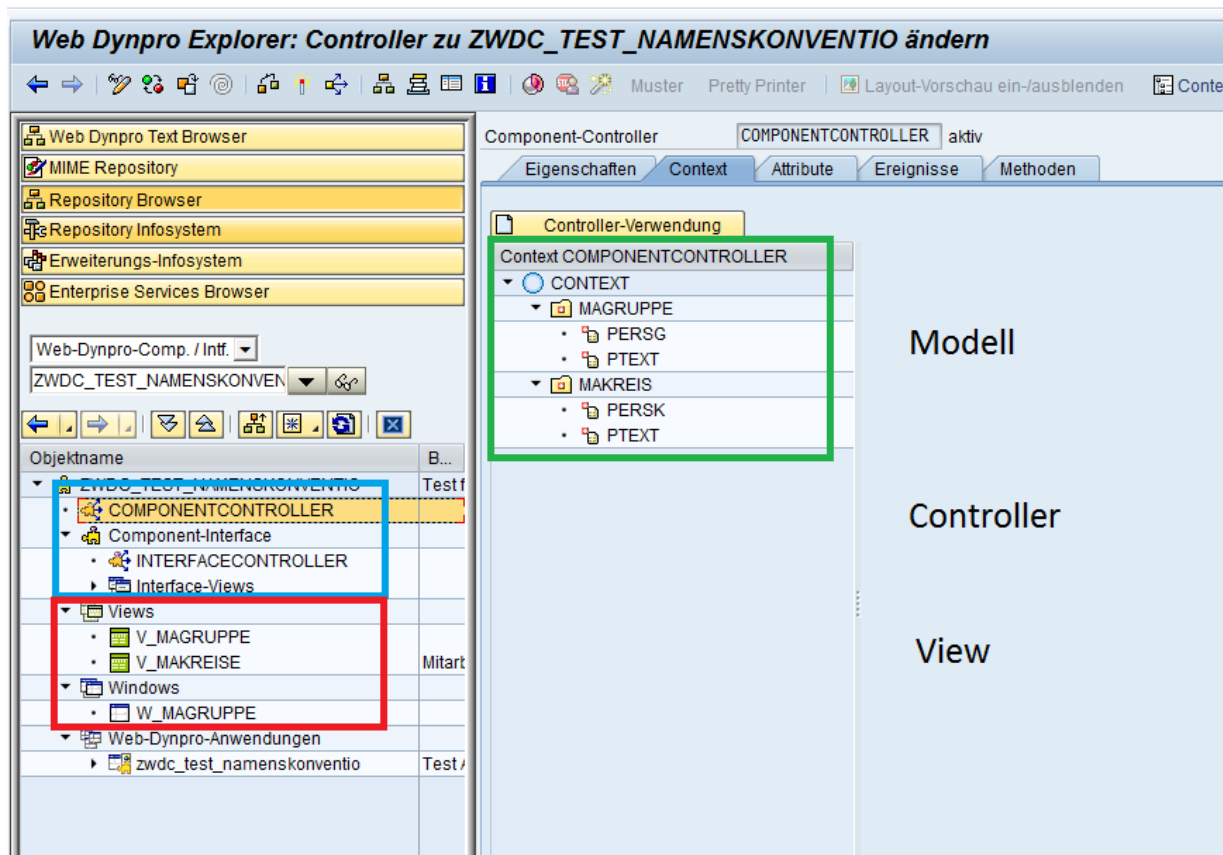
Art	prozessüber- greifend	spezieller Prozess	Beispielhaft für UEB (Überstundenpro- zess)
Präfix	PROC	<Prozess>	UEB
Funktionsbaustein	ZHR_PROC_*	ZHR_<Prozess>_*	ZHR_UEB_*
Klassen	ZCL_HR_PROC_*	ZCL_HR_<Prozess>_*	ZCL_HR_UEB_*
Aktionsprofil		ZHR_<Prozess>	ZHR_UEB
Prozesstyp (CU)	ZHR_PROC_*	ZHR_<Prozess>_*	
Cycles mit Entscheidung	ZHR_PROC_CYC LE_WF*		
Cycles nur mit Informa- tion	INFO*		
Umlaufart		ZHR_<Prozess>_*	ZHR_UEB_*
Entscheidungsprofil	ZHR_PROC_*		
Entscheidungen	ZHR_PROC_*		
Bearbeitergruppen	ZHR_PROC_*		
Service Provider	ZHR_PROC_DIO_*		
Service Provider Variante	ZHR_PROC_DIO_*	ZHR_<Prozess>_DIO_*	ZHR_UEB_DIO_*
Anwendungsbereich	ZHR_PROC_*	ZHR_<Prozess>_*	ZHR_UEB_*
Anwendungsprozess	ZHR_PROC_*		
Entwicklungsbereich	ZHR_PROC*		
Task Manager Elementart		ZHR_<Prozess>_SPS_TASK	ZHR_UEB_SPS_TASK
Template-ID		ZHR_<Prozess>	ZHR_UEB

11.19 Namenskonventionen für Web Dynpro ABAP

11.19.1 Einleitung

Die Vorschläge für die Namenskonventionen werden in Abschnitten dokumentiert, die dem Aufbau der Web Dynpro Architektur entsprechen:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 44 von 56

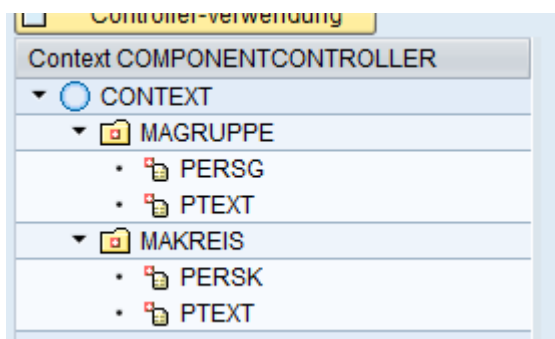


Die Web-Dynpro-Komponente sollte mit ZWDC_ beginnen (WebDynproComponent). Die Web Dynpro-Anwendung sollte denselben Namen tragen wie die Komponente und mit _APP enden (ZWDC_<Bezeichnung>_APP).

11.19.2 Modell

Hierbei wird die Datenhaltung (Context) und die Datenermittlung betrachtet.

11.19.2.1 Context



Die Kontextknoten sind sprechend zu benennen. Wird der Kontextknoten anhand einer Diktionsstruktur abgelegt, werden die Attributnamen übernommen und können nicht verändert werden. Bei manueller Anlage von Attributen werden deswegen keine Prefixe vorgegeben.

Zum Auslesen und Setzen des Kontextes kann der Wizard verwendet werden. Das generierte Coding entspricht im Großen und Ganzen den Programmerrichtlinien der TK.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 45 von 56

Auslesen eines Kontextattributes:

```

2
3     DATA lo_nd_magruppe TYPE REF TO if_wd_context_node.
4
5     DATA lo_el_magruppe TYPE REF TO if_wd_context_element.
6     DATA ls_magruppe TYPE wd_this->element_magruppe.
7     DATA lv_persg TYPE wd_this->element_magruppe-persg.
8
9     * navigate from <CONTEXT> to <MAGRUPPE> via lead selection
10    lo_nd_magruppe = wd_context->get_child_node( name = wd_this->wdctx_magruppe ).
11
12    * @TODO handle non existant child
13    * IF lo_nd_magruppe IS INITIAL.
14    *   ENDIF.
15
16    * get element via lead selection
17    lo_el_magruppe = lo_nd_magruppe->get_element( ).
18    * alternative access via index
19    * lo_el_magruppe = lo_nd_magruppe->get_element( index = 1 ).
20    * @TODO handle not set lead selection
21    IF lo_el_magruppe IS INITIAL.
22    *   ENDIF.
23    * get single attribute
24    lo_el_magruppe->get_attribute(
25      EXPORTING
26        name = `PERSG`
27      IMPORTING
28        value = lv_persg ).
29

```

Setzen eines Kontextknotens:

```

42
43     DATA lo_nd_makreis TYPE REF TO if_wd_context_node.
44
45     DATA lt_makreis TYPE wd_this->elements_makreis.
46
47     * navigate from <CONTEXT> to <MAKREIS> via lead selection
48     lo_nd_makreis = wd_context->get_child_node( name = wd_this->wdctx_makreis ).
49
50    * @TODO handle non existant child
51    * IF lo_nd_makreis IS INITIAL.
52    *   ENDIF.
53
54    * @TODO compute values
55    * e.g. call a model function
56    *
57    lo_nd_makreis->bind_table( new_items = lt_makreis set_initial_elements = abap_true ).
58
59

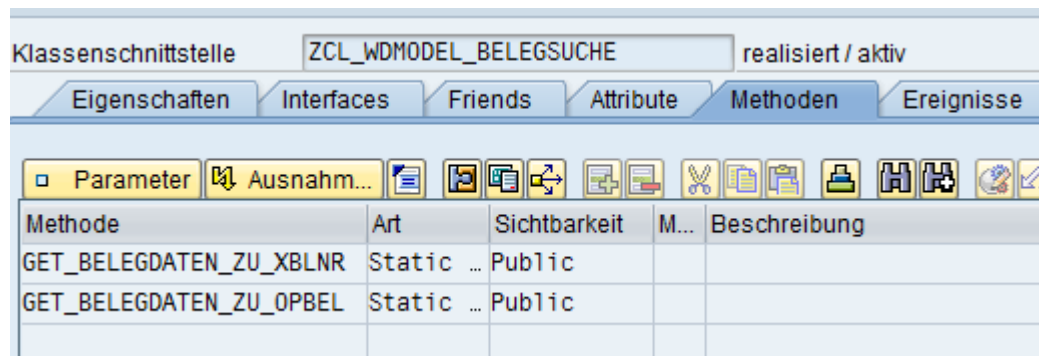
```

11.19.2.2 Datenermittlung

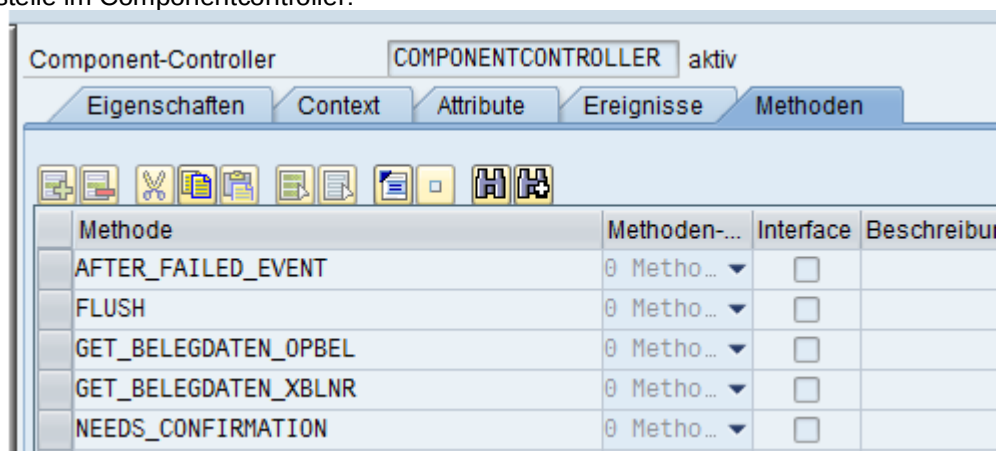
Die Datenermittlung findet im Componentcontroller mit Hilfe der sog. Model-Klassen statt. Für den Anstoß der Datenermittlung werden in Componentcontroller Methoden zur Verfügung gestellt, die von den Views (z.B. einer User-Aktion wie Buttonclick) aufgerufen werden können.

Beispiel für eine Model-Klasse:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 46 von 56

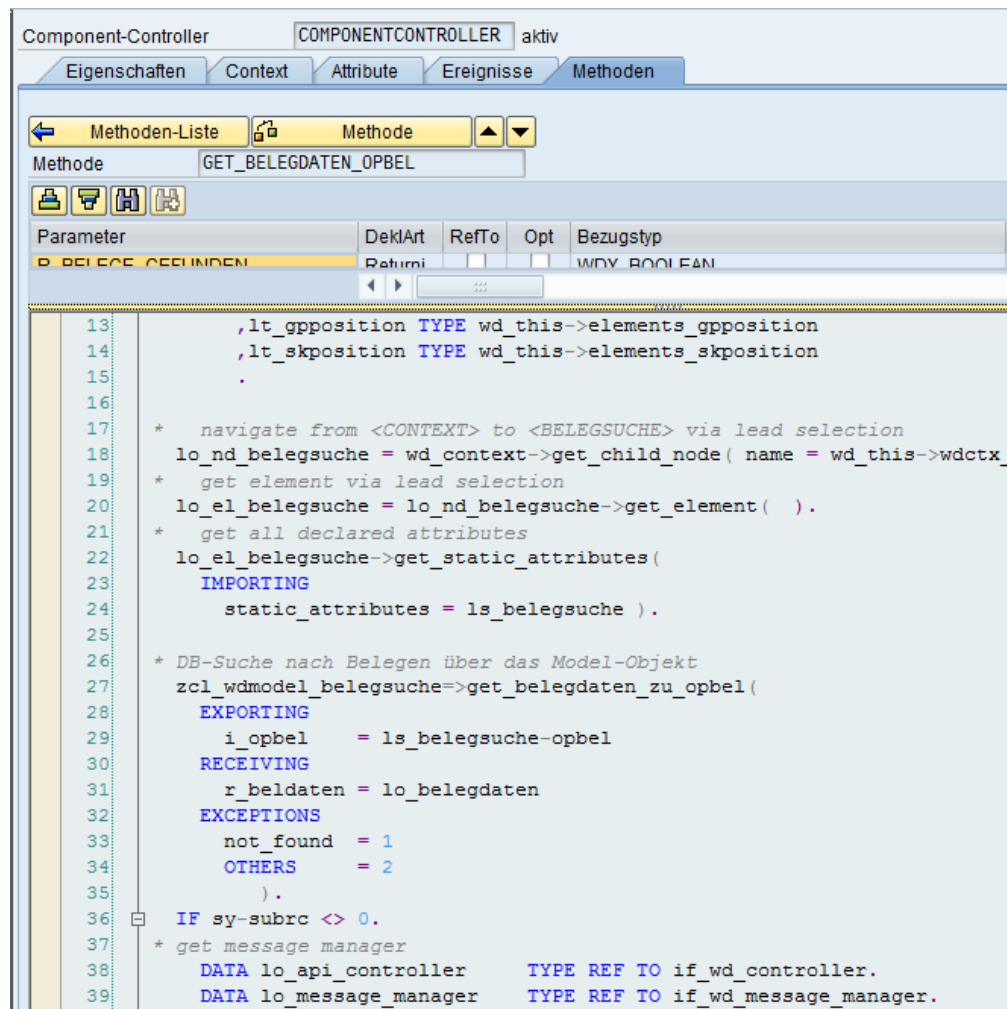


Schnittstelle im Componentcontroller:



Beispiel der Nutzung:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 47 von 56



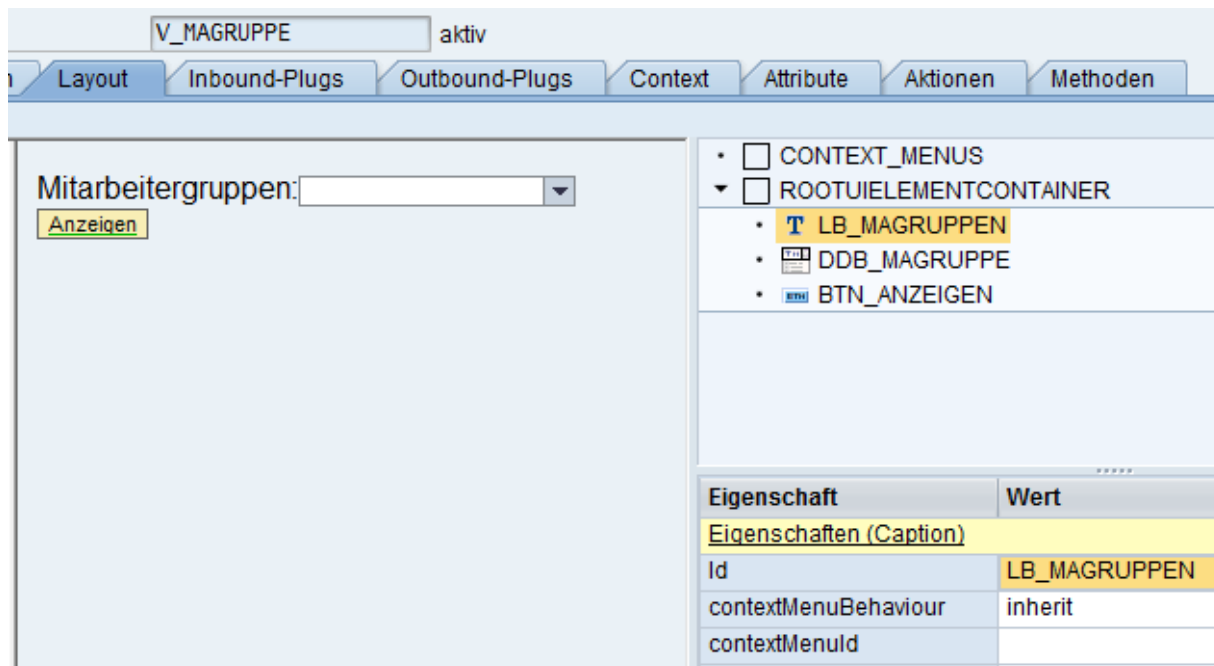
11.19.3 View / Window

Views beginnen mit V_ ,
Windows mit W_.

11.19.4 UI-Elemente

UI-Elemente sollten Präfixe aufweisen, die auf die Typen des Elementes hinweisen:

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 48 von 56



BTN_	Button	CAPTION_	Caption
CB_	CheckBox	CBG_	CheckBoxGroup
DDBI_	DropDownByIndex	DDBK_	DropDownByKey
FD_	FileDownload	FU_	FileUpload
GR_	Gruppe	IF_	InputField
ILB_	ItemListBox	IMG_	Image
LB_	Label	PH_	PageHeader
RB_	Radiobutton	RBGI_	RadioButtonIndex
RBGK_	RadioButtonKey	T_	Tab
TBL_	Table	TBLH_	TableHeader
TBLC_	TableColumn	TBLCE_	TableCellEditor
TIT_	TreeItemType	TNT_	TreeNodeType
TR_	Tree	TS_	TabStrip
TSCB_	TriStateCheckBox	TV_	TextView
VC_	ViewContainerUIElement		

11.19.5 Plugs

11.19.5.1 Outbound-Plugs

TO_ <sprechender Name Aktion>

z.B. TO_SHOW_ITEMS

11.19.5.2 Inbound-Plugs

ON_ <sprechender Name Auslöser>

z.B. ON_CLOSE, ON_SELECT_LINE

FROM_ <sprechender Name Auslöser>

z.B. FROM_DETAIL_LISTE

11.19.6 Controller

CC_ <Bezeichnung>

Custom-Controller

CONF_ <Bezeichnung>

Configuration-Controller

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 49 von 56

11.19.7 Controllermethoden

IS_<Bedingung> ist Bedingung erfüllt?
 GETCTX_<Bezeichnung> Daten aus Context lesen
 SETCTX_<Bezeichnung> Daten im Context setzen
 GETMODEL_<Bezeichnung> Daten aus Modell lesen
 SUPPLY_<Bezeichnung> Supply-Funktion
 ON_<Ereignis> Ereignisbehandler

11.19.8 Assistance-Klassen

ZCL_WDA_<Bezeichnung> Assistance-Klassen

11.19.9 Konfigurationen

11.19.9.1 Komponenten-Konfiguration

<Komponentenname>_CC Komponenten-Konfiguration

11.19.9.2 Anwendung-Konfiguration

<Anwendungsname>_AC Anwendungs-Konfiguration

11.19.10 Floorplan-Manager

Für Floorpläne
 bspw.:
 ZWD_FPM_...
 Feeder Klassen
 GO_ bei Definition im ComponentController analog NET311

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 50 von 56

12 Einheitliches Layout

12.1 Programmaufbau

Kurzbeschreibung

Im Kopf steht eine Kurzbeschreibung (mind. 5 Zeilen)

Änderungshistorie

Änderungshistorie (Datum, Änderer, kurze Änderungsbeschreibung. Die aktuellste Änderungszeile steht oben)

```
*-----
* 20.02.2002 Änderer   Kurztext
*      ...
* 01.02.2002 Änderer   Kurztext
*      ...
*-----
```

globale Deklarationen

Ereignisse (soweit zutreffend)

Definition der Form-Routinen (soweit zutreffend)

12.2 maximale Zeilenlänge

Die maximale Zeilenlänge darf 132 Zeichen nicht überschreiten.

12.3 Kommentierung

Programmkomentierungen erfolgen:

am **Anfang eines Programms** (eigener Kommentarblock)

Kurze Beschreibung, welchen Zweck das Programm erfüllt, und welche Voraussetzungen für das Programm gegeben sein müssen.

```
*-----
* Kurzbeschreibung Programm
* ...
* ...
* ...
* ...
*-----
```

am **Anfang einer Codeeinheit** (eigener Kommentarblock)

Kurze Beschreibung der Funktion der Codeeinheit.

```
*-----
* Kurzbeschreibung Codeeinheit
* ...
*-----
FORM.
```

innerhalb des Programmcodes (jeweils in einer eigenen Zeile)

Einzelnen Kommentare, die die Lesbarkeit des Programmcodes erhöhen. Der Kommentar steht oberhalb der Codezeile.

```
IF ZPERNR-BEGDA = LV_STICHTAG_ALT.
...
* neuer Stichtag
ELSE.
* Wohnort liegt in Westdeutschland
IF ZPERNR-WOKNZ = K_WOKNZ_WEST.
```

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 51 von 56

```

...
ENDIF.
...
ENDIF.

```

Diese dürfen den Programmfluss nicht unterbrechen d.h. sie müssen **mindestens** so weit eingerückt sein wie das jeweilige Coding.

ggf. bei Datendeklarationen und Übergabeparametern (möglichst in der gleichen Zeile)
diese Kommentare sollten nur verwendet werden, wenn der Sinn nicht aus dem
Namen der Variablen ersichtlich ist

```

INFOTYPES: 0008,      "Infotyp - Basisbezüge
            0053.      "Infotyp - Betriebsrenten

```

In jedem Fall gilt, ein überschaubares Coding mit sprechendem Namen ist besser als lange ausführliche Kommentierung.

12.4 Maximale Befehlsanzahl pro Zeile

Aus Gründen der Lesbarkeit darf pro Zeile jeweils nur ein Befehl stehen.

Beispiel.:
statt

```
bkpf-bukrs = k_buk1. bkpf-gjahr = k_gj01. bkpf-belnr = k_numk1.
```

besser

```

bkpf-bukrs      = k_buk1.
bkpf-gjahr      = k_gj01.
bkpf-belnr      = k_numk1.

```

12.5 Hervorhebung von Verarbeitungsblöcken durch Einrückungen

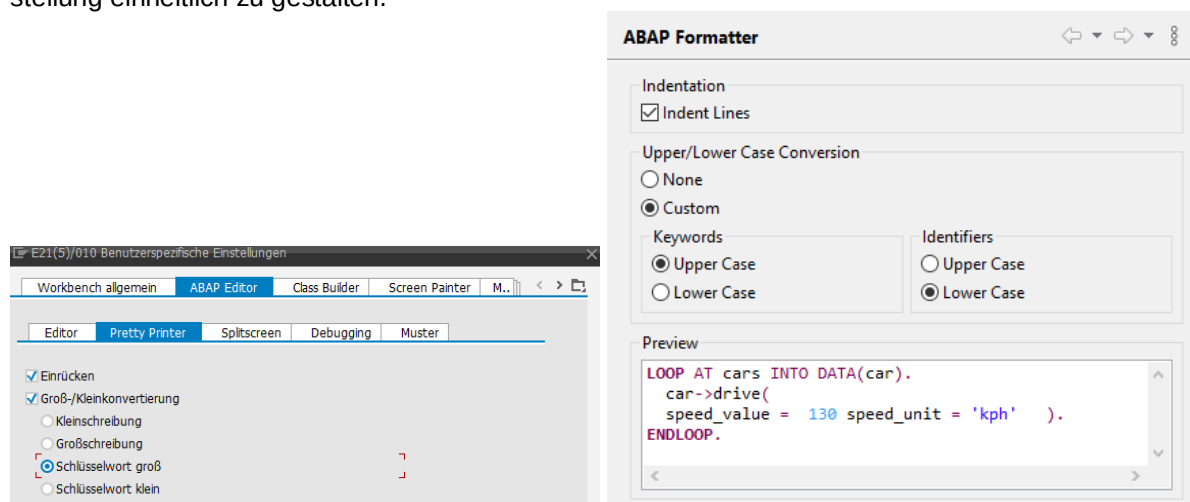
Wird durch Einsatz des Pretty Printers unterstützt. Gleiches gilt für den ABAP Formatter im Eclipse.

12.6 Verständlichkeit von Abfragen

Abfragen sind einfach zu halten und zu strukturieren. Sie müssen für einen Dritten verständlich sein

12.7 Einsatz maschineller Unterstützung

Die Programme sind durch Anwendung von "Pretty Printer" bzw. „ABAP Formatter“ mit folgender Einstellung einheitlich zu gestalten.



SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 52 von 56

12.8 Verwendung von Templates

Unter den Namen ZVORL_xx werden Templates bereitgestellt für Reports und Funktionsbausteine (FuGruppe ZVORL) (s. Beispiel am Ende). Gültige Templates s. 16.4 Anhang.

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 53 von 56

13 Kennzeichnung der Sourcen

Jede Source ist mindestens mit den Attributen "Anwendung" "Status" und "Entwickungsklasse" zu versehen.

14 Dokumentationsdokumente "technische Entwürfe"

Entwicklungsvorhaben werden in technischen Entwürfen dokumentiert. Diese beschreiben auf Basis der fachlichen Vorgaben die Art der technischen Implementierung. In Ihnen sind mindestens zu beschreiben:

- Modulübergreifende Ablauflogik
- Entwürfe von Masken
- Auflistung der Namen von IMG-relevanten Einträge
- Auflistung der Namen kundeneigener Tabellen
- Auflistung der Namen erstellter ABAP-Programme
- Auflistung der Namen von Funktionsbausteinen / Funktionsgruppen
- Auflistung der Namen von Klassen
- Auflistung der Namen von WebDynpro

15 Maschinelle Prüfung

Zur maschinellen Prüfung wird die Funktionalität SCI eingesetzt. Bei der Prüfung dürfen keine Fehler entstehen. Sofern erforderlich und begründbar, sind entsprechende Pragmas zu verwenden. Sofern für einen Sachverhalt keine Pragmas vorhanden sind können #EC- Kommentierungen verwendet werden. Die bei TK vorgeschriebene SCI-Prüfungsvariante wird ausgeführt bei der Funktion "Auftrag - Gesamtprüfung - Objekte"

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmierrichtlinie.docx Nur für den Dienstgebrauch		Seite 54 von 56

16 Anhang

16.1 Checklisten für die Programmprüfung

Der im P42(001) hinterlegte Stand bildet die jeweils gültigen Checklisten ab.

Stand 30.01.2024:

Checklisten-Typ: Workbench

- 1 Entspricht der Programmcode den Programmierrichtlinien?
- 2 Ist der Code selbsterklärend, d.h., kann er durch Lesen verstanden werden (nicht durch Debugging)?
- 3 Ist der Code ohne Fehler generierbar?
- 4 Erläutern die Kommentare den Zweck und die Verwendung?
- 5 Sind Workarounds und Ausnahmefälle gesondert mit Grund und Datum kommentiert?
- Sind alle nicht mehr benötigten Workarounds entfernt?
- 6 Werden beschreibende Namen für Variablen verwendet?
- 7 Sind alle Variablen so lokal wie möglich und so global wie nötig definiert?
- 8 Ist die Länge der Implementierung einer Form, Funktion, Methode, etc. angemessen?
- 9 Sind die Möglichkeiten, Funktionalität auszulagern, hinreichend genutzt?
- 10 Ist die Verzweigungsstruktur einfach genug? Sind komplizierte Fälle aufgeteilt über Forms / Funktionen / Klassen?
- 11 Werden übermäßig komplexe Lösungen vermieden, z.B. bei IF-Abfragen?
- 12 Ist der Code Unicode fähig?
- 13 Werden keine obsoleten Sprachkonstrukte verwendet?
- 14 Ist die erweiterte Programmprüfung via Transaktion SCI fehlerfrei?
- 15 Ist die Anzahl der Warnungen der erweiterten Programmprüfung via Transaktion SCI verringert worden oder es existieren keine Warnungen?
- 16 Sind für alle neuen Tabellen Berechtigungsgruppen vergeben und Vorgaben zur Protokollierung umgesetzt?
- 17 Wurde die Job-Dokumentation angepasst, sofern notwendig?
- 18 Wurde die technische Dokumentation angepasst, sofern notwendig?
- 19 Wurde eine ICF-Service-Aktivierung beauftragt, sofern notwendig?
- 20 Es gibt keine offenen relevanten Findings

Alle Punkte der Checkliste sollten mit 'Ja' beantwortet sein, wenn der Code zur Freigabe ansteht!

Checklisten-Typ: SAPUI5-App

- 1 Programmierrichtlinien eingehalten
- 2 Chrome Entwicklertools zur App-Optimierung eingesetzt
- 3 SAPUI5 Support Assistant zur App-Optimierung eingesetzt
- 4 App-Verhalten auf SAP Fiori Frontend-Server überprüft
- 5 App auf relevanten Geräten getestet
- 6 GitHub-Repository und -Branches korrekt gepflegt

Alle Punkte der Checkliste sollten mit 'Ja' beantwortet sein, wenn der Code zur Freigabe ansteht!

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 55 von 56

16.2 Beispielprogramm

```

REPORT zvorl_hr.
*-----
*   Programmkurzbeschreibung
*   Rahmenprogramm für HR-Programme
*   ...
*-----
*   01.01.2002 Wegener   ...
*   01.01.2001 Wegener   Ersterstellung
*
*---+---1---+---2---+---3---+---4---+---5---+---6---+---7--
*-----
*   globale Deklaration
*-----
*
*       +-----
*       ! Infotypes
*       +-----
*
*       +-----
*       ! Classes
*       +-----
*
*       +-----
*       ! Tables
*       +-----
*
*       +-----
*       ! Include
*       +-----
*
*       +-----
*       ! Typen
*       +-----
*
*       +-----
*       ! Konstanten
*       +-----
*
*       +-----
*       ! Variablen
*       +-----
*
*       +-----
*       ! Tabellen (intern)
*       +-----
*
*       +-----
*       ! Parameter / Selektionsvariablen
*       +-----
*-----
*   Ereignisse
*-----
INITIALIZATION.
  PERFORM          initialisierung.

AT SELECTION-SCREEN.
  PERFORM          eingabe-pruefen.

START-OF-SELECTION.
  PERFORM          verarbeitung.

*-----
*   FORMS

```

SAP R/3		SAP R/3
Dokumentart: Richtlinie	Version: 1.8	Erstelldatum: 1.1.2003
Autor: IT.BA.SAP	Durchwahl: - 2035	Aktualdatum: 01.02.2025
Ablage: N:\IB.1\SAP\Richtlinien\ENTWURF tkone CA-KO- Programmerrichtlinie.docx Nur für den Dienstgebrauch		Seite 56 von 56

```

*-----
FORM initialisierung.
*           lokale Definitionen
DATA: lv_variable      TYPE c.
....

*           Implementierung der Form
....

ENDFORM.

```

16.3 Gültige Nachrichtenklassen / Syslog-Meldungsklassen

Nachrichtenklasse	Kurzbeschreibung
Z0	TK-Nachrichtenklasse (Basis)
Z1	TK-Nachrichtenklasse (Finanzbuchhaltung)
Z2	TK - Nachrichten Anlagenbuchhaltung
ZCD	TK-Nachrichtenklasse FS-CD
ZH	TK-Nachrichtenklasse (Personalwesen)
ZX	TK-Nachrichtenklasse cross-system

Syslog-Meldungsklasse	Kurzbeschreibung
Z0	TK-Klasse für Meldungen aus dem Auftragsmonitor
Z1	TK-Klasse für Meldungen bei Fehlerfällen, die zum programmgesteuerten Rücksetzen führen (Rollback-Auslösung)

16.4 Templates

Name	Kurzbeschreibung
zvorl_hr	Vorlage HR-Programme
zvorl_fico	Vorlage FI/CO-Programme
zvorl_bc	Vorlage Basis-Programme
zvorl_userexit	Vorlage User-Exit mit Firmentrennung
zvorl	Funktionsgruppe mit Mustern für FuBa

16.5 Hauptgebiete / Untergebiete

Hauptgebiet (Liste: FI, CO, HR, BC, MQ, CD, TR, MM, RE, PS, SD)

Untergebiet nach Teilmodul im Hauptgebiet

16.6 Liste der Programmberechtigungsgruppen

Obere Ebene:

ZF Schutzgruppe „Finanz“

ZP Schutzgruppe „Personal“

ZS Schutzgruppe „Basis“

Jeweils ggf. mit Untergliederungen (Abstimmung ausführen)