

# Installationsleitfaden



AppKnight

## BTRK Bugtracker



Bugtracker

<i>Version</i>	<i>2401</i>
<i>Status</i>	<i>Freigegeben</i>
<i>Datum</i>	<i>29.01.2024</i>

# Inhaltsverzeichnis

---

## *Inhalt*

1	Vorwort .....	2
2	AppKnight-Addons .....	2
2.1	Bugtracker .....	2
2.2	Transaktionsaufruf .....	2
3	Installation .....	3
4	Einrichtung .....	3
4.1	Berechtigungen .....	3
4.1.1	Spezielle Berechtigungen .....	3
4.1.2	Customizingberechtigung .....	3
4.2	Einrichtung über Customizingmenü .....	4
4.3	Einrichtung über Bereichsmenü .....	4
4.4	Nummernkreisintervall pflegen .....	5
4.5	Benutzerabhängige Konfiguration .....	6
4.5.1	Definition Hauptmandant .....	6
4.5.2	Ticketzuordnung bei Transport .....	7
4.6	Fehlerkategorien & Prioritäten .....	9
4.7	Parameter .....	9
4.8	Mailtext .....	10
4.9	Produkte, Komponenten und Schlagworte (Tags) .....	10
4.10	Statuspflege .....	11
4.10.1	Folgestatus definieren .....	11
4.11	Standard Mailtext .....	11
4.11.1	Beispieltext .....	11
4.11.2	Felddefinitionen verwenden .....	12
4.12	Benutzerrollen .....	12
5	Individuelle Anpassungen .....	12
5.1	Kundeneigene Felder .....	12
5.1.1	Darstellung im Ticket .....	14
5.1.2	Definition .....	15
5.2	Kundeneigenes Dynpro .....	15
5.2.1	Felddefinitionen .....	16
5.2.2	Realisierungsideen .....	18

## 1 Vorwort

Herzlichen Dank, dass Sie sich für dieses Produkt entschieden haben! Unser Ziel ist es, gute und zuverlässige Software für SAP-Systeme zu liefern.

Ein weiteres Ziel ist es, Grundfunktionalitäten möglichst so zu kapseln, dass diese in verschiedenen anderen Produkten verwendet werden können. Dadurch profitieren Sie von der stetigen Weiterentwicklung selbst der kleinsten Bestandteile.

Bitte beachten Sie den [generellen Leitfaden](#), in dem wir Informationen zur Verfügung stellen, die für alle AppKnight-Produkte gelten.

## 2 AppKnight-Addons

Die AppKnight-Produkte werden in einem separaten Namensraum erstellt: /AKN/. Die einzelnen Repositoryobjekte der Produkte erhalten jeweils ein vierstelliges Produktkürzel. So ist schnell ersichtlich, zu welchem Produkt ein Objekt gehört.

### 2.1 Bugtracker

Der Bugtracker ist ein Addon-Produkt für Ihr SAP R/3-System. Das Kürzel für den Bugtracker ist BTRK. Das zugeordnete Paket lautet daher: /AKN/BTRK.

### 2.2 Transaktionsaufruf

Der Namensraum für kundeneigene Objekte und Partnerobjekte beginnt mit einem Schrägstrich „/“ gefolgt von einem bis zu sechs-stelligen alphanumerischen Code und wiederum gefolgt von einem „/“.

Transaktionen unterliegen ebenfalls dieser Namenskonvention. Leider können diese Transaktionen nicht direkt in die OK-Codezeile eingegeben werden. Sie müssen das Kommando „/n“ oder „/o“ voranstellen.

- „/n“ bedeutet: Transaktion in aktuellem Modus starten
- „/o“ bedeutet: Transaktion in neuem Modus starten

Beispiel: /n/AKN/BTRK01

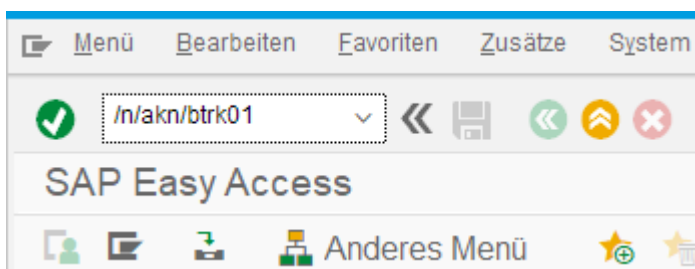


Abbildung 1 Eingabe Transaktion in OK-Codezeile

Groß-/ Kleinschreibung spielt keine Rolle!

### 3 Installation

Für die Installation der zur Verfügung gestellten Transportdateien beachten Sie bitte den [Generellen Leitfaden](#) für AppKnight-Produkte.

### 4 Einrichtung

Der AppKnight Bugtracker ist ein fertiges Produkt. Bevor Sie es jedoch nutzen können, müssen einige Grundeinstellungen (Customizing) vorgenommen werden. Die hier vorgeschlagenen Werte sind Richtwerte. Sie können sich bei der Einrichtung Ihre eigenen Einstellungen erzeugen.

Sofern es Einstellungen sind, die Sie exakt so übernehmen müssen (Nummernkreise, Attribute, etc.), weisen wir Sie explizit darauf hin.

#### 4.1 Berechtigungen

Für den Bugtracker gibt es zwei Berechtigungsobjekte:

- /AKN/BTRK1 [AppKnight] Bugtracker Activity
- /AKN/BTRK2 [AppKnight] Bugtracker main authorizations

/AKN/BTRK1 ist die allgemeine Berechtigung

/AKN/BTRK2 definiert zusätzlich die Berechtigungen über Produkt-ID's und Komponenten-ID's.

Zum Anlegen eines Tickets benötigt der Anwender die Berechtigung mit ACTVT 01 in beiden Berechtigungsobjekten.

Notwendige Aktivitäten:

- Zum Anlegen: ACTVT 01
- zum Ändern: ACTVT 02
- zum Anzeigen ACTVT 03

##### 4.1.1 Spezielle Berechtigungen

Normalerweise soll ein Anwender ein Ticket in einem finalen Status nicht mehr ändern dürfen. Ein Key-User oder Administrator sollte diese Möglichkeit jedoch haben: Aktivität 36 erlaubt die Änderung eines finalen Status.

##### 4.1.2 Customizingberechtigung

Für die Pflege der Einstellungen im Bugtracker-Customizing benötigen Sie die Berechtigung /AKN/BTRK1 mit der Aktivität „70 - Administer“

## 4.2 Einrichtung über Customizingmenü

Starten Sie die Transaktion /n/AKN/BTRK\_C um die Einstellungen vorzunehmen. Alle notwendigen Einstellungen sind aus dem hier aufgeführten Menü zu erreichen:

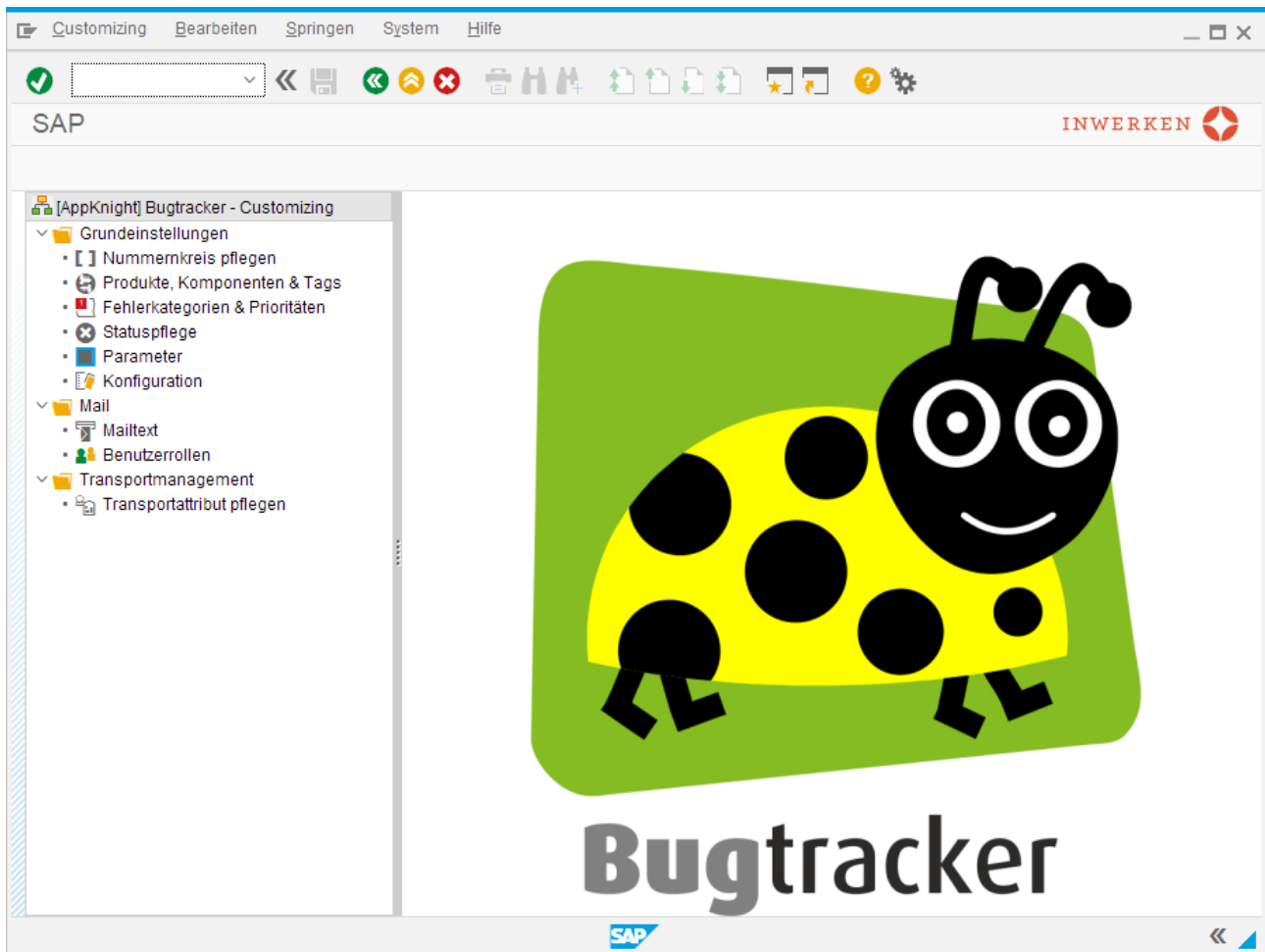


Abbildung 2 Transaktion /akn/btrk\_c – Einstellungen

## 4.3 Einrichtung über Bereichsmenü

Sie können ein Großteil der Einstellungen auch über das Bereichsmenü /n/AKN/BTRK erreichen. Geben Sie den Namen des Bereichsmenüs in die OK-Codezeile des SAPGUI ein.

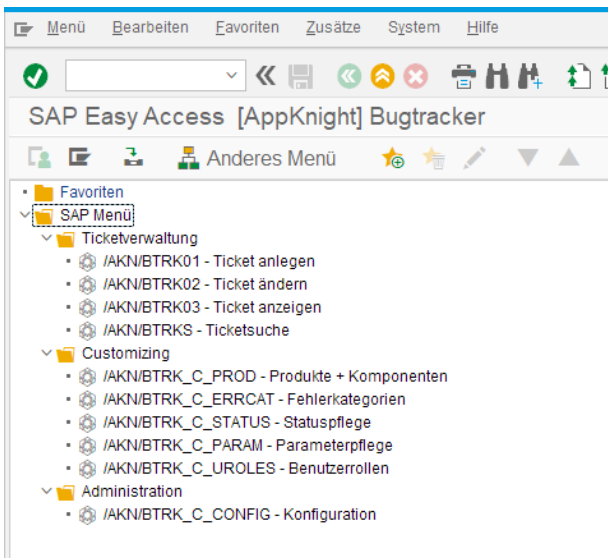


Abbildung 3 Bereichsmenü /akn/btrk

Eventuell sehen Sie den Transaktionscode nicht. Sie können dieses Verhalten einstellen über das Menü **Zusätze • Einstellungen** und dann **Technische Namen anzeigen**.

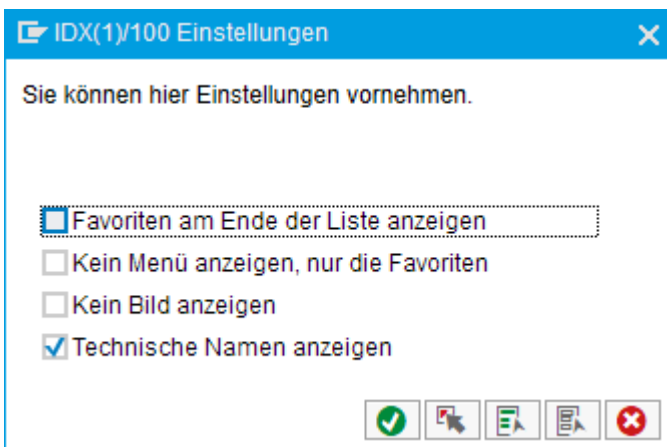


Abbildung 4 Favoriten - Technische Namen anzeigen

#### 4.4 Nummernkreisintervall pflegen

Der AppKnight Bugtracker verwendet den Nummernkreis /AKN/BTRK, um die Ticketnummern zu erzeugen. Sie müssen das Nummernkreisintervall „01“ anlegen. Sie können einen Nummernbereich wählen, beispielsweise 1 – 1999999. Sie können je Produkt-ID einen separaten Nummernkreis definieren.

Um das Intervall anzulegen, klicken Sie auf die mittlere Schaltfläche „Intervalle Ändern“.

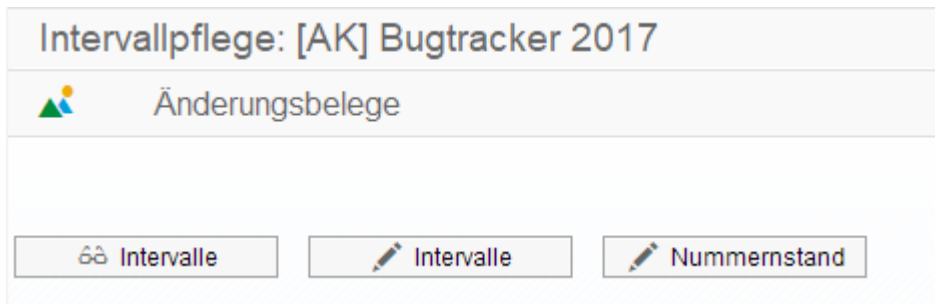


Abbildung 5 Transaktion SNRO - Intervalle pflegen

Tragen Sie das Nummernintervall „01“ ein und geben Sie die Start- und Endnummer ein.



Abbildung 6 Transaktion SNRO - Intervall einfügen

## 4.5 Benutzerabhängige Konfiguration

Sie können einige Funktionen benutzerabhängig einstellen. Bisher werden die folgenden Einstellungen unterstützt:

- Definition Hauptmandant
- Abfrage einer Ticketzuordnung bei
  - Anlage eines Transportauftrags
  - Freigabe eines Transportauftrags

### 4.5.1 Definition Hauptmandant

Der Bugtracker ist Mehr-Mandantenfähig. Das bedeutet, Sie können die Tickets in einem Mandanten verwalten (der Hauptmandant) und zusätzlich in beliebigen anderen Mandanten, zum Beispiel in einem Mandanten für Customizing, auf die Tickets des Hauptmandanten zugreifen.

In dem Hauptmandanten werden alle Einstellungen für die Transportanlage und die Transportfreigabe definiert. Auch die Benutzerabhängige Einstellung wird hier vorgenommen. In allen anderen Mandanten definieren Sie lediglich den Hauptmandanten.

Tragen Sie die Nummer des Mandanten in die Spalte „Benutzername“ ein.

### 4.5.2 Ticketzuordnung bei Transport

Verwenden Sie den Parameter „TRSC“ um je Benutzer einzustellen, ob bei Transportauftrags-**anlage** ein Popup erscheint, in dem der Anwender eine Ticketnummer zuordnen muss. Verwenden Sie den Parameter „TRSP“ um je Benutzer einzustellen, ob bei Transportauftrags**freigabe** ein Popup erscheint.

Sicht "[AppKnight] Bugtracker - Konfiguration" anzeigen: Übersicht

🔍 📄 📄 📄

[AppKnight] Bugtracker - Konfiguration

Konf. Typ	Benutzername	Freigabe	Status
CLNT Konfiguration Multi-Clients (Hauptmandant) ▾	100	0	▾
TRSC Konfiguration Transport Popup "Anlage" ▾	EWULFF	2 Optional	▾ KWT
TRSP Konfiguration Transport Popup "Freigabe" ▾		3 Deaktiviert	▾
TRSP Konfiguration Transport Popup "Freigabe" ▾	EWULFF	2 Optional	▾ KWT

Abbildung 7 Beispielkonfiguration

Folgende Ausprägungen für die Spalte „Freigabe“ sind wählbar:

- Ja: Der Anwender muss eine Ticketnummer eintragen
- Optional: Das Popup erscheint, aber der Anwender muss keine Ticketnummer eintragen
- Deaktiviert: Das Popup erscheint nicht

Tragen Sie in der Spalte „Status“ den Typ der Transportaufträge ein, zu denen eine Abfrage erfolgen soll. Die gängigsten Transportauftragstypen sind:

- K: Workbench-Auftrag
- T: Transport von Kopien
- W: Customizing-Auftrag

Nutzen Sie die F4-Hilfe, um die Transportauftragstypen auszuwählen.



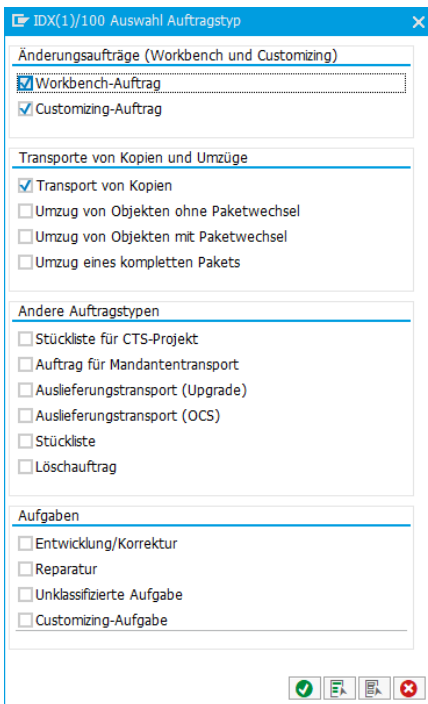


Abbildung 8 Auswahl Transportauftragstypen

### **Hinweis:**

Die Abfrage erfolgt mit Hilfe des Business-AddIns CTS\_REQUEST\_CHECK in der Business-AddIn-Implementierung /AKN/BTRK\_REQ\_CHK (Klasse /AKN/CL\_BTRK\_IM\_TRANSPORT, Methode IF\_EX\_CTS\_REQUEST\_CHECK~CHECK\_BEFORE\_RELEASE sowie IF\_EX\_CTS\_REQUEST\_CHECK~CHECK\_BEFORE\_CREATION).

CHECK\_BEFORE\_RELEASE Methode ist für Prüfung und Zuweisung des Transporte vor Transportfreigabe und die CHECK\_BEFORE\_CREATION Methode ist für Prüfung und Zuweisung der Transporte bei Anlage eines Transportes zuständig.

Wenn Sie eine eigene Abfrage programmieren möchten, zum Beispiel um die Ticketnummer aus dem Auftragstext zu extrahieren oder um weitere Informationen aus anderen Tabellen oder Systemen anzuzeigen, dann deaktivieren Sie einfach die Bugtracker-Implementierung, kopieren diese in eine eigene Implementierung und aktivieren diese dann.

Starten Sie die Transaktion SE19, tragen im Bereich *Implementierung bearbeiten* im Feld *Klassisches BAdI – Implementierung* den Namen /AKN/BTRK\_REQ\_CHK ein. Klicken Sie auf die Schaltfläche *Ändern* und deaktivieren Sie den BAdI. Legen Sie dann einen eigenen BAdI in der Transaktion SE19 an.

Möchten Sie bei Anlage eines Transportes einen Vorschlag für das zuzuweisende Ticket ermöglichen, können Sie diese Funktion mit Hilfe des Erweiterungsspot /AKN/BTRK\_BUG\_PROPOSAL der Badi-Definition /AKN/BTRK\_BUG\_PROPOSAL zur Verfügung stellen. Eine Beispielimplementierung /AKN/CL\_BTRK\_BUG\_PROPOSAL\_EXAM wie eine Prüfung des zuzuweisenden Tickets aussehen kann, ist an der Badi-Definition als Beispiel-Klasse hinterlegt.

Um die Funktion zu verwenden, legen Sie eine Implementierung in der Transaktion SE19 im Bereich *Implementierung bearbeiten* im Feld *Neues BAdI* dafür an und verwenden Sie die den Badi-Definitionsnamen /AKN/BTRK\_BUG\_PROPOSAL.

## 4.6 Fehlerkategorien & Prioritäten

Mit Fehlerkategorien können Sie Kategorien und Unterkategorien bzw. Fehlerprioritäten bestimmen.

## 4.7 Parameter

Folgende Parameter werden unterstützt:

Parametername	Parameterwert	Beschreibung
<b>MAILTEXT_SO10</b>	/AKN/BTRK_MAIL_TEXT_01	Name des Textbausteins, der für das Senden der Email verwendet wird.
<b>SO1_NAME</b> <b>SO2_NAME</b> <b>SO3_NAME</b>	Name einer SAP-System-ID	Tragen Sie die System-ID eines ihrer SAP-Systeme ein, um Informationen über in der Auftragsübersicht den Transportstatus zu den Transportaufträgen zu erhalten
<b>SO1_NAME</b>		Kurzbezeichnung eines ersten angebotenen SAP Systems
<b>SO2_NAME</b>		Kurzbezeichnung eines zweiten angebotenen SAP Systems
<b>SO3_NAME</b>		Kurzbezeichnung eines dritten angebotenen SAP Systems
<b>COMMENT_-DIALOG_TYPE</b>		Steuerung, welches Popup für das Erfassen von Kommentaren verwendet wird. Durch Parameterwert „DYNPROPOPUP“ wird das neue Popup verwendet, in dem die Funktionstasten genutzt werden können (F5 um den Kommentar zu übernehmen, ESC zum Abbrechen).

*Tabelle 1 Parameter*

Die SO1\_NAME, SO2\_NAME und SO3\_NAME Parameter sind optional. Diese werden sofern verwendet im Bereich „Transporte“ angezeigt und zeigen den letztbekanntesten Return Code des Transportes für das jeweilige System an.

#### 4.8 Mailtext

Sie können einen eigenen Mailtext definieren. Den zu verwendenden Text weisen Sie dem Parameter MAILTEXT\_SO01 zu. Den Mailtext erstellen und bearbeiten Sie mit der Transaktion SO10.

#### 4.9 Produkte, Komponenten und Schlagworte (Tags)

Mit *Produkten* sind im AppKnight Bugtracker die Bereiche gemeint, für die Sie Tickets anlegen können. Sie können je Produkt-ID einen eigenen Nummernkreis definieren. Typische Produkte sind zum Beispiel:

- SAP
- Hardware
- Software

Zu jedem Produkt können Sie Unterprodukte, sogenannte *Komponenten* anlegen. Typische Komponenten sind zum Beispiel:

- SAP
  - Zusatzprogrammierung
  - Modul SD
  - Modul MM
  - Übersetzung
  - Transporte
- Hardware
  - PC/ Notebooks
  - Netzwerk
  - Zubehör

Zu jedem Produkt können Sie *Schlagworte*, sogenannte *Tags*, anlegen. Die Schlagworte können Ihnen im Ticket helfen, das Problem besser zu beschreiben. Wenn im SAP-Umfeld zum Beispiel ein Anwender ein Problem mit einem Programm oder einer Transaktion hat, dann können Sie die Schlagworte „REPORT“ und „TRANSACTION“ definieren. In der Erfassungsmaske können Sie diese Tags auswählen und das entsprechende Objekte direkt benennen. Dies ist besonders hilfreich, wenn Sie zum Beispiel auswerten möchten, wie oft es Probleme mit bestimmten Programmen gab.

Typische Schlagworte sind zum Beispiel:

- SAP
  - Programm
  - Transaktion
  - Berechtigung

- System-ID
- Hardware
  - Seriennummer
  - Identnummer
  - Inventarnummer

## 4.10 Statuspflege

Der AppKnight Bugtracker arbeitet, wie jedes andere Ticketsystem auch, mit verschiedenen Status, um den Bearbeitungszustand des Tickets zu dokumentieren. Sie können die Status frei wählen und an Ihre Bedürfnisse anpassen.

Achten Sie jedoch darauf, dass es einen Status vom Typ „Anfangsstatus“ gibt. Es können mehrere Status vom Typ „Endestatus“ definiert werden. Beim Setzen eines Status vom Typ „Endestatus“ wird das Feld „Endedatum“ in den Hauptdaten des Tickets gesetzt.

Typische Statusdefinitionen sind:

- Neu (Anfangsstatus)
- In Arbeit
- Im Test
- Erledigt (Endestatus)
- In Prüfung
- Abgelehnt (Endestatus)

### 4.10.1 Folgestatus definieren

Sie können für jeden Status einen *Folgestatus* definieren. Wenn der Status geändert wird, kann nur ein Status ausgewählt werden, der als Folgestatus definiert wurde.

Dadurch vermeiden Sie zum Beispiel, dass ein Ticket vom Status „Neu“ ohne weitere Prüfung oder Tests auf den Status „Erledigt“ gesetzt wird.

## 4.11 Standard Mailtext

Der Mailversand spielt eine wichtige Rolle im Ticketsystem. Es ist ein Standardtext im System hinterlegt. Sie können diesen Text ändern oder einen anderen Text in den Parametern des Bugtrackers einstellen.

### 4.11.1 Beispieltext

```
* Hello!  
*  
* The ticket &/AKN/BTRK_DYNPRO-BUG_ID& has been changed.  
* topic: &/AKN/BTRK_DYNPRO-SUMMARY&  
* Please check what to do next.  
*  
* Cheers  
* Your Bugtracker.  
*
```

```
U2 Further information:
B1 Deadline: &/AKN/BTRK_DYNPRO-DEADLINE&
U2 Assigned persons
B1 &USER_ROLES&
*   ~~~
*   This mail was sent by Bugtracker
*   https://AppKnight.de
```

#### 4.11.2 Felddefinitionen verwenden

Im Mailtext können Sie Felder des Bugtrackers einbinden. Sie können alle Felder benutzen, die in der Struktur /AKN/BTRK\_DYNPRO vorhanden sind.

Eine Sonderrolle nehmen die folgenden Felddefinitionen ein:

##### 4.11.2.1 USER\_ROLES

Die Felddefinition USER\_ROLES gibt alle im Ticket definierten Benutzerrollen aus. Unter der Überschrift „Beteiligte Personen“ werden alle im Ticket definierten Benutzer in ihrer jeweiligen Funktion ausgegeben.

#### 4.12 Benutzerrollen

In einem Ticket gibt es verschiedene Benutzerrollen (Entwickler, Koordinator, Tester etc.). Sie können diese Benutzerrollen frei wählen. Sinnvollerweise sollten die folgenden Benutzerrollen definiert werden:

- Zugewiesen/ Aktueller Bearbeiter (Wer ist aktuell an der Bearbeitung des Tickets?)
- Entwickler (Wer ist technisch für die Fehlerbehebung zuständig?)
- Tester (Wer ist für den Test zuständig?)
- Reporter (Wer hat den Fehler gemeldet?)

## 5 Individuelle Anpassungen

Der Bugtracker ist ein Produkt mit bestimmten Standardfunktionen. Diese können nicht oder nur eingeschränkt angepasst werden. Um dennoch größtmögliche Individualität zu ermöglichen, stehen im Bugtracker folgende Funktionen zur Verfügung mit denen Sie den Bugtracker an Ihre Bedürfnisse anpassen können:

- Kundeneigene Felder
- Kundendynpro

Die Funktionen und Arbeitsweise wird in den folgenden Kapiteln beschrieben.

### 5.1 Kundeneigene Felder

Mit Hilfe der Transaktion /AKN/BTRK\_C\_CF können Sie Kundeneigene Felder definieren. Sie finden die Transaktion auch im Customizingmenu:

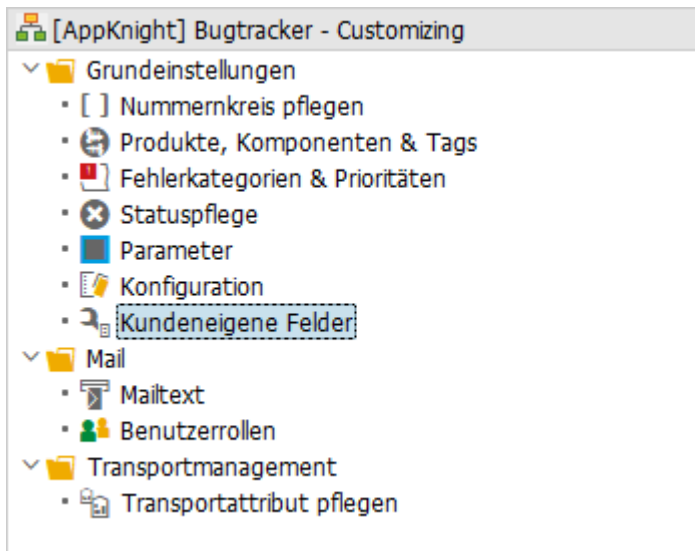


Abbildung 9 Kundeneigene Daten im Customizingsmenu

Kundeneigene Felder werden in jedem Ticket auf der Registerkarte „Kundendaten“ angezeigt.

Es stehen Ihnen 10 Felder zur Verfügung (CF01 – CF10), die Sie nach Belieben definieren können.  
Einbindung in die Ticketsuche

Die Kundeneigenen Felder haben eine eigene Registerkarte: Kundeneigene Felder. Hier sehen Sie alle definierten Felder. Haben Sie keine Felder definiert, dann ist dieser Bereich leer.

[AppKnight] Bugtracker - Ticketsuche

**Selektion Ticketdaten**

Ticketnummer	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Produkt	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Komponente	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Ende Datum	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Fehlerbeschreibung	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Status	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Ticketkategorie	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Priorität	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Auftrag/Aufgabe	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>

Volltextsuche	Eingrenzung Verwaltungsdaten	Benutzer	Kundeneigene Felder
---------------	------------------------------	----------	---------------------

Buchungskreis	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
ABAP Laufzeitfehler	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Kundenfeld 3	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Kundenfeld 4	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Kundenfeld 5	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>
Kundenfeld 6	<input type="text"/>	bis	<input type="text"/>	<input type="button" value="➔"/>

Abbildung 10 Kundeneigene Felder in der Ticketsuche

### 5.1.1 Darstellung im Ticket

Im Ticket selbst werden die Kundeneigenen Felder in der Registerkarte „Kundendaten“ angezeigt:

**Kundeneigene Felder**

Buchungskreis	1000
ABAP Laufzeitfehler	AMDP_EXECUTION_FAILED
Kundenfeld 3	
Kundenfeld 4	
Kundenfeld 5	
Kundenfeld 6	

Abbildung 11 Kundeneigene Daten in dem Ticket

## 5.1.2 Definition

Rufen Sie die Pflege der Kundeneigenen Felder im Bugtracker-Customizingmenu auf oder über die Transaktion /AKN/BTRK\_C\_CF.

Sie müssen entweder eine Beschreibung eingeben oder einen Tabellennamen und Feldnamen definieren, auf den sich das Feld beziehen soll. Das Feld erbt alle Eigenschaften vom Referenzfeld:

- Beschreibung
- Länge
- Suchhilfe
- Fremdschlüsselprüfung

Kundeneigene Felder, die nicht definiert wurden, werden nicht angezeigt.

Die Kundeneigene Bezeichnung übersteuert die Bezeichnung aus dem Referenzfeld.

In unserem Beispiel wurden die Felder im Customizing wie folgt definiert:

[AppKnight] Bugtracker - Kundeneigene Felder			
Custom field name	Beschreibung	Tabellenname	Feldname
CF01		LFB1	BUKRS
CF02		SNAPT	ERRID
CF03	Kundenfeld 3		
CF04	Kundenfeld 4		
CF05	Kundenfeld 5		
CF06	Kundenfeld 6		

Abbildung 12 Pflegedialog Kundeneigene Felder

Achten Sie darauf, dass die Referenz, die Sie benutzen, nicht die Prüftabelle selbst ist, denn dieses hat in der Regel keine Suchhilfe. In unserem Fall haben wir den Buchungskreis verwendet und uns für die Tabelle mit den Lieferanten LFB1 entschieden. Wenn Sie das Feld T001-BUKRS verwenden, dann haben Sie zwar die Bezeichnung und die Länge des Feldes „Buchungskreis“, aber die Suchhilfe fehlt.

Die Kundeneigenen Felder sind generell in jedem Ticket, unabhängig vom Produkt, Bereich oder Status, sichtbar. Wenn Sie produktabhängige Felder nutzen möchten, nutzen Sie am besten die Schlagworte.

## 5.2 Kundeneigenes Dynpro

Sie haben die Möglichkeit, die Tabelle /AKN/BTRK\_BUG mit Appends zu erweitern. Die Felder des Appends können Sie in einer eigenen Funktionsgruppe mit einem eigenen Dynpro einbinden. Sie können je Produkt-ID ein separates Dynpro einbinden.

Die Konfiguration der Kundeneigenen Dynpros erfolgt über die Transaktion /AKN/BTRK\_CSCR oder über das Customizingmenü und den Menüpunkt „Kundendynpro“.



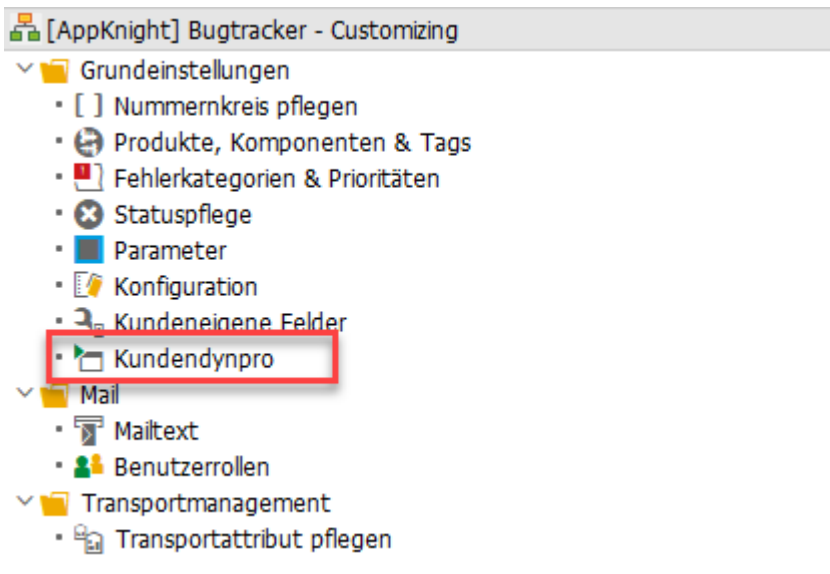


Abbildung 13 Menü "Kundeneigenes Dynpro"

### 5.2.1 Felddefinitionen

Rufen Sie die Transaktion auf und pflegen Sie die Felder analog zu unserem Beispiel wie folgt:

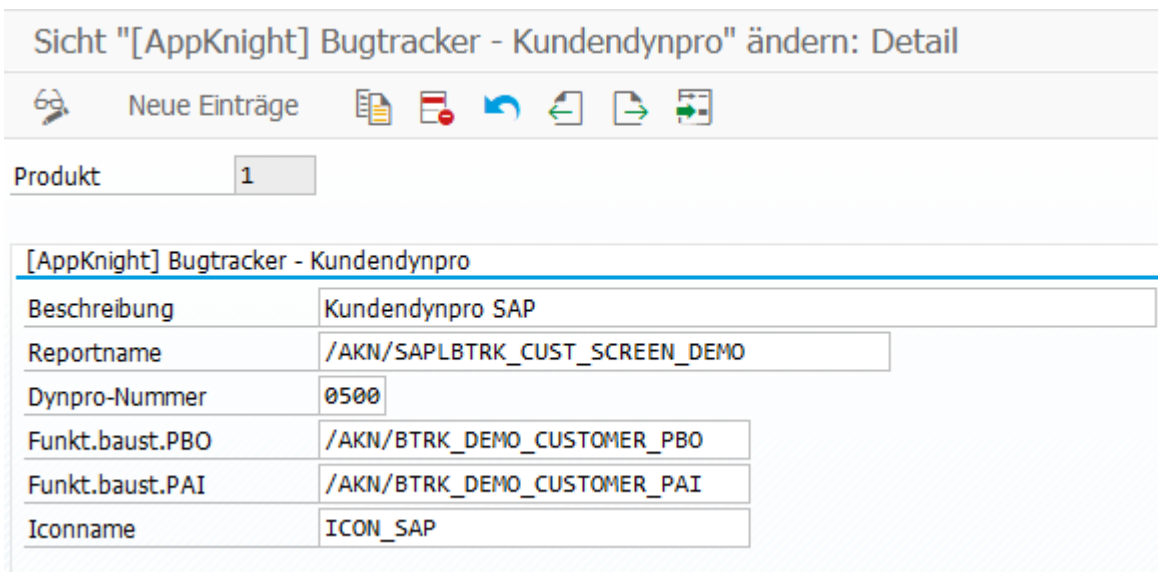


Abbildung 14 Einstellungen für Kundendynpro

Lassen Sie das Feld „Produkt“ leer, um die Einstellungen für alle Tickets zu definieren. Sie können für jedes Produkt gesonderte Einstellungen definieren.

#### 5.2.1.1 Beschreibung

Die Beschreibung erscheint als Text zusammen mit dem Iconname auf der Registerkarte.

#### 5.2.1.2 Reportname

Der Reportname ist der Name der Funktionsgruppe, die Sie für diesen Zweck anlegen müssen. Wie genau die Funktionsgruppe angelegt werden muss, erfahren Sie weiter unten.

### 5.2.1.3 Dynpronummer

Die Dynpronummer, die in Ihrer Funktionsgruppe angelegt sein muss. Das Dynpro muss als *Subscreen* definiert sein.

### 5.2.1.4 Funktionsbaustein PBO

Sie benötigen einen Funktionsbaustein mit dessen Hilfe das Bugtracker-Applikationsobjekt an Ihre Funktionsgruppe übertragen werden kann. Der Funktionsbaustein muss definiert sein wie der Demo-Funktionsbaustein /AKN/BTRK\_DEMO\_CUSTOMER\_PBO. Kopieren Sie diesen in einen Z-Baustein oder legen Sie ihn schnell selbst an:

```
FUNCTION /akn/btrk_demo_customer_pbo.
*-----
*""Local Interface:
* IMPORTING
* REFERENCE(IO_APPLICATION) TYPE REF TO /AKN/CL_BTRK_APPLICATION
*-----
gr_application = io_application. /akn/btrk_dynpro = gr_application->dynpro_from_bug( ).
ENDFUNCTION.
```

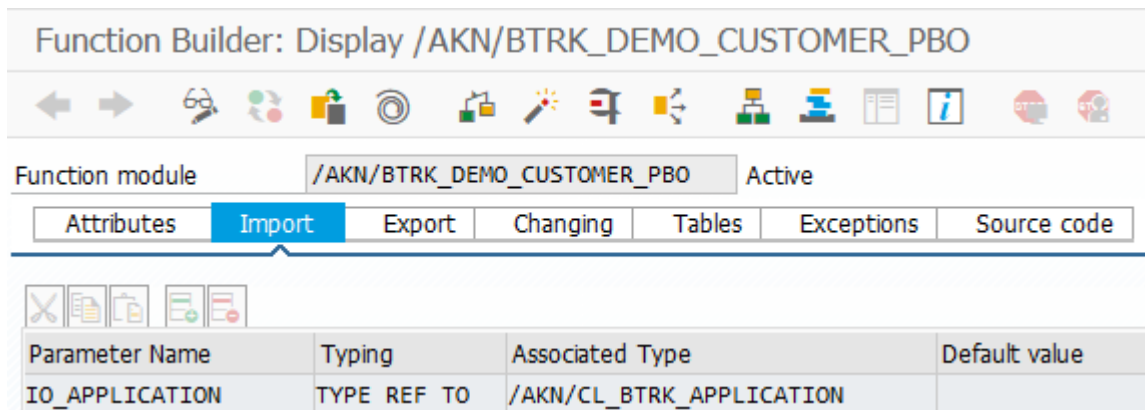


Abbildung 15 Schnittstelle „Import“ des Funktionsbausteins

Im TOP-Include der Funktionsgruppe müssen die folgenden Variablen definiert werden:

```
FUNCTION-POOL /akn/btrk_cust_screen_demo.
TABLES /akn/btrk_dynpro.
DATA gr_application TYPE REF TO /akn/cl_btrk_application.
```

Die Tables-Anweisung ist notwendig, um die Daten des Tickets auf dem Dynpro anzeigen zu können.

Die Referenzvariable GR\_APPLICATION enthält die gesamte Bugtracker-Applikation, mit deren Hilfe Sie an die Daten zum Aktuellen Ticket gelangen.

In der Variablen GR\_APPLICATION->MV\_MAINTENANCE\_MODE steht der aktuelle Bearbeitungsmodus des Tickets. Sie müssen bei Anzeige (Maintenance\_mode = „D“) die Felder auf Ihrem Dynpro entsprechend in den Anzeigemodus versetzen (SCREEN-INPUT = „0“).

#### 5.2.1.5 Funktionsbaustein PAI

Der Parameter ist bisher nicht erforderlich. Sie können das Feld leer lassen.

#### 5.2.1.6 Iconname

Der Iconname wird zusammen mit der Beschreibung in der Registerkarte angezeigt.

### 5.2.2 Realisierungsideen

Im Folgenden zeigen wir Ihnen Möglichkeiten, die Sie auf Ihrem Kundenindividuellen Dynpro realisieren könnten.

#### 5.2.2.1 Absprung in spezielle Transaktionen

Wenn Sie den Bugtracker für die Verwaltung von Berechtigungen verwenden, dann könnten Sie ein Produkt „Berechtigungen“ definieren und für dieses Produkt Buttons zum Absprung in die Berechtigungspflege programmieren.

Wenn Sie die Schlagworte verwenden, um definierte Werte abzuspeichern (Buchungskreis, Reportname, Jobname oder ähnliches), dann können Sie auch hier gezielt Aktionen für die einzelnen Schlagworte definieren:

- Absprung in die Jobdefinition
- Absprung in den Editor zu einem Programm, Funktionsbaustein oder Klasse
- Absprung in die Formulardefinition
- Absprung in das Material, den Kundenbeleg oder die Bestellung per RFC

Genauso gut könnten Sie direkt Fragen auf Ihre häufigsten Probleme einbinden:

- Ist der in den Schlagworten erfasste Benutzer gerade gesperrt oder ist das Kennwort abgelaufen?
- Welche Buchungsperiode ist gerade aktiv?
- Anzeige von fehlerhaften Workflows

#### 5.2.2.2 Anzeige ähnliche Tickets

In diesem Installationsleitfaden haben wir beispielhaft das Kundenfeld 2 als „Name eines Shortdumps“ definiert. Sie könnten auf Ihrem Kundendynpro eine ALV-Grid-Liste mit allen Tickets anzeigen, die den gleichen Shortdump haben. So können Sie dort nach Lösungen suchen, was bei diesem Shortdump zu tun war.

**Tipp:** Definieren Sie einen neuen Status „Erledigt (Lösungsvorschlag)“, der ein Ticket als Musterlösung für wiederkehrende Probleme kennzeichnet.