

Kapitel 4

Customer Exits

In diesem Kapitel zeige ich Ihnen, wie Sie mit Customer Exits arbeiten können. Zu dieser Arbeit gehören die Erweiterung von Standardquelltext sowie die Erweiterung von Menüs und Oberflächen.

Customer Exits sind eine logische Weiterentwicklung der in Kapitel 3, »Modifikationen«, vorgestellten User Exits (Unterprogramme). Sie sind die Vorgänger der in Kapitel 5, »Klassische Business Add-ins«, vorgestellten objektorientierten Business Add-ins (BADIs).

Bei einem Customer Exit muss der Kunde keine Modifikation am SAP-Standardobjekt vornehmen, um beispielsweise ein eigenes Z-Include in eine USEREXIT-Formroutine zu hinterlegen. Stattdessen wird ein von SAP bereitgestellter Funktionsbaustein aufgerufen, der bereits ein solches Z-Include enthält. Wie Sie einen Customer Exit implementieren, zeige ich Ihnen in Abschnitt 4.1.

Customer Exits bieten nicht nur die Möglichkeit, den Quelltext zu erweitern, sondern auch Menüs und Oberflächen. Diese beiden Varianten stelle ich Ihnen in Abschnitt 4.3, »Menüerweiterungen«, und Abschnitt 4.4, »Screen-erweiterungen«, vor.

4.1 Customer Exits implementieren

Abbildung 4.1 zeigt einen solchen Funktionsbaustein (der im Rahmen von Customer Exits auch *Funktions-Exit* genannt wird) mit dem bereits enthaltenen Include ZXVVAU04.

Funktions-Exit

Dieses Z-Include wird zwar eingebunden, existiert im System aber noch nicht. Der Kunde kann es durch einen Doppelklick anlegen und darin anschließend die gewünschte Zusatzlogik programmieren. Dadurch arbeitet der Kunde im Kundennamensraum Z, und es findet keine Modifikation des SAP-Standards statt. Die zusätzliche Logik wird durch den Standard-funktionsbaustein aufgerufen (hier EXIT_SAPMV45A_002).

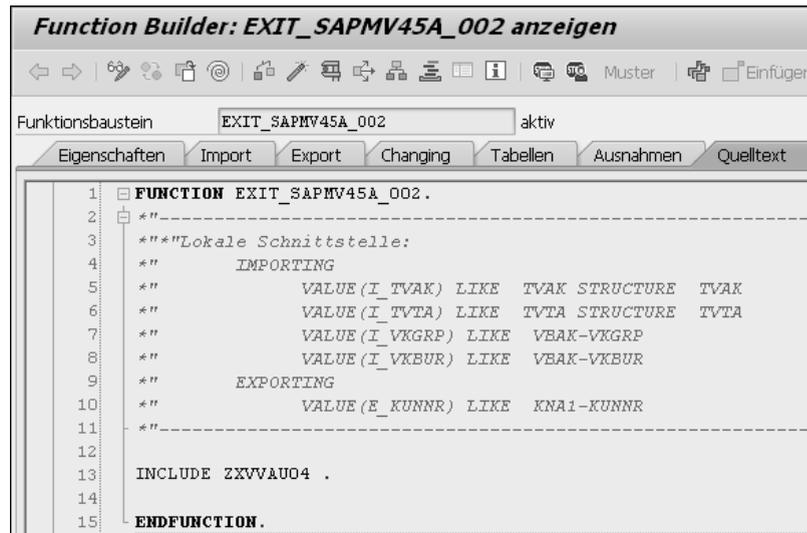


Abbildung 4.1 Ein Customer-Exit-Funktionsbaustein

Aufruf eines Customer Exits

Die Funktions-Exits werden über die Anweisung `CALL CUSTOMER-FUNCTION` aufgerufen. Der Funktionsbaustein wird dabei über eine dreistellige Nummer identifiziert.

Wie bei einem Funktionsbaustein üblich, werden `IMPORTING`-, `EXPORTING`-, `CHANGING`- und `TABLES`-Parameter bereitgestellt. SAP stellt über diese Schnittstellen die Datenobjekte aus den Standardtransaktionen bereit, auf die Sie in Ihrer Erweiterung zugreifen und die Sie gegebenenfalls manipulieren können. Listing 4.1 stellt die Syntax eines solchen Customer-Exit-Aufrufes dar.

```
CALL CUSTOMER-FUNCTION 'XXX'
  [EXPORTING ...]
  [IMPORTING ...]
  [CHANGING ...]
  [TABLES ...]
  [EXCEPTIONS ...].
```

Listing 4.1 Aufruf eines Customer Exits

Der Funktionsbaustein wird nach dem folgenden Schema benannt: `EXIT_<Programmname>_<3stelligeZahl>`. Der Aufruf funktioniert jedoch nur, wenn die dahinterliegende Erweiterung auch implementiert und aktiviert wurde. Ein nicht aktivierter Customer Exit wird nicht aufgerufen. Sie als Entwickler auf Kundenseite müssen sich daher darum kümmern, den jeweiligen Customer Exit zu aktivieren.

Für die Arbeit mit Customer Exits stehen Ihnen zwei Transaktionen zur Verfügung:

- **Transaktion SMOD (SAP-Erweiterungen)**
Diese Transaktion wird von SAP genutzt, um Customer Exits anzulegen. Sie können die Transaktion benutzen, um Erweiterungen zu finden, die Komponenten der Erweiterungen zu sehen und zu analysieren.
- **Transaktion CMOD (Projektverwaltung von SAP-Erweiterungen)**
Mit dieser Transaktion können Sie die Erweiterungen und ihre Komponenten in einem Projekt implementieren und die Erweiterung aktivieren.

In diesem Abschnitt zeige ich Ihnen, wie Sie mithilfe dieser beiden Transaktionen einen herkömmlichen Customer Exit für eine Quelltexterweiterung implementieren können. Im Folgenden soll bei der Anlage eines Kundenauftrags (Transaktion VA01, Kundenauftrag anlegen) der Auftraggeber vorgelegt werden. Abbildung 4.2 stellt das Ergebnis dieser Erweiterung dar: Direkt nach dem Aufruf der Bestellanlage steht im Feld **Auftraggeber** bereits der Wert »123456«.

Beispiel:
Transaktion VA01
erweitern



Abbildung 4.2 Vorbelegung eines Standardfeldes mithilfe eines Customer Exits

Um dies umzusetzen, müssen Sie die Erweiterung V45A0002 implementieren. Diese Erweiterung können Sie sich in Transaktion SMOD anzeigen lassen. Tragen Sie hierzu, wie in Abbildung 4.3 gezeigt, den Namen der Erweiterung auf der Startoberfläche von Transaktion SMOD ein, und klicken Sie anschließend auf die Schaltfläche **Anzeigen**.

Erweiterung
anzeigen

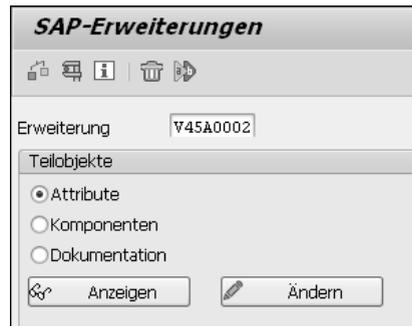


Abbildung 4.3 Transaktion SMOD

Sie gelangen nun zu der Detailsicht, die Sie in Abbildung 4.4 sehen. In dieser stehen nicht viele Informationen, in der Regel gibt Ihnen hier nur der **Kurztext** Auskunft darüber, wozu die Erweiterung verwendet werden kann. In diesem Beispiel deutet der Kurztext darauf hin, dass diese Erweiterung der Vorbelegung eines Auftraggebers in einem Verkaufsbeleg dient.



Abbildung 4.4 Attribute einer SAP-Erweiterung in der Transaktion SMOD



Migrierte Customer Exits

Wenn Sie einen Customer Exit öffnen, kann es vorkommen, dass Sie wie in Abbildung 4.5 direkt die folgende Meldung erhalten:

»Die Erweiterung ... wurde bereits in die BAdI-Definition ... migriert.«

Dies sehen Sie auch daran, dass in den Eigenschaften der Status **migriert** steht. In einem solchen Fall sollten Sie nicht den Customer Exit, sondern das BAdI verwenden.

Über die Transaktion SE18 können Sie dieses BAdI öffnen. Wie Sie es implementieren können, beschreibe ich in Kapitel 5, »Klassische Business Add-ins«.

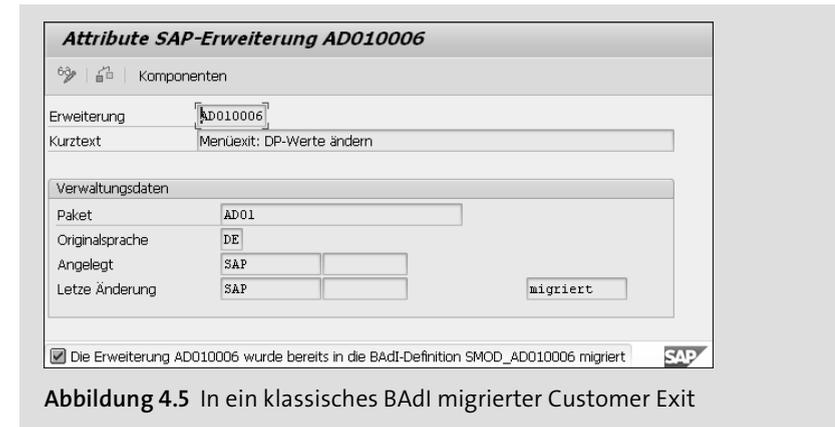


Abbildung 4.5 In ein klassisches BAdI migrierter Customer Exit

Interessant sind an dieser Stelle vor allem die Komponenten der Erweiterung, also die Information, was die Erweiterung eigentlich anbietet. Zur Ansicht der Komponenten wechseln Sie, indem Sie auf die Schaltfläche **Komponenten** in der Menüleiste klicken. In unserem Beispiel bietet die Erweiterung lediglich einen Funktions-Exit (EXIT_SAPMV45A_002) an, wie Sie in Abbildung 4.6 sehen. Es werden von ihr keine Funktionscodes für Menüerweiterungen, keine Dynpro-Bereiche für Screenerweiterungen und auch keine Includes (für die Erweiterung von Datenbanktabellen) angeboten. Das heißt, mit dieser Erweiterung können Sie lediglich den Quelltext über den angebotenen Funktions-Exit erweitern.

Komponenten der Erweiterung

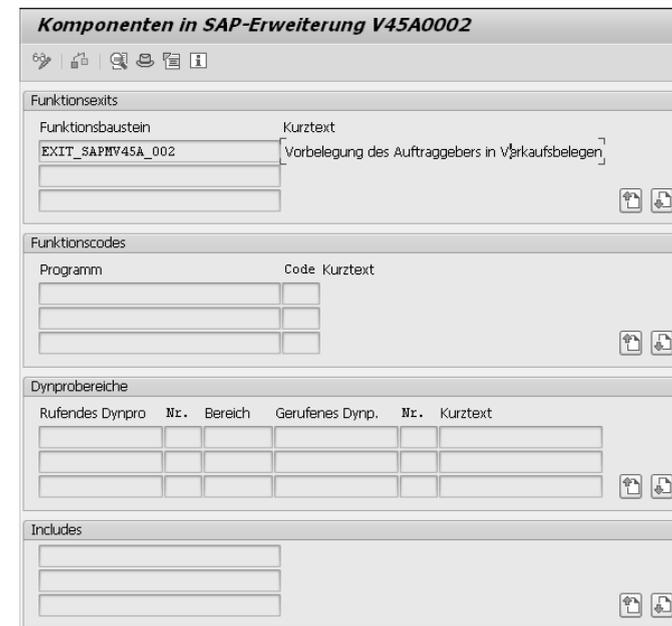


Abbildung 4.6 Anzeige der Komponenten eines Customer Exits

In Transaktion SMOD können Sie Erweiterungen nur finden und sich die Komponenten und die Dokumentation der Erweiterung anzeigen lassen. Wie an diesem Beispiel veranschaulicht, haben aber nicht alle Erweiterungen eine Dokumentation.

Erweiterung implementieren

Das Implementieren einer solchen Erweiterung findet dagegen in Transaktion CMOD statt. Folgende Schritte sind hierfür notwendig:

1. Anlage eines neuen Erweiterungsprojekts und Zuordnen einer Kurzbeschreibung
2. Zuweisen des Customer Exits, das innerhalb des Erweiterungsprojekts implementiert werden soll
3. Implementierung der Komponenten des zugewiesenen Customer Exits
4. Aktivieren des Erweiterungsprojekts

Erweiterungsprojekt anlegen

Wir beginnen mit der Anlage eines Erweiterungsprojekts:

1. Wechseln Sie in die Transaktion CMOD, und geben Sie auf der in Abbildung 4.7 dargestellten Startoberfläche einen Namen für ein neues Erweiterungsprojekt ein (hier »ZSD_VA«). Den Namen können Sie frei wählen, auch außerhalb Ihres Kundennamensraumes.



Abbildung 4.7 Anlage eines neuen Erweiterungsprojekts in Transaktion CMOD

2. Klicken Sie anschließend auf die Schaltfläche **Anlegen**, und vergeben Sie auf der nächsten Oberfläche im Feld **Kurztext** eine Kurzbeschreibung für das Erweiterungsprojekt (siehe Abbildung 4.8).

Erweiterung zuordnen

Klicken Sie anschließend auf die Schaltfläche **Zuordnung Erweiterung** in der Funktionsleiste. Hier können Sie nun, wie in Abbildung 4.9 dargestellt, diesem Erweiterungsprojekt die eben gefundene Erweiterung V45A0002 zuordnen. Tragen Sie die Erweiterung in der ersten Spalte **Erweiterung** ein, und bestätigen Sie die Eingabe mit .



Abbildung 4.8 Anzeige der Attribute eines Erweiterungsprojekts

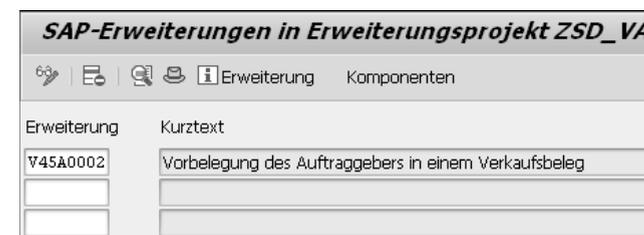


Abbildung 4.9 Zuordnen des Customer Exits zum Erweiterungsprojekt

Zuordnen von Erweiterungen

Sie können beliebig viele Customer Exits auf diese Weise einem Erweiterungsprojekt zuordnen, um zusammengehörige Implementierungen zu sammeln. Sie haben aber natürlich auch die Möglichkeit, für jedes Customer Exit ein neues Erweiterungsprojekt anzulegen.

Nun können Sie die Komponenten der Erweiterung implementieren:

1. Wechseln Sie über die Schaltfläche **Komponenten** (siehe Abbildung 4.9) in die Komponenten der zugeordneten Erweiterungen. Wie Sie in Abbildung 4.10 sehen können, ist für unser Beispiel nur der Funktions-Exit EXIT_SAPMV45A_002 aufgelistet, also der Funktions-Exit, der für die Vorbelegung des Auftraggebers in einem Verkaufsbeleg zuständig ist. Stellt ein Customer Exit mehrere Funktions-Exits oder sogar Menü- und Screenerweiterungen bereit, sähen Sie diese hier ebenfalls aufgelistet.



Komponenten implementieren

ZSD_VA anzeigen			
Zuordnung Erweiterung Erweiterung			
Projekt			ZSD_VA Erweiterung für VA
Erweiterung	Impl	Bsp	V45A0002 Vorbelegung des Auftraggebers in einem Verkaufsbeleg
Funktionsexit			EXIT_SAPMV45A_002

Abbildung 4.10 Auflistung der implementierbaren Komponenten in einem Erweiterungsprojekt

- Diesen Funktions-Exit gilt es jetzt zu implementieren. Klicken Sie hierzu doppelt auf den Namen des Funktionsbausteins (EXIT_SAPMV45A_002).
- Dadurch gelangen Sie in den in Abbildung 4.11 dargestellten Function Builder.

```

1 FUNCTION EXIT_SAPMV45A_002.
2
3  ***"Lokale Schnittstelle:"
4  IMPORTING
5      VALUE(I_TVAK) LIKE TVAK STRUCTURE TVAK
6      VALUE(I_TVTA) LIKE TVTA STRUCTURE TVTA
7      VALUE(I_VKGRP) LIKE VBAK-VKGRP
8      VALUE(I_VKBUR) LIKE VBAK-VKBUR
9  EXPORTING
10     VALUE(E_KUNNR) LIKE KNA1-KUNNR
11
12
13  INCLUDE ZXVVAU04 .
14
15  ENDFUNCTION.

```

Abbildung 4.11 Der Funktions-Exit mit dem enthaltenen Z-Include

- Klicken Sie hier doppelt auf den Namen des Includes in Zeile 13 (hier ZXVVAU04).
- Bestätigen Sie die folgende Warnung (je nach Systemeinstellung als Pop-up-Fenster oder Meldung in der Statusleiste) mit :

»Programmnamen ZX... sind für Includes v. Exit-Funktionsgruppen reserviert«
- Dadurch wird das Include angelegt (siehe Abbildung 4.12). Innerhalb des Includes können Sie nun Ihre zusätzliche Logik programmieren.

```

1  *%
2  *%  Include          ZXVVAU04
3  *%
4
5  e_kunnr = '123456'.

```

Abbildung 4.12 Implementiertes Include für den Funktions-Exit

- In Abbildung 4.12 wird zur Demonstration der Funktionsweise der Auftraggeber »123456« als Wert für den Export-Parameter E_KUNNR vorbelegt. Natürlich können Sie hier auch eine komplexe Ermittlungslogik implementieren.
- Aktivieren** Sie das Include, und wechseln Sie anschließend zurück in die Erweiterung. In der Anzeige der Komponenten finden Sie nun hinter dem Funktions-Exit einen grünen Haken , siehe Abbildung 4.13) vor. Für die Erweiterung selbst wird allerdings darüber noch eine rote Ampel angezeigt. Dies bedeutet, dass die Erweiterung nicht aktiviert ist und daher auch nicht ausgeführt wird.

ZSD_VA anzeigen			
Zuordnung Erweiterung Erweiterung			
Projekt			ZSD_VA Erweiterung für VA
Erweiterung	Impl	Bsp	V45A0002 Vorbelegung des Auftraggebers in einem Verkaufsbeleg
Funktionsexit			EXIT_SAPMV45A_002

Abbildung 4.13 Anzeige des implementierten Funktions-Exits in der Komponentenliste des Erweiterungsprojekts

Im nächsten Schritt müssen Sie das Erweiterungsprojekt aktivieren. Wechseln Sie hierfür über die Schaltfläche **Anzeigen <-> Ändern** in den Änderungsmodus, und klicken Sie anschließend auf **Aktivieren** . Alternativ können Sie, wie in Abbildung 4.14 gezeigt, auch zurück zum Startbildschirm der Transaktion CMOD wechseln, Ihr Projekt angeben und dieses über das entsprechende Icon in der Funktionsleiste aktivieren.

Erweiterungs-
projekt aktivieren



Abbildung 4.14 Aktivieren und Deaktivieren eines Customer Exits



Aktivieren und deaktivieren

Sie können den Customer Exit über **Projekt deaktivieren** (🔍) jederzeit wieder inaktiv setzen. Dadurch wird die Erweiterung nicht durchlaufen, und der Standard ist wiederhergestellt. Dies sollten Sie auch immer tun, bevor Sie die Implementierung des Customer Exits ändern, um eventuellen Datenschiefezustand oder Programmabbrüche zu vermeiden.

Wenn Sie nun zurück in die Komponentenanzeige des Erweiterungsprojekts wechseln, werden Sie sehen, dass ein grünes Ampelsymbol für die Erweiterung angezeigt wird (siehe Abbildung 4.15). Dies bedeutet, dass diese Erweiterung jetzt beim Durchlaufen des Standardcodes aufgerufen wird.

ZSD_VA anzeigen			
🔍 Zuordnung Erweiterung ⓘ Erweiterung			
Projekt	<input type="checkbox"/>		ZSD_VA Erweiterung für VA
Erweiterung	Impl <input type="checkbox"/>	Bsp	V45A0002 Vorbelegung des Auftraggebers in einem Verkaufsbeleg
Funktionsexit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EXIT_SAPMV45A_002

Abbildung 4.15 Anzeige einer aktiven Erweiterung in der Komponentenliste des Erweiterungsprojekts

Erweiterung testen

Sie können die Erweiterung testen, indem Sie die Transaktion VA01 aufrufen. Wenn Sie hier einen Auftrag anlegen, wird, wie in Abbildung 4.2 dargestellt, der Wert »123456« im Feld **Auftraggeber** vorbelegt sein.

Das Durchlaufen der Erweiterung können Sie auch nachvollziehen, indem Sie einen Breakpoint in der Zeile mit dem Include setzen (siehe Abbildung 4.16) und anschließend den Debugger starten. Lassen Sie sich hier die lokalen Variablen und Parameter über die Registerkarte **Locals** anzeigen. Hier können Sie genauer erfahren, welche Daten an den Funktionsbaustein übertragen werden, und gegebenenfalls Werte ändern, um die Funktionsweise der Transaktion besser zu verstehen.

Damit haben Sie erfolgreich ein Customer Exit mit einem Funktions-Exit implementiert und aktiviert. Die hier vorgestellte Vorgehensweise ist für alle Customer Exits dieselbe. Für Menüerweiterungen und Screenerweiterungen ist sie allerdings etwas umfangreicher, wie Sie in Abschnitt 4.3 und Abschnitt 4.4 feststellen werden.

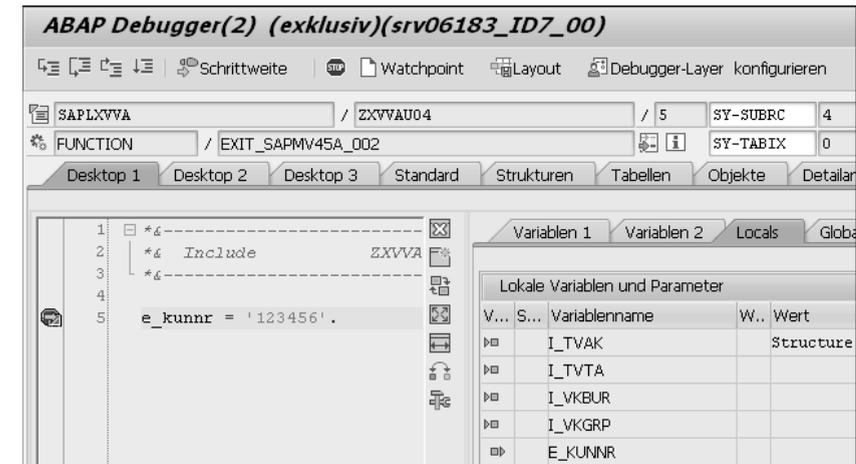


Abbildung 4.16 Anzeige des Funktions-Exits im Debugger

4.2 Funktions-Exits finden

Funktions-Exits von Customer Exits, über die Sie Quelltexterweiterungen durchführen können, können Sie im System auf unterschiedlichen Wegen finden. Dazu gehören folgende Möglichkeiten:

- Tabelle MODSAP
- **F4**-Hilfe in Transaktion SMOD (anhand der Kurzbeschreibung oder gegebenenfalls der Dokumentation)

Um den in Abschnitt 4.1 vorgestellten Customer Exit zur Vorbelegung des Auftraggebers in der Tabelle MODSAP zu finden, müssen Sie zuerst die Aufrufstelle in dem Programm finden, das von der zu erweiternden Transaktion VA02 ausgeführt wird. Welches Programm das ist, können Sie wie folgt herausfinden:

Aufrufstelle im Programm

1. Öffnen Sie Transaktion VA02, und wechseln Sie über den Pfad **System • Status** im Hauptmenü in die Statussicht dieser Transaktion (siehe Abbildung 4.17).
2. Mit einem Doppelklick auf das Feld **Programm(Dynpro)** gelangen Sie in das Programm SAPMV45A im Editor.
3. Rufen Sie hier die Suchen-Funktion (🔍) aus der Funktionsleiste auf.
4. Über das Fenster **Suchen/Ersetzen** durchsuchen Sie den Quelltext nach dem Begriff »CALL CUSTOMER-FUNCTION« (siehe Abbildung 4.18). Sie können diese Suche global im Programm oder auch global im Rahmenprogramm durchführen.



Abbildung 4.17 Ermitteln des Programms im Status einer Transaktion

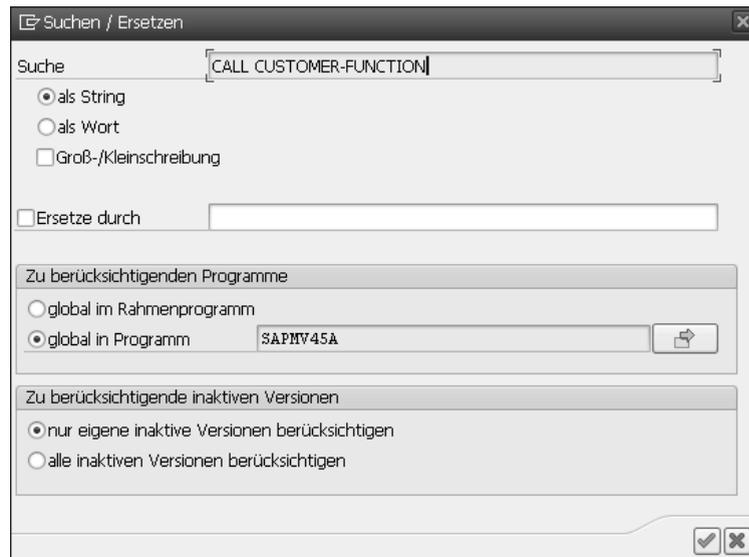


Abbildung 4.18 Suche nach Customer Exits im Quelltext

- In der Ergebnisliste, die Sie in Abbildung 4.19 sehen, stoßen Sie auf den Aufruf im Include MV45AFOA_AG_VORSCHLAGEN. Das klingt ganz so, als könnten Sie sich damit den Auftraggeber (AG) vorschlagen lassen.

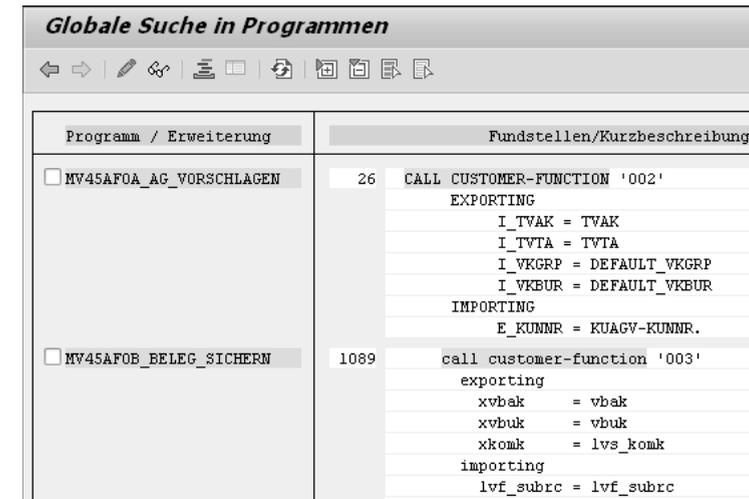


Abbildung 4.19 Gefundene Aufrufstellen von Customer Exits

Wie Sie in Abbildung 4.19 sehen können, gibt der Aufruf nur eine Nummer »002« vor (und nicht den tatsächlichen Namen des Funktionsbausteins). Nun gilt es, von dieser Nummer den Namen der Erweiterung und des Funktionsbausteins abzuleiten. Um dies zu bewerkstelligen, hilft es zu wissen, dass sich der Name des Funktionsbausteins aus dem Präfix EXIT_, gefolgt vom Namen des Hauptprogramms und der Nummer des Aufrufes zusammensetzt. In diesem Beispiel lautet der Name des Funktionsbausteins also EXIT_SAPMV45A_002.

Diesen Funktionsbaustein könnten Sie nun in der Transaktion SE37 suchen und über die Dokumentation des Funktionsbausteins den Namen des zugehörigen Customer Exits herausfinden. Allerdings sind diese Dokumentationen meistens nicht gepflegt. In diesem Fall können Sie wie folgt vorgehen, um den Namen der Erweiterung herauszufinden:

- Rufen Sie Transaktion SE11 (oder SE16N) und im Anschluss die Datenbanktabelle MODSAP auf.
- Tragen Sie im dritten Feld **Erweiterung** (nicht im gleichnamigen ersten Feld) den Namen des Funktionsbausteins »EXIT_SAPMV45A_002« ein, wie in Abbildung 4.20 gezeigt.
- Starten Sie die Suche über die Schaltfläche **Ausführen** (F5).
- In der Ergebnisanzeige steht in der einzigen Zeile im ersten Feld nun der Name der Erweiterung (siehe Abbildung 4.21). In unserem Beispiel ist das die Erweiterung V45A0002. Wichtig ist hier auch der Typ der Komponente. Suchen Sie nach dem Typ »E« für Funktions-Exits (in der Spalte **Typ Komponente**).

Funktionsbaustein ableiten

Tabelle MODSAP

Feldname	O.	Von-Wert	Bis-Wert	Mehr	Ausgabe	Technischer Name
Erweiterung					<input checked="" type="checkbox"/>	NAME
Typ Komponente					<input checked="" type="checkbox"/>	TYP
Erweiterung		EXIT_SAPMV45A_002			<input checked="" type="checkbox"/>	MEMBER
Kurztext					<input checked="" type="checkbox"/>	MODTEXT

Abbildung 4.20 Suche nach Customer Exits in der Tabelle MODSAP

Erweiterung	Typ Komponente	Erweiterung
V45A0002	E	EXIT_SAPMV45A_002

Abbildung 4.21 Gefundene Customer Exits in der Tabelle MODSAP

F4-Hilfe Auch über die **[F4]**-Hilfe in Transaktion SMOD haben Sie die Chance, den passenden Customer Exit zu finden. Öffnen Sie dazu die Transaktion SMOD, und geben Sie im Eingabefeld **Erweiterung** einen Teil des Programmnamens ein, in unserem Beispiel »VA45*« für das Programm SAPMV45A (siehe Abbildung 4.22). Diese Suche wird nicht immer zu eindeutigen Ergebnissen führen. Es fangen z. B. sehr viele Programme mit M* an (zweithäufigster Anfang eines Programmnamens nach SAP*).

Exitname	Kurzbeschreibung
V45A0001	Ermittlung der alternativen Materialien für Produktselekt
V45A0002	Vorbelegung des Auftraggebers in einem Verkaufsbeleg
V45A0003	Collector for customer function modulpool MV45A
V45A0004	Kopieren Verpackungsvorschlag

Abbildung 4.22 Suche nach Customer Exits in Transaktion SMOD

Eine höhere Chance haben Sie in der Regel über das Eingabefeld **Kurztext** im Repository-Infosystem. Dieses rufen Sie auf, indem Sie auf die Schaltfläche **Neue Selektion** in dem Ergebnisfenster aus Abbildung 4.22 klicken. Im Repository-Infosystem können Sie sich sogar noch mehr Selektionsfelder einblenden lassen, indem Sie in dem in Abbildung 4.23 dargestellten Pop-up-Fenster auf die Schaltfläche **Alle Selektionen** in der Funktionsleiste unten klicken. Abbildung 4.23 zeigt die Ansicht mit allen Selektionsparametern für Customer Exits. Über diese Selektionsparameter können Sie jetzt anhand von Schlagwörtern, z. B. »Verkaufsbeleg«, nach einem zu Ihrer Transaktion passenden Customer Exit suchen.

Repository-Infosystem

Exitname	Kurztext	Paket	Anwendungskomponente
V45A*			

Abbildung 4.23 Suche nach Customer Exits im Repository-Infosystem

4.3 Menüerweiterungen

Das Ziel von Menüerweiterungen ist es, den GUI-Status eines Dynpros um neue, eigene Funktionseinträge zu erweitern. Der GUI-Status definiert alle Menüs und alle Funktionen, die sich auf einem Dynpro in der Funktionsleiste finden. Er ordnet jeder Funktion einen Funktionscode zu, auf den Sie im PAI-Modul eines Dynpros (PAI = Process After Input, also das durch eine Benutzeraktion ausgelöste Ereignis) reagieren können.

GUI-Status erweitern

Abbildung 4.24 zeigt einen so erweiterten Status der Transaktion VCO1 (Kontrakte anlegen). Die Funktion **Meine Funktion** im Menü **Informationen** wurde hier mithilfe des Menü-Exits eines Customer Exits hinzugefügt.

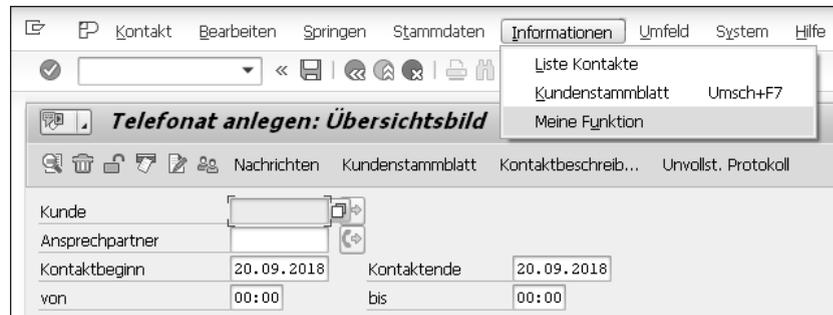


Abbildung 4.24 Eine implementierte Menüerweiterung in der Transaktion VCO1

4.3.1 Funktionscode und Status der Menüerweiterung finden

Funktionscodes

Wenn innerhalb eines Customer Exits Funktionscodes für die Menüerweiterung definiert wurden, werden diese innerhalb der Komponentenanzeige der Erweiterung in der Transaktion SMOD im Bereich **Funktionscodes** angezeigt. Sind keine Funktionscodes innerhalb eines Customer Exits definiert, so werden Ihnen hier auch keine angezeigt. In Abbildung 4.25 ist dies für die Erweiterung V43A5X1 dargestellt.

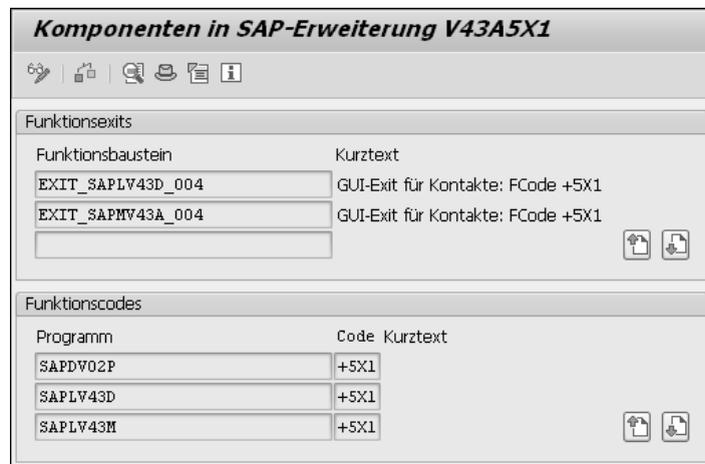


Abbildung 4.25 Funktionscodes in der Komponentenliste eines Customer Exits

Über solche Funktionscodes können Sie die aufgerufene erweiterte Funktion identifizieren. Wenn der Benutzer in der erweiterten Transaktion auf die Menüerweiterung klickt, wird der für diese Funktion definierte Funkti-

onscode in das Systemfeld SY-UCOMM geschrieben. Dieses können Sie letztlich in den bereitgestellten Funktionen des Customer Exits auswerten und darauf reagieren.

Insgesamt definiert diese Erweiterung vier Funktionscodes. Es werden allerdings nur drei in der Liste angezeigt. Über einen Klick auf die Schaltfläche **Ab** (📄) können Sie sich den vierten anzeigen lassen (siehe Abbildung 4.26).

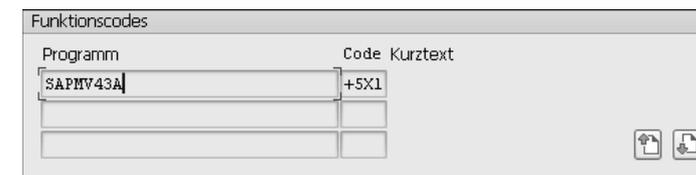


Abbildung 4.26 Ein weiterer definierter Funktionscode

Streng genommen ist hier nur ein Funktionscode (hier +5X1) definiert, der allerdings übergreifend in insgesamt vier Programmen verwendet wird. Alle Funktionscodes fangen mit »+« an, was andeutet, dass es sich um Platzhalter handelt, die durch Ihre eigenen Funktionen ersetzt werden können. Dazu ist es zunächst wichtig herauszufinden, wo im Status der einzelnen Programme diese Funktionscodes eigentlich vorkommen.

Dies können Sie herausfinden, indem Sie doppelt auf einen Funktionscodeeintrag klicken (beispielsweise auf den Eintrag in Abbildung 4.26). Dadurch gelangen Sie zum Menu Painter, den Sie in Abbildung 4.27 sehen. Hier können Sie unterhalb des Knotens **Status** sehen, in welchen GUI-Status der Funktionscode +5X1 definiert ist.

Funktionscodes
lokalisieren

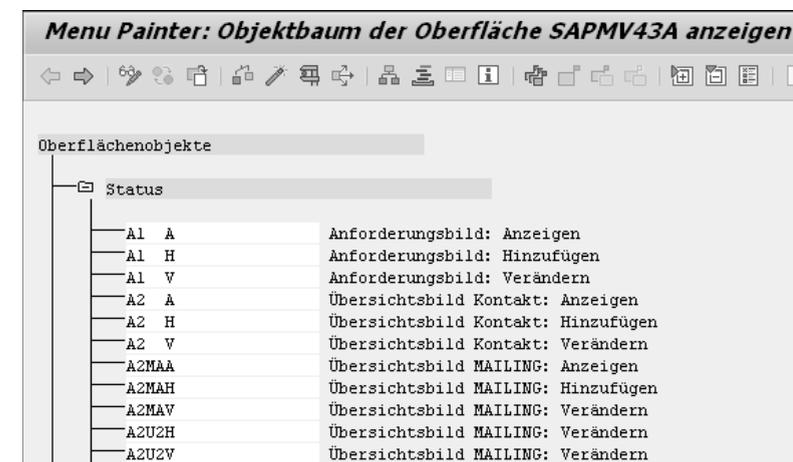


Abbildung 4.27 Anzeige der GUI-Status, in denen der Funktionscode +5X1 im Programm SAPMV43A definiert ist

Der Funktionscode in unserem Beispiel ist in deutlich mehr als einem GUI-Status definiert. Die hinter den einzelnen Status angezeigten Kurzbeschreibungen lassen einen oft schon erahnen, auf welcher Oberfläche (und damit in welcher Transaktion) der Funktionscode verwendet wird.

Richtigen Status auswählen

In diesem Kontext ist es wichtig zu wissen, welche Oberfläche (also welches Dynpro) in welcher Transaktion Sie erweitern wollen. Wollen Sie beispielsweise den Hauptbildschirm der Transaktion VC01 bearbeiten, können Sie den zugehörigen Status herausfinden, indem Sie diesen Bildschirm aufrufen und die Statusübersicht über den Menüpfad **System • Status** aufrufen. Hier finden Sie den **Oberflächenstatus** im Bereich **SAP-Daten • Repository-Daten**. In unserem Beispiel müssen wir also den GUI-Status **A1 H** anpassen (siehe Abbildung 4.28).

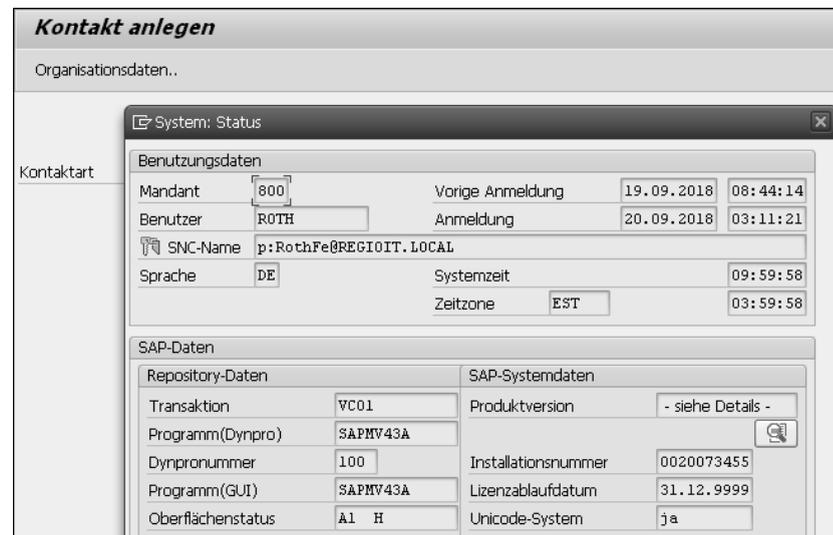


Abbildung 4.28 Der Status der Transaktion VC01 (Kontrakte anlegen)

Um nun herauszufinden, wo innerhalb dieses Status der Funktionscode +5X1 definiert ist, öffnen Sie die Detailsicht zu diesem Status über einen Doppelklick auf den Namen (hier **A1 H**). In dieser Detailsicht können Sie nun die einzelnen Bereiche (**Menüleiste**, **Drucktastenleiste**, **Funktionstasten**) nach dem Funktionscode +5X1 durchsuchen (siehe Abbildung 4.29). In diesem Fall befindet sich der Funktionscode +5X1 im Bereich **Menüleiste** unterhalb des Hauptmenüs **Informationen**. Damit haben Sie herausgefunden, dass der Funktionscode +5X1 innerhalb der Erweiterung V43A5X1 für den zu ergänzenden Eintrag unter dem Hauptmenü **Informationen** zuständig ist.

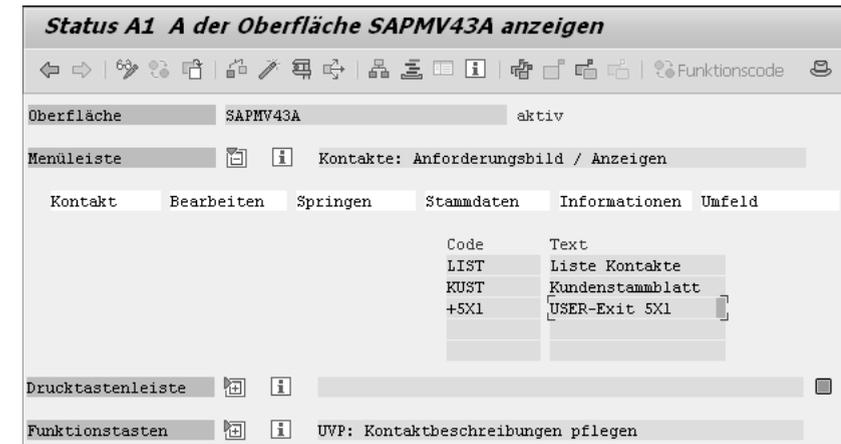


Abbildung 4.29 Der hinterlegte Funktionscode +5X1 im Status der Oberfläche

Wenn Sie die anderen Bereiche durchsuchen, werden Sie unter dem Hauptmenü **Springen** auch den Funktionscode +3X1 und unter dem Hauptmenü **Umfeld** den Funktionscode +6X1 finden (siehe Abbildung 4.30). Auch diese sind an dem Pluszeichen als Platzhalter für eigene Funktionen zu erkennen.

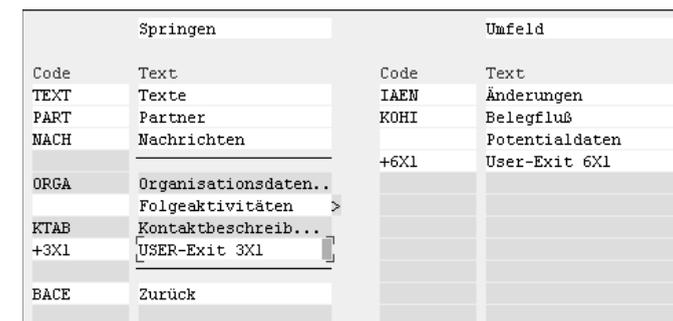


Abbildung 4.30 Weitere definierte Menüerweiterungen in einem Status

Sie sind allerdings nicht in dem gleichen Customer Exit definiert, wie in Abbildung 4.31 zu erkennen, sondern in Customer Exits mit ähnlichen Namen (die jeweils die Nummer des Funktionscodes enthalten).

Diese klare Trennung kann für andere Transaktionen aber auch anders ausschauen. So kann mitunter auch ein Customer Exit existieren, der mehrere Funktionscodes für mehrere verschiedene Transaktionen enthält. Einen solchen Menü-Exit gilt es nun zu implementieren. Die einzelnen Schritte der Implementierung zeige ich Ihnen im folgenden Abschnitt.

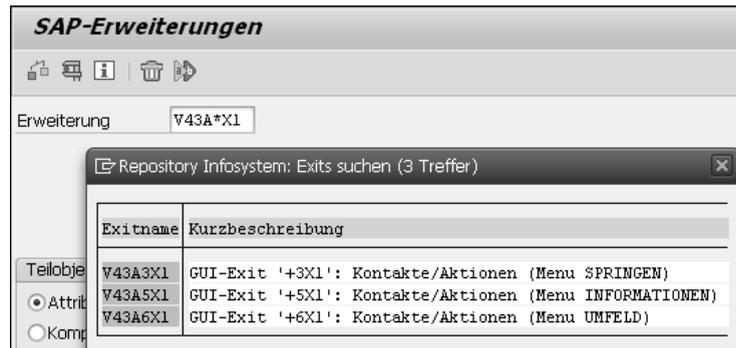


Abbildung 4.31 Ein Customer Exit für jeden Menü-Exit

4.3.2 Menüerweiterungen implementieren

Neue Funktion
hinzufügen

In diesem Abschnitt zeige ich Ihnen, wie Sie den Funktionscode +5X1 der im vorangegangenen Abschnitt vorgestellten Erweiterung V43A5X1 für die Transaktion VCO1 implementieren, um die Funktion **Meine Funktion** im Menü anzuzeigen (siehe Abbildung 4.24). Wenn ein Anwender diese Funktion auswählt, soll das Programm darauf reagieren.

1. Rufen Sie dazu die Transaktion CMOD auf, und legen Sie ein neues Erweiterungsprojekt an. Den Namen des Projekts können Sie frei wählen, in diesem Beispiel nenne ich es »ZVC_MENU«.
2. Geben Sie einen **Kurztext** zur Beschreibung des Projekts ein, und klicken Sie anschließend auf die Schaltfläche **Zuordnung Erweiterung** in der Funktionsleiste (siehe Abbildung 4.32).



Abbildung 4.32 Attribute des angelegten Erweiterungsprojekts ZVC_MENU

3. Tragen Sie, wie in Abbildung 4.33 gezeigt, die Erweiterung »V43A5X1« in die Liste der Erweiterungszuordnungen ein, und bestätigen Sie Ihre Eingabe mit .
4. Wechseln Sie danach über die Schaltfläche **Komponenten** in der Menüleiste in die in Abbildung 4.34 dargestellte Komponentenansicht.
5. Klicken Sie hier im Bereich **Menüexit** doppelt auf den Eintrag +5X1 für das Programm SAPMV43A.



Abbildung 4.33 Zuordnen der Erweiterung für die Menüerweiterung

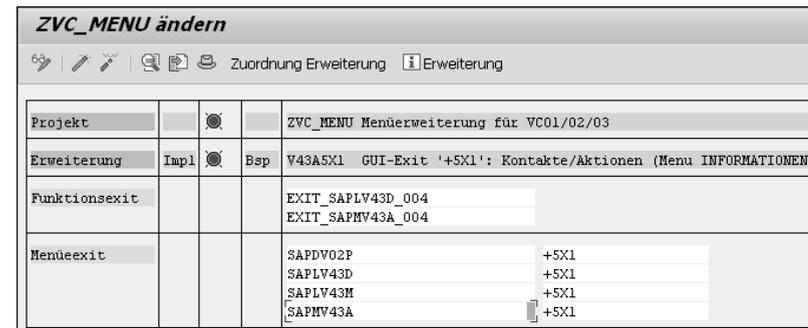


Abbildung 4.34 Anzeige der Menü-Exits in der Komponentenliste

6. Dadurch erscheint das Pop-up-Fenster aus Abbildung 4.35, in dem Sie nun einen **Funktionstext** für den Funktionscode vergeben können, in unserem Beispiel »Meine Funktion«. Dieser Text wird im Menü angezeigt.
7. Nach einem Klick auf **Übernehmen** haben Sie den Funktionscode +5X1 für das Programm SAPMV43A erfolgreich aktiviert.



Abbildung 4.35 Vergabe eines Funktionstextes für einen Funktionscode

Icon hinzufügen

Die Angabe des Funktionstextes ist die Minimalanforderung, um einen Funktionscode zu aktivieren. Sie können aber zusätzlich noch ein Standard-Icon im Feld **Icon** hinterlegen sowie einen Text, der zu diesem Icon als



Tooltip angezeigt werden soll (Feld **Icontext**). Nutzen Sie hier zur Auswahl die **[F4]**-Hilfe, in der Ihnen alle verfügbaren Icons angezeigt werden. Die so hinterlegten Icons werden allerdings nur in der Funktionsleiste angezeigt und nicht im Hauptmenü.

8. Aktivieren Sie das gesamte Erweiterungsprojekt, indem Sie auf das Icon **Aktivieren** (🔍) in der Funktionsleiste klicken (siehe Abbildung 4.34).

Menüerweiterung
testen

Nach dem Aktivieren können Sie die Transaktion VCO2 aufrufen, einen Kontrakt auswählen und schauen, ob Ihre neue Funktion dem Hauptmenü **Informationen** hinzugefügt wurde.

Programmlogik
implementieren

Nun müssen Sie noch definieren, wie die Transaktion bei einem Klick auf diese neue Funktion reagieren soll. Dazu dient der Funktions-Exit `EXIT_SAPMV43A_004` innerhalb der Erweiterung. Auch diesen sehen Sie in der Komponentenliste in Abbildung 4.34. Anhand der Beschreibung des Funktionsbausteins (»GUI-Exit '+5X1': Kontakte/Aktionen«) in Abbildung 4.34 wird deutlich, dass dieser Funktionsbaustein ausschließlich für den Funktionscode +5X1 aufgerufen wird. Das muss auch so sein, da kein Importparameter den ursprünglich ausgelösten Funktionscode mitgibt (und das Systemfeld `sy-ucomm` leer ist). Dies kann bei anderen Erweiterungen allerdings unterschiedlich sein. So kann es für mehrere Funktionscodes einen Funktionsbaustein geben, in dem der aktuell ausgelöste Funktionscode als Importparameter mit übergeben wird.

Für diese Erweiterung reicht es aus, das `Include ZXV43U04` in dem Funktionsbaustein `EXIT_SAPMV43A_004` wie bei einer normalen Erweiterungsimpementierung anzulegen (siehe Abschnitt 4.1, »Customer Exits implementieren«) und mithilfe der mitgegebenen Importparameter zu implementieren.

Funktionsaufruf
testen

Um den Aufruf der Funktion zu testen, können Sie einen Breakpoint im `Include` hinzufügen. So sehen Sie, welche Daten Sie innerhalb des Funktions-Exits zur Verfügung haben. Auf diese Art und Weise können Sie auch überprüfen, ob die Vermutung stimmt, dass dieser Funktions-Exit diesen Funktionscode behandelt. Tut der Funktions-Exit dies nicht, wird er beim Auswählen des neuen Menüeintrags in Transaktion VCO1 auch nicht aufgerufen.

Abbildung 4.36 zeigt diesen Funktions-Exit im Debugger. So wird z. B. im Funktions-Exit über den Importparameter `FAKTYP` der Bearbeitungsmodus der Transaktion mitgegeben (mit dem Wert »V« für die Aktion **Bearbeiten/Anlegen** und dem Wert »A« für eine **Anzeige**). Diese Parameter können Sie nun nutzen, um die Logik Ihrer Menüerweiterung nach Belieben zu programmieren.

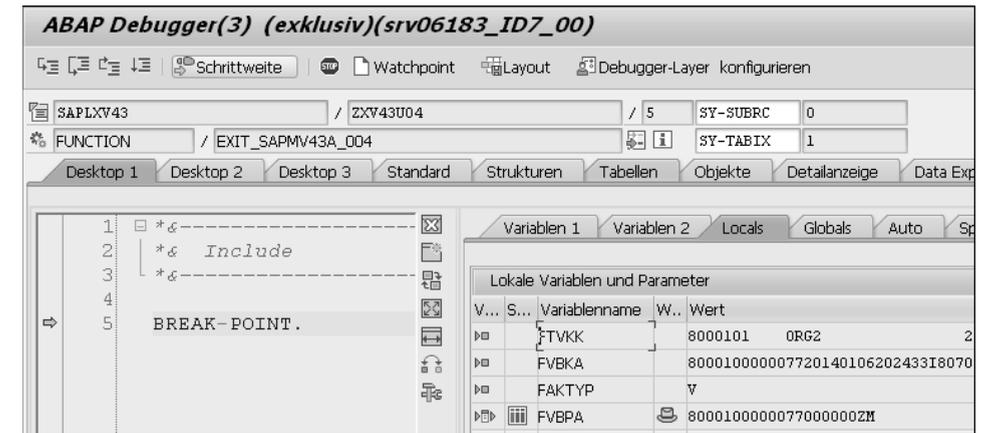


Abbildung 4.36 Der Funktions-Exit für die Menüerweiterung im Debugger

Damit haben Sie erfolgreich eine Menüerweiterung über einen Customer Exit implementiert.

4.3.3 Menüerweiterungen finden

Neben dem klassischen Weg herauszufinden, ob eine Transaktion eine Menüerweiterung hat, den ich in Abschnitt 4.2, »Funktions-Exits finden«, beschrieben habe (Suche über die **[F4]**-Hilfe in Transaktion SMOD), ist folgendes der einfachste Weg:

Tabelle MODSAP

1. Öffnen Sie die Tabelle MODSAP mit Transaktion SE11 oder SE16N.
2. Geben Sie als Typ »C« an.
3. Tragen Sie den Programmnamen in das Feld **MEMBER** (Erweiterung) ein, gefolgt von einer Wildcard (*). Abbildung 4.37 zeigt die Eingaben anhand des Beispiels aus Abschnitt 4.3.2, »Menüerweiterungen implementieren«, also für das Programm SAPMV43A.

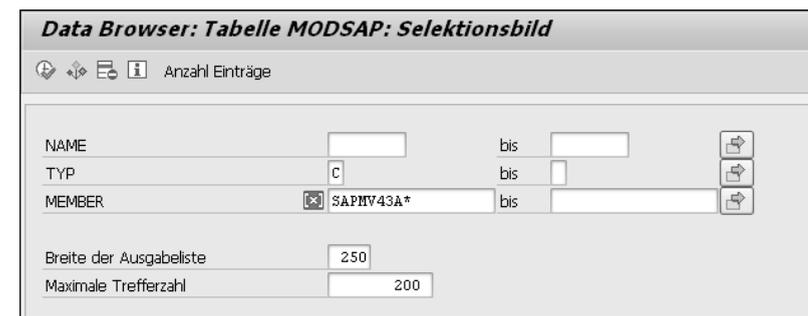
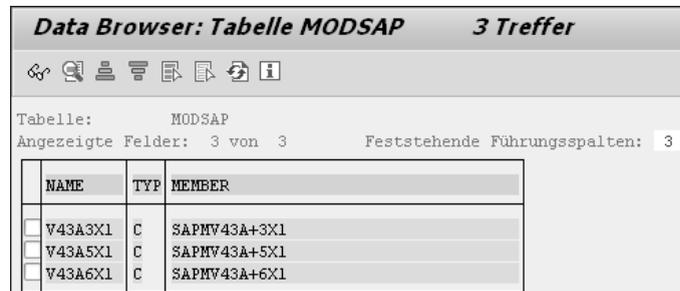


Abbildung 4.37 Die Tabelle MODSAP zur Suche von Menüerweiterungen

4. Führen Sie die Suche über das entsprechende Symbol in der Funktionsleiste  aus.
5. Ihnen wird das Ergebnis wie in Abbildung 4.38 angezeigt.



Data Browser: Tabelle MODSAP 3 Treffer		
NAME	TYP	MEMBER
V43A3X1	C	SAPMV43A+3X1
V43A5X1	C	SAPMV43A+5X1
V43A6X1	C	SAPMV43A+6X1

Abbildung 4.38 Ergebnisliste zur Suche nach Menüerweiterungen in der Tabelle MODSAP

Insgesamt drei Funktionscodes sind innerhalb des Programms SAPMV43A definiert. Die erste Spalte (**NAME**) zeigt den Namen der Erweiterung an, die Sie nun in der Transaktion SMOD analysieren und, falls gewünscht, in den in Abschnitt 4.3.2, »Menüerweiterungen implementieren«, beschriebenen Schritten implementieren können.

4.4 Screenerweiterungen

Erweiterung der Oberfläche

Bei Screenerweiterungen geht es um die Erweiterung der Oberfläche von Standardtransaktionen. Hierbei dreht es sich immer um einen vordefinierten *Subscreenbereich* auf einem Dynpro innerhalb dieser Standardtransaktionen. In einem solchen Subscreenbereich kann zur Laufzeit ein von Ihnen frei definiertes Dynpro geladen werden. Bei Customer Exits sind diese Dynpros bereits mit einer Nummer vordefiniert und können von Ihnen angelegt werden.

Erweiterungen eines Dynpro-Bereichs

Ob ein Customer Exit solche Dynpro-Bereiche enthält, können Sie herausfinden, indem Sie sich die Komponentenansicht für die Erweiterung in der Transaktion SMOD anzeigen lassen. Abbildung 4.39 zeigt dies für die Erweiterung MM06E005. Mit dieser Erweiterung können Sie neue Registerkarten in den Transaktionen ME21N, ME22N und ME23N (Kundendaten) anlegen.

Im Bereich **Dynprobereiche** finden Sie die hinterlegten Screenerweiterungen. In dieser Erweiterung werden beispielsweise das Dynpro 0101 für den Kopf der Bestellung und das Dynpro 0111 für die Position einer Bestellung angeboten. Daneben gibt es weitere Screens, deren Liste Sie mit den Schaltflächen **Auf**  und **Ab**  durchblättern können.



Komponenten in SAP-Erweiterung MM06E005					
Funktionsexits					
Funktionsbaustein			Kurztext		
EXIT_SAPMM06E_006			Export Daten zum Kundensubscreen für Einkaufsb...		
EXIT_SAPMM06E_007			Export Daten zum Kundensubscreen für Einkaufsb...		
EXIT_SAPMM06E_008			Import Daten vom Kundensubscreen für Einkaufsb...		
Funktionscodes					
Programm	Code	Kurztext			
Dynprobereiche					
Rufendes Dynpro	Nr.	Bereich	Gerufenes Dynp.	Nr.	Kurztext
SAPMM06E	0101	CUSTSCR1	SAPLXM06	0101	Subscreen Kopf Bestellu...
SAPMM06E	0111	CUSTSCR1	SAPLXM06	0111	Subscreen Position Best...
SAPMM06E	0201	CUSTSCR1	SAPLXM06	0201	Subscreen Kopf Vertrag
Includes					
CI_EKKODB					
CI_EKPODB					

Abbildung 4.39 Komponenten der Erweiterung MM06E005

Für jeden Subscreenbereich gibt es außerdem Funktions-Exits, die sich darum kümmern, die Daten zu dem jeweiligen Bereich zu transportieren und letztlich auch wieder auszulesen. Die Zuordnung dieser Funktions-Exits ist in Transaktion SMOD nur über den Kurztext ersichtlich. In der Erweiterung MM06E005 gehören beispielsweise die folgenden Funktions-Exits zu dem Dynpro 0101 für den Kopf der Bestellung:

Funktions-Exits

- EXIT_SAPMM06E_006
- EXIT_SAPMM06E_007
- EXIT_SAPMM06E_008

Zu dem Dynpro 0111 für die Positionsebene bei Bestellungen gehören dagegen die folgenden Funktions-Exits:

- EXIT_SAPMM06E_016
- EXIT_SAPMM06E_017
- EXIT_SAPMM06E_018

Append-Strukturen Zusätzlich stellt der Customer Exit zwei bereits vorbereitete Append-Strukturen bereit:

- CI_EKKODB zur Erweiterung der Tabelle EKKO für den Bestellkopf
- CI_EKPODB zur Erweiterung der Tabelle EKPO für die Bestellpositionen

Den Umgang mit Append-Strukturen habe ich Ihnen in Abschnitt 2.1, »Strukturerweiterungen«, gezeigt. In der Tat ist dies meiner Ansicht nach eines der besten Features von Screenerweiterungen: Es werden die Standardstrukturen der Datenbanktabellen verwendet, die automatisch von der SAP-Verarbeitungslogik der betreffenden Transaktionen gelesen und geändert werden. Dieselben Strukturen stehen Ihnen innerhalb der Funktions-Exits im Customer Exit als Parameter zur Verfügung. Wenn Sie also ein Feld auf der Datenbank erweitern, bedeutet dies, dass dieses Feld automatisch bis in den Customer Exit durchgereicht wird. Befüllen Sie die zusätzlichen Felder im Customer Exit mit einem Wert, kümmert sich die Verarbeitungslogik der Transaktion automatisch darum, diesen Wert auch in der Datenbank zu speichern. Sie müssen sich also nicht extra um das Speichern der zusätzlichen Felder kümmern. Im folgenden Abschnitt zeige ich Ihnen dies an einem Beispiel.

4.4.1 Screenerweiterungen implementieren

In diesem Abschnitt zeige ich Ihnen die Implementierung eines Subscreenbereichs an einem Beispiel. Abbildung 4.40 zeigt den fertig implementierten Customer Exit MM06E005, mit dem eine neue Registerkarte **Kundendaten** auf Kopfebene einer Bestellung in der Transaktion ME22N (Ändern einer Bestellung) angelegt wurde. Jede Eingabe, die ein Anwender auf dieser Registerkarte im Feld **Zusatzfeld** tätigt, wird beim Speichern automatisch auch in der Datenbanktabelle EKKO gespeichert, die die Köpfe aller Bestellungen enthält.



Abbildung 4.40 Hinzugefügte Registerkarte und Zusatzfeld in Transaktion ME21N

Bevor ich Ihnen Schritt für Schritt zeige, wie Sie eine solche Screenerweiterung für einen Customer Exit implementieren können, gebe ich Ihnen einen Überblick über die benötigten Schritte:

Implementierungsschritte

1. Erweiterungsprojekt in der Transaktion CMOD anlegen und die Erweiterung dem Erweiterungsprojekt zuordnen
2. Datenbanktabelle EKKO über das in der Erweiterung angegebene Include CI_EKKODB erweitern
3. Dynpro 0101 für den Kopf anlegen und als Subscreen definieren
4. Felder im Layout des Screens mit Verweis auf die Erweiterungsstruktur CI_EKKODB hinzufügen
5. Funktions-Exit EXIT_SAPMM06E_006 zum Transport der Daten auf den Subscreen anlegen und implementieren
6. Funktions-Exit EXIT_SAPMM06E_008 zum Auslesen der Daten vom Subscreen anlegen und implementieren
7. Steuerung der Eingabefähigkeit des Zusatzfeldes anhand des Bearbeitungsmodus (Anzeige oder Ändern) implementieren

Zunächst legen Sie das Erweiterungsprojekt an und ordnen diesem die Erweiterung zu:

Erweiterungsprojekt anlegen

1. Öffnen Sie dazu Transaktion SMOD, geben Sie einen Namen für Ihr neues Erweiterungsprojekt an, z. B. »ZMM_ME«, und klicken Sie auf die Schaltfläche **Anlegen** (siehe Abbildung 4.41).

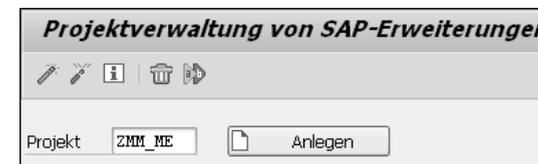


Abbildung 4.41 Anlage eines neuen Erweiterungsprojekts

2. Geben Sie auf der nächsten Oberfläche einen Kurztext für das neue Erweiterungsprojekt ein, und wechseln Sie über die Schaltfläche **Zuordnung Erweiterung** zu der Oberfläche, die Sie in Abbildung 4.42 sehen.

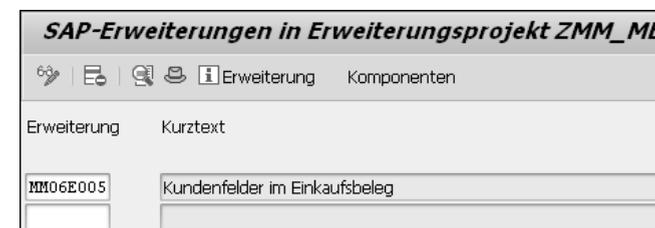


Abbildung 4.42 Zuordnen der Erweiterung zum Erweiterungsprojekt

- Tragen Sie die Erweiterung »MMO6E005« in die Liste ein, und bestätigen Sie Ihre Eingabe mit . Dadurch werden alle Komponenten dieser Erweiterung in Ihr Erweiterungsprojekt übernommen.

Komponenten- übersicht

Diese Komponenten können Sie sehen, wenn Sie über die Schaltfläche **Komponenten** zu der Komponentenansicht in Abbildung 4.43 wechseln. Sie finden hier dieselben Komponenten wie in Transaktion SMOD (siehe Abbildung 4.39), mit dem Unterschied, dass Sie sie über diese Übersicht direkt implementieren können.

ZMM_ME ändern					
Zuordnung Erweiterung  Erweiterung					
Projekt				ZMM_ME Erweiterung Bestelltransaktion	
Erweiterung	Impl	△	Bsp	MMO6E005 Kundenfelder im Einkaufsbeleg	
Funktionsexit				EXIT_SAPMMO6E_006 EXIT_SAPMMO6E_007 EXIT_SAPMMO6E_008 EXIT_SAPMMO6E_009 EXIT_SAPMMO6E_012 EXIT_SAPMMO6E_013 EXIT_SAPMMO6E_014 EXIT_SAPMMO6E_016 EXIT_SAPMMO6E_017 EXIT_SAPMMO6E_018	
Screenexit				SAPMMO6E 0101 CUSTSCR1 SAPLXM06 0101 SAPMMO6E 0111 CUSTSCR1 SAPLXM06 0111 SAPMMO6E 0201 CUSTSCR1 SAPLXM06 0201 SAPMMO6E 0211 CUSTSCR1 SAPLXM06 0211 SAPMMO6E 0301 CUSTSCR1 SAPLXM06 0301 SAPMMO6E 0311 CUSTSCR1 SAPLXM06 0311	
Includetabellen				CI_EKKODB CI_EKPODB	

Abbildung 4.43 Die implementierbaren Komponenten des Erweiterungsprojekts

Tabelle EKKO erweitern

Im ersten Schritt erweitern wir die Datenbanktabelle EKKO (Kopfdaten zur Bestellung) um ein neues Feld. Dieses soll auf der neuen Registerkarte **Kundendaten** zur Eingabe angeboten werden.

- Für diese Erweiterung bietet der Customer Exit im Bereich **Includetabellen** der Komponentenübersicht das Include CI_EKKODB an. Klicken Sie doppelt auf den Namen dieses Includes, und bestätigen Sie die Frage, ob dieses angelegt werden soll, mit **Ja**.
- Dadurch gelangen Sie zu der Oberfläche zur Anpassung einer ABAP-Dictionary-Struktur, die Sie in Abbildung 4.44 sehen. Fügen Sie hier ein neues Feld »ZZUSATZ« vom Typ CHAR10 hinzu.

- Aktivieren** Sie diese Struktur anschließend über das Icon in der Funktionsleiste . Das Aktivieren kann mitunter einige Minuten dauern, da die Tabelle EKKO ebenfalls neu aktiviert werden muss.

Dictionary: Struktur ändern						
Hierarchiedarstellung Append-Struktur...						
Struktur	CI_EKKODB	neu(überarbeitet)				
Kurzbeschreibung	Zusatzfelder für Bestellkopf					
Eigenschaften Komponenten Eingabehilfe/-prüfung Währungs-/Mengenfelder						
Eingebauter Typ 1 / 1						
Komponente	Typisierungsart	Komponententyp	Datentyp	Länge	DezS...	Kurzbeschreibung
ZZUSATZ	1 Type	CHAR10	CHAR	10	0	Characterfeld der Länge 10

Abbildung 4.44 Erweitern der Struktur CI_EKKODB

Z-Include für komplexe Erweiterungen

Für komplexere Erweiterungen bietet es sich an, innerhalb einer CI-Struktur (hier CI_EKKODB) ein weiteres Z-Include anzulegen, in dem Sie Ihre Zusatzfelder definieren. Dies hat den Vorteil, dass Sie so das ursprüngliche Z-Include an mehreren Stellen (z. B. für das Erweitern von BAPI-Funktionsbausteinen und anderen Strukturen) verwenden können und die Felder nicht immer wieder einzeln neu hinzufügen müssen. Wird später beispielsweise die Struktur CI_EKKODB um ein neues Feld erweitert, müssen Sie selbst daran denken, alle anderen Implementierungsstellen um dasselbe Zusatzfeld zu erweitern.

Wie Abbildung 4.45 zeigt, hat SAP die Struktur CI_EKKODB als Include in der Datenbanktabelle EKKO hinzugefügt. Durch das Aktivieren der Struktur haben Sie also die Standardtabelle EKKO um ein neues Feld erweitert.

Dictionary: Tabelle anzeigen						
Technische Einstellungen Indizes... Append...						
Transp. Tabelle	EKKO	aktiv				
Kurzbeschreibung	Einkaufsbelegkopf					
Eigenschaften Auslieferung und Pflege Felder Eingabehilfe/-prüfung Währungs-/Mengenfelder						
Suchhilfe Eingebauter Typ						
Feld	Key	Ini...	Datenelement	Datentyp	Länge	DezS... Kurzbeschreibung
RELOC_ID	<input type="checkbox"/>	<input type="checkbox"/>	/ISDFPS/RELOC_ID	CHAR	10	0ID der Verlegung
RELOC_SEQ_ID	<input type="checkbox"/>	<input type="checkbox"/>	/ISDFPS/RELOC_S...	CHAR	4	0Verlegeschritt-ID
SOURCE LOGSYS	<input type="checkbox"/>	<input type="checkbox"/>	LOGSYS	CHAR	10	0Logisches System
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	CI_EKKODB	STRU	0	0
.APPEND	<input type="checkbox"/>	<input type="checkbox"/>	FPIA_A_INT	STRU	0	0
.INCLUDE	<input type="checkbox"/>	<input type="checkbox"/>	FPIA_S_INTEREST	STRU	0	0

Abbildung 4.45 Die erweiterte Datenbanktabelle EKKO

Dynpro anlegen Im nächsten Schritt geht es nun darum, das Dynpro anzulegen, das als neue Registerkarte in der Bestelltransaktion angezeigt werden soll und das eben angelegte Feld zur Eingabe anbieten soll:

1. Wechseln Sie dazu zurück in die Komponentenanzeige der Erweiterung. Wie in Abbildung 4.46 zu erkennen, sind im Bereich **Screenexit** mehrere Dynpros angegeben. Das Dynpro für den Kopf ist das Dynpro 0101. Dies können Sie nur der Kurzbeschreibung in Transaktion SMOD entnehmen (siehe aus Abbildung 4.39). Diese Kurzbeschreibung wird in der Komponentenübersicht nicht mehr angezeigt.

Screenexit						
	SAPMM06E	0101	CUSTSCR1	SAPLXM06	0101	
	SAPMM06E	0111	CUSTSCR1	SAPLXM06	0111	
	SAPMM06E	0201	CUSTSCR1	SAPLXM06	0201	
	SAPMM06E	0211	CUSTSCR1	SAPLXM06	0211	
	SAPMM06E	0301	CUSTSCR1	SAPLXM06	0301	
	SAPMM06E	0311	CUSTSCR1	SAPLXM06	0311	

Abbildung 4.46 Die definierten Screen-Exits in der Komponentenliste

2. Klicken Sie hier doppelt auf die Zeile mit dem Dynpro 0101, um es anzulegen.
3. Sie erhalten die Meldung, dass das Dynpro 0111 zum Programm SAPLXM06 angelegt wird. Bestätigen Sie diese Meldung mit .
4. Sie gelangen dadurch zur Pflege des Dynpros in den Screen Painter (siehe Abbildung 4.47). Geben Sie hier eine **Kurzbeschreibung** für das neue Dynpro ein, und wählen Sie **Subscreen** als **Dynprotyp** aus.

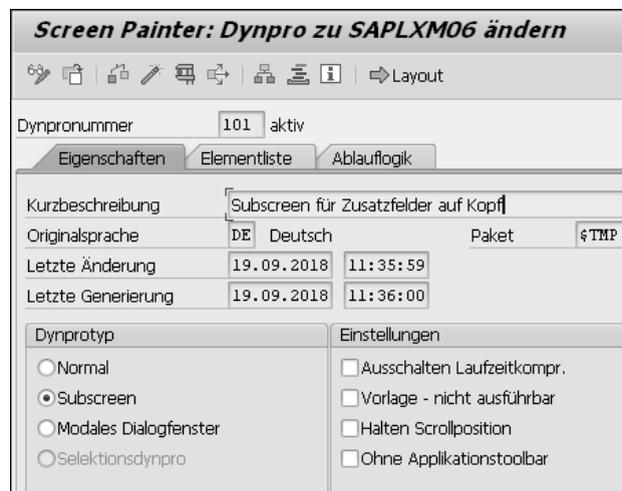


Abbildung 4.47 Anlage des Subscreens für die Screenerweiterung

Dynpro-Typ »Subscreen«

Wenn Sie vergessen, den Dynpro-Typ **Subscreen** auszuwählen, werden Sie beim Aufruf der neuen Registerkarte in der Bestelltransaktion einen Laufzeitfehler mit der Meldung erhalten, dass das Dynpro falsch definiert ist oder bereits verwendet wird.

5. Wechseln Sie anschließend über die Schaltfläche **Layout** in den Layout-Editor.
6. Legen Sie dort ein Eingabefeld mit dem Namen »EKKO_CI-ZZUSATZ« an, wie in Abbildung 4.48 gezeigt.
7. Ergänzen Sie dazu noch ein Label mit dem Text »Zusatzfeld«, um das Eingabefeld zu beschriften. Hier bietet es sich an, die Eigenschaft **Groß- und Kleinschreibung** zu aktivieren, damit die Eingabe des Anwenders nicht immer automatisch in Großbuchstaben umgewandelt wird.

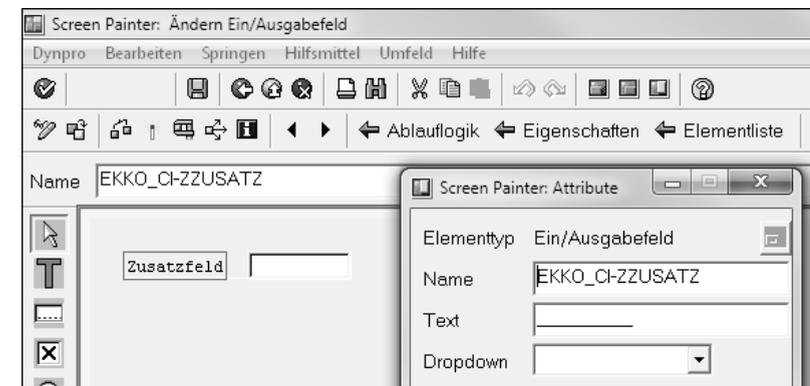


Abbildung 4.48 Hinzufügen des Eingabefeldes auf dem Subscreen

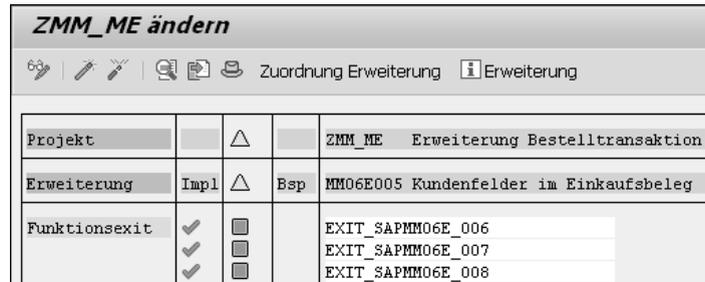
Bezug auf die Include-Struktur EKKO_CI

Achten Sie darauf, dass Ihr Dynpro-Feld sich über den Namen auf die Include-Struktur EKKO_CI, und nicht auf die Datenbanktabelle EKKO bezieht. Andernfalls kann später die Bestelltransaktion den eingetragenen Wert nicht verarbeiten, da das SAP-System einen Bezug auf die Struktur EKKO_CI erwartet. Die korrekte Benennung des Eingabefeldes ist daher sehr wichtig.

8. **Aktivieren**  Sie das neu angelegte Dynpro, und wechseln Sie anschließend zurück zur Komponentenansicht aus Abbildung 4.49.

9. Hier können Sie testen, ob die neue Registerkarte **Kundendaten** in der Bestelltransaktion bereits angezeigt wird. Aktivieren Sie hierzu die Erweiterung, indem Sie in der Funktionsleiste der Komponentenansicht auf die Schaltfläche **Aktivieren**  klicken.

Vor den einzelnen Erweiterungskomponenten wird nun jeweils ein grünes Ampelsymbol angezeigt (siehe Abbildung 4.49). Ihr Erweiterungsprojekt wurde damit erfolgreich aktiviert.



ZMM_ME ändern			
63 Zuordnung Erweiterung Erweiterung			
Projekt			ZMM_ME Erweiterung Bestelltransaktion
Erweiterung	Impl	Bsp	MM06E005 Kundenfelder im Einkaufsbeleg
Funktionsexit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EXIT_SAPMM06E_006
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EXIT_SAPMM06E_007
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EXIT_SAPMM06E_008

Abbildung 4.49 Die aktivierten Komponenten des Customer Exits

Screenenerweiterung testen

Wenn Sie jetzt in die Transaktion ME22N wechseln, werden Sie auf der Kopfebene bereits die Registerkarte **Kundendaten** sehen können (siehe Abbildung 4.50). Auch wenn noch keine Bestellung ausgewählt ist (oder Sie sich im Anzeigemodus befinden), kann ein Wert in das neue Eingabefeld **Zusatzfeld** eingetragen werden. Dies müssen wir in einem späteren Schritt noch unterbinden.



Abbildung 4.50 Hinzugefügte Registerkarte und Zusatzfeld in Transaktion ME21N

Datenverarbeitung implementieren

Im nächsten Schritt geht es jedoch erst einmal darum, den auf der Oberfläche eingegebenen Wert an die Datenbanktabelle **EKKO** in das Zusatzfeld **ZZUSATZ** zu übertragen und auch wieder auslesen zu können. Zur Realisierung dieser Anforderung stellt der Customer Exit die folgenden beiden Funktions-Exits bereit:

- EXIT_SAPMM06E_006 für den Export der Daten zum Kundensubscreen für den Einkaufsbelegkopf (PBO = Process Before Output, also vor Anzeige des Kundensubscreens)
- EXIT_SAPMM06E_008 für den Import der Daten aus dem Kundensubscreen für den Einkaufsbelegkopf (PAI, also nach der Anzeige des Kundensubscreens)

Gehen Sie wie folgt vor, um diese Datenverarbeitung zu implementieren:

1. Wechseln Sie wieder in die Komponentenansicht des Customer Exits aus Abbildung 4.49, und klicken Sie doppelt auf den Funktions-Exit EXIT_SAPMM06E_006.
2. Sie gelangen zur Anzeige des Funktionsbausteins im ABAP Editor (Transaktion SE37, siehe Abbildung 4.51).

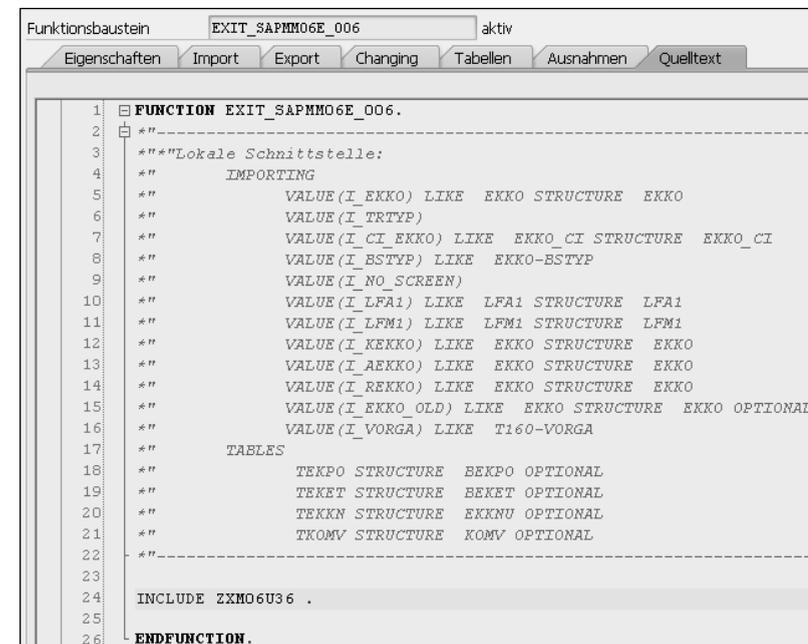


Abbildung 4.51 Der Funktions-Exit EXIT_SAPMM06E_006 zum Export der Daten auf die Screenenerweiterung des Belegkopfes

3. Klicken Sie hier doppelt auf den Include-Namen **ZXM06U36** (in Abbildung 4.51 in Zeile 24).
4. Bestätigen Sie die folgende Warnung (je nach Systemeinstellung nicht in einem Pop-up-Fenster) mit :

»Programmnamen ZX... sind für Includes v. Exit-Funktionsgruppen reserviert«

5. Programmieren Sie das Include ZXM06U36 mit dem Quellcode aus Listing 4.2. Die zweite Zeile sorgt dafür, dass das Zusatzfeld aus der Datenbanktabelle EKKO in die Erweiterung übertragen wird.

```
gv_trtyp      = i_trtyp. "Übernahme des Bearbeitungsmodus
ekko_ci-zzusatz = i_ekko-zzusatz.
```

Listing 4.2 Implementierung des Exports der Daten zum Kundensubscreen

Aktivieren können Sie dieses Include erst später, da das Feld GV_TRTYP noch unbekannt ist.

6. Wechseln Sie zurück in die Komponentenansicht, und wiederholen Sie die vorangegangenen Schritte für den Funktions-Exit EXIT_SAPMM06E_008, in dem Sie das Include ZXM06U37 anlegen. Programmieren Sie dieses Include mit dem Quellcode aus Listing 4.3. Hier sorgt die erste Zeile dafür, dass der vom Benutzer eingegebene Wert an die Standardimplementierung übergeben wird.

```
e_ci_ekko = ekko_ci.
e_ci_update = 'X'.
"Wichtig, da ansonsten der eingegebene Wert nicht
"gespeichert wird!
```

Listing 4.3 Implementierung des Imports der Daten vom Kundensubscreen

7. Nun gilt es noch, das Top-Include anzulegen, in dem das Feld GV_TRTYP definiert wird. Dieses Feld wird später den Bearbeitungsmodus der Transaktion enthalten (den Wert »A« für die Anzeige und den Wert »V« für das Ändern oder die Anlage einer Bestellung). Rufen Sie hierzu in Transaktion SE80 die Funktionsgruppe XM06 auf, wie in Abbildung 4.52 gezeigt.

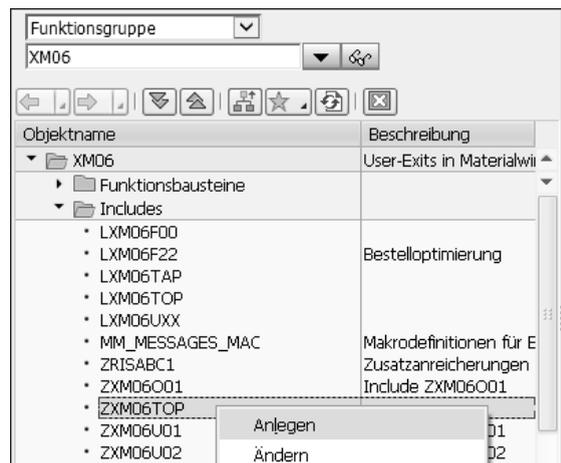


Abbildung 4.52 Anlegen des Top-Includes für die Screenerweiterung

8. Suchen Sie dort im Ordner **Includes** das Include ZXM06TOP, klicken Sie es mit der rechten Maustaste an, und legen Sie es über das Kontextmenü an.
9. Fügen Sie in dem so angelegten Include die folgende Zeile ein:

```
DATA : gv_trtyp(1) TYPE c.
```

10. **Aktivieren** (☑) Sie alle angelegten Includes anschließend. Achten Sie darauf, dass Sie wirklich alle inaktiven Entwicklungsobjekte aktivieren.

Im letzten Arbeitsschritt geht es nur noch darum, dass das Eingabefeld über den Bearbeitungsmodus gesteuert wird:

Steuerung des Eingabefeldes

- Hierzu müssen Sie über die Komponentenliste aus Abbildung 4.46 zurück in das vorhin angelegte Dynpro 0101 wechseln.
- Wechseln Sie innerhalb des Dynpros auf die Registerkarte **Ablauflogik**, die Sie in Abbildung 4.53 sehen.
- Entfernen Sie hier das Kommentarzeichen in Zeile 2 (MODULE STATUS_0101) für das PBO-Modul.
- Klicken Sie anschließend doppelt auf diese Zeile, und lassen Sie das Modul im angegebenen Include ZXM06001 anlegen.

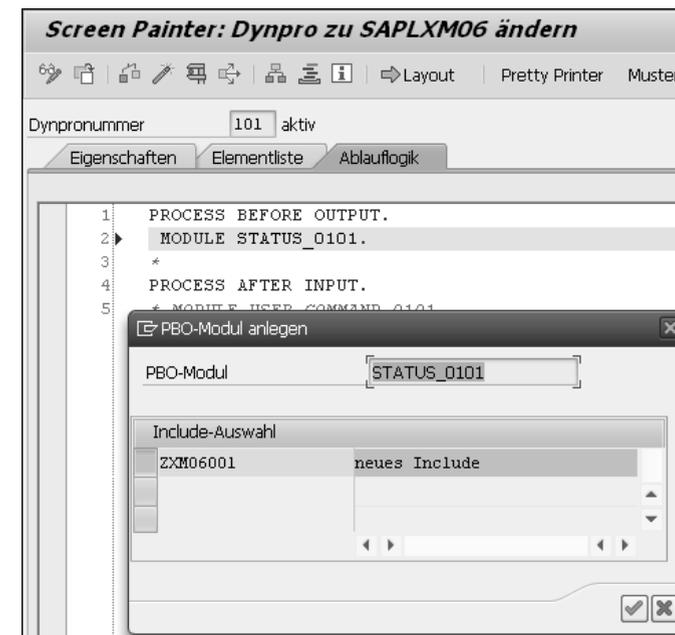


Abbildung 4.53 Anlage des PBO-Moduls für das Dynpro 101

5. Prägen Sie das Modul mit dem Quellcode aus Listing 4.4 aus. Wenn das Feld GV_TRTYP den Wert »A« enthält, befindet sich die Bestelltransaktion

im Anzeigemodus. Deshalb müssen die hinzugefügten Zusatzfelder über `screen-input = 0` auf den Status »nicht eingabebereit« gesetzt werden.

```
MODULE status_0101 OUTPUT.
  IF gv_trtyp = 'A'.
    LOOP AT SCREEN.
      screen-input = 0.
      MODIFY SCREEN.
    ENDLOOP.
  ELSE.
    LOOP AT SCREEN.
      screen-input = 1.
      MODIFY SCREEN.
    ENDLOOP.
  ENDIF.
ENDMODULE.
```

Listing 4.4 Steuerung der Eingabefähigkeit des Zusatzfeldes abhängig vom Bearbeitungsmodus

6. **Aktivieren**  Sie die Änderungen anschließend.

Erweiterung testen

Damit ist die Implementierung des Customer Exits zur Erweiterung des Bestellkopfes erfolgreich abgeschlossen. Wenn Sie jetzt erneut in die Transaktion ME22N wechseln, eine Bestellung auswählen und diese ändern, können Sie in dem neuen Eingabefeld einen Wert »Zusatzwert« eintragen und abschließend speichern.

Wenn Sie die Bestellung erneut aufrufen oder über die Transaktion SE11 oder SE16N in Tabelle EKKO nachschauen, werden Sie feststellen, dass der eingegebene Wert wie in Abbildung 4.54 erfolgreich gespeichert wurde.

Tabelle EKKO anzeigen	
Prüftabelle...	
RELEASE DATE	
VSART	
HANDOVERLOC	
SHIPCOND	
FORCE ID	
FORCE CNT	0
RELOC ID	
RELOC SEQ ID	
SOURCE LOGSYS	
ZZUSATZ	Zusatzwert
VZSKZ	

Abbildung 4.54 Eingetragener Wert für das Zusatzfeld ZZUSATZ in der Tabelle EKKO

Damit haben Sie erfolgreich die Bestelltransaktion mit einem Customer Exit um eine neue Registerkarte mit einem neuen Feld erweitert. Dieses Feld wird verarbeitet, und die Daten werden in der Bestellkopftabelle EKKO gespeichert. Beim Anzeigen einer Bestellung werden sie auch wieder ausgelesen und angezeigt.

4.4.2 Screenerweiterungen finden

Neben der in Abschnitt 4.2, »Funktions-Exits finden«, beschriebenen Suche nach einem Customer Exit über die **F4**-Hilfe der Transaktion SMOD ist der im Folgenden beschriebene Weg der einfachste, um herauszufinden, ob eine Transaktion Screenerweiterungen vorsieht:

Suche in Tabelle MODSAP

1. Öffnen Sie die Tabelle MODSAP in Transaktion SE11 oder SE16N. Geben Sie als Typ »S« (Screens) an.
2. Tragen Sie zusätzlich den Namen des hinter der zu erweiternden Transaktion liegenden Programms in das Feld **MEMBER** ein, gefolgt von einer Wildcard (*). Abbildung 4.55 zeigt die Eingabe anhand des Beispiels aus Abschnitt 4.4.1, »Screenerweiterungen implementieren«, für das Programm SAPMM06E.
3. Führen Sie anschließend die Suche aus .

Abbildung 4.55 Screenerweiterungen in der Tabelle MODSAP suchen

4. Das Ergebnis wird Ihnen wie in Abbildung 4.56 ausgegeben. In unserem Beispiel sind insgesamt sechs Screenerweiterungen innerhalb des Programms definiert. Die erste Spalte (**NAME**) zeigt den Namen der Erweiterung an, die Sie nun in Transaktion SMOD analysieren können, um herauszufinden, wo genau der Subscreen eingebettet worden ist.

