

Der Weg zum Programm

ABAP® – Die Programmiersprache

des SAP®-Systems R/3®

ISBN 3-8273-1960-9

3

Eine sehr wichtige, aber nicht die einzige Voraussetzung zur erfolgreichen Programmierung eigener Anwendungen innerhalb des R/3-Systems ist die Kenntnis der Programmiersprache ABAP. Diese Sprache ist trotz aller Modifikationen der letzten Jahre keine universell einsetzbare Sprache wie z.B. C oder Pascal, auch wenn sie sich in der Praxis als sehr flexibel erwiesen hat. Vielmehr wurde sie speziell für das SAP-System entworfen. Die Sprache wird ständig weiterentwickelt und aktuellen Erfordernissen angepasst. Von einem Releasestand zum anderen ergeben sich zum Teil beträchtliche Ergänzungen. Einige Sprachelemente sind völlig auf ausgewählte Elemente der Entwicklungsumgebung bzw. auf die SAP-Philosophie zugeschnitten und können nur im Zusammenhang mit diesen Elementen beschrieben werden. Es ist deshalb wenig sinnvoll, die einzelnen Kommandos der Programmiersprache losgelöst vom übrigen SAP-Umfeld zu betrachten. Viel wichtiger als die detaillierte Beschreibung aller Kommandos mit zum Teil Dutzenden Parametern und Aufrufoptionen ist zunächst die Vermittlung von Grundwissen zu den verschiedenen Bestandteilen einer R/3-Anwendung. Dieses Kapitel soll daher keine vollständige Beschreibung der ABAP-Kommandos sein. Vielmehr erläutert es eine Auswahl der wichtigsten Anweisungen und demonstriert an sehr einfachen Beispielen deren praktischen Einsatz, das Zusammenwirken mit anderen Elementen der Entwicklungsumgebung und damit auch die Programmstruktur einer Anwendung. Dieser Abschnitt ist daher nicht als vollständige Referenz der Programmiersprache, sondern als schrittweise praktische Einführung in die R/3-Programmierung zu verstehen. Dieses Kapitel eignet sich daher weniger zum Nachschlagen bestimmter Fakten, sondern sollte sequenziell durchgearbeitet werden.

Bereits in den zurückliegenden Versionen, vor allem seit der Version 4.6, versucht die SAP die Sprache ABAP und das Konzept des Gesamtsystems intensiv auf die Erfordernisse des Internets auszurichten. Dafür wurden einige neue Programmierkonzepte eingeführt, die mit bisherigen Traditionen brechen. Trotz aller neuen Elemente ist die Kenntnis der seit Jahren existierenden Program-

miermodelle aber keineswegs überflüssig. Zum einen können nicht alle Anwendungen von einer Version zur anderen komplett umgestellt werden. Pflege und Anpassung existierender Anwendungen stellt aber einen großen Anteil der praktischen ABAP-Programmierung dar. Zum anderen erleichtert die Kenntnis der Wurzeln des R/3-Systems das Verständnis für die neuen Elemente. Darüber hinaus bringt ein Internet-basierter Zugriff auf das R/3-System nicht nur Vorteile mit sich. Die auf dem HTML-Standard beruhenden Objekte der Benutzerschnittstelle verfügen nicht über die Komplexität und Interaktivität der Elemente, die in einer reinen Desktop-Anwendung zur Verfügung stehen.

Am genannten Anliegen orientiert sich auch der Aufbau dieses Kapitels. Voraussetzung zur Erstellung eigener Programme ist die Kenntnis der Entwicklungsumgebung und der diversen Werkzeuge. Im ersten Unterabschnitt lernen Sie die grundlegenden Eigenschaften der Entwicklungsumgebung kennen, in dem Sie ein denkbar einfaches Programm erzeugen. Der darauf folgende Abschnitt beschreibt grundlegende Kommandos, die in allen Programmtypen eingesetzt werden können. Nahezu alle dieser Anweisungen sind von globaler Bedeutung. Die Kenntnis dieser Anweisungen ist auf jeden Fall Voraussetzung zum Verständnis aller ABAP-Anwendungen. Dabei werden zwar Programme vom Typ *Online-Report* benutzt, diese Programme sind aber keine vollwertigen Reports im engeren Sinn des Wortes. Vielmehr dienen sie nur als Hilfsmittel zur praktischen Demonstration der ersten Anweisungen.

Eine R/3-Anwendung besteht nicht nur aus Quelltext, sondern umfasst auch weitere Elemente. Zu diesen Elementen gehört die so genannte *Oberfläche*. Hinter diesem Begriff verbergen sich vor allem die Menüs, über die Sie eine Anwendung steuern. In nahezu allen Dialoganwendungen und vielen Reports müssen Sie eigene Menüs verwenden. Der dritte Abschnitt beschreibt die Eigenschaften dieser Menüs sowie das entsprechende Pflegewerkzeug.

Echte Reports mit ihren speziellen Eigenschaften und Anweisungen stehen im Mittelpunkt des vierten Unterabschnitts. Einige ABAP-Anweisungen können nur in Reports benutzt werden. Außerdem besitzen Reports eine spezielle Programmstruktur. Abgesehen von der Beschreibung dieser speziellen Anweisungen festigt das Studium dieses Abschnittes Kenntnisse zu den bereits vorgestellten Anweisungen allgemeiner Natur.

Das starre Ablaufschema der klassischen Reports kann durch interaktive Sprachelemente aufgebrochen werden. Der fünfte Unterabschnitt beschäftigt sich insbesondere mit diesen Erweiterungen, wie z.B. Verzweigungen in Teillisten.

Logische Datenbanken erleichtern in Reports die Datenselektion und die Wiederverwendbarkeit von Quelltext. Reports können auf eine logische Datenbank zugreifen, müssen dies aber nicht. Der sechste Abschnitt beschreibt die Bearbeitung und Verwendung von logischen Datenbanken.

Im siebenten Unterabschnitt wird das Dynpro-Konzept genauer erläutert. Dazu ist es zunächst erforderlich, die Struktur einer dialogorientierten Anwendung

zu beschreiben. Erst dann werden die Anweisungen, mit denen diese Programmstruktur realisiert wird, vorgestellt.

Der daran anschließende Abschnitt beschreibt die Verknüpfung zwischen List- und Dialogverarbeitung.

Ein wichtiges Hilfsmittel zur Bereitstellung anwendungsübergreifender Funktionalität und zur Modularisierung von Anwendungen sind Funktions- und Dialogbausteine. Auch für diese Bausteine ist eine eigenständige Programmierweise erforderlich. Einige vorgefertigte, allgemein verwendbare Funktionsbausteine werden durch das System bereitgestellt. Der vorletzte Abschnitt dieses Kapitels führt in die Bearbeitung dieser Bausteine ein, der letzte bietet einige Beispiele zur Nutzung der vom System bereitgestellten Funktionsbausteine an.

Der objektorientierten Erweiterung von ABAP und den damit verbundenen Prinzipien ist ein eigenes Kapitel (Kapitel 4, ABAP Objects) gewidmet. Dieses Kapitel beschäftigt sich ausdrücklich nur mit Kommandos, die unabhängig von der objektorientierten Erweiterung benutzt werden können.

Noch eine Vorbemerkung zur eigentlichen Oberfläche. Ab Version 4.6 steht eine völlig neu gestaltete, auf dem Einsatz von Dynpro-Controls beruhende Oberfläche zur Verfügung. Diese Oberfläche ähnelt den Werkzeugen, die z.B. aus der Windows-Welt bekannt sind. Abbildung 3.1 zeigt Ihnen beispielhaft eine Anwendung in der neuen Darstellungsform.

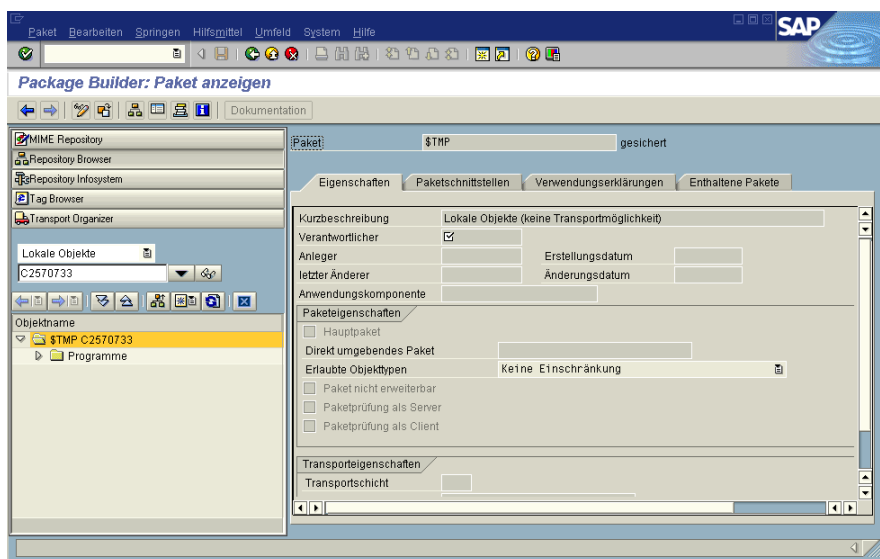


Abbildung 3.1
Beispiel für neue Oberflächen-Elemente

© SAP AG

Die neue Oberfläche ermöglicht eine einfachere und schnellere Navigation zwischen den einzelnen Werkzeugen, stellt aber, abgesehen von einem neuen Editor, noch keine prinzipiell neuen Bearbeitungsmöglichkeiten zur Verfügung. Zur Bearbeitung der diversen Elemente werden auch von der neuen Oberfläche die bereits aus älteren Versionen vorhandenen Werkzeuge benutzt.

Durch die gleichzeitige Darstellung von Navigation und Pflegewerkzeug in einem Fenster vergrößert sich der Platzbedarf im Vergleich zu älteren R/3-Versionen (z.B. 3.x) erheblich. Sie können daher mit der Funktionstaste - den Navigationsbereich ein- und ausblenden.

3.1 Die Entwicklungswerkzeuge

Eine R/3-Anwendung kann sehr komplex werden und besteht aus verschiedenen Elementen. Zur Bearbeitung jedes dieser Elemente stehen eigene Werkzeuge bereit. Um dem Programmierer die Arbeit zu erleichtern, bietet der *Repository Browser* die Gesamtsicht auf alle Elemente einer Anwendung und erlaubt von einer einzigen Oberfläche aus den Zugriff auf alle notwendigen Pflegewerkzeuge. Unter Release 3.x wurde der Repository Browser als Object Browser bezeichnet. Beachten Sie bitte die Abhängigkeit der Begriffe vom Releasestand.

Das erste Programm wird ein einfacher Report sein. Der Begriff *Report* wurde bereits im Kapitel 2 erwähnt. Reports sind Programme, die Auswertungslisten erzeugen, diese auf dem Bildschirm darstellen und bei Bedarf drucken. In seiner Grundform arbeitet der Report nicht dialogorientiert. Allgemeiner ausgedrückt: Reports können nach dem Start ohne weitere Aktivität des Anwenders eine Ausgabe auf dem Bildschirm erzeugen.

Da Reports nicht interaktiv sind, benötigen sie weder Dynpros noch Oberflächen oder Ablauflogik. Ein Report besteht im einfachsten Fall nur aus einem einzigen Quelltext mit einer sehr einfachen Struktur. Hinzu kommt, dass Reports als einzige Programmart sofort aus der Entwicklungsumgebung heraus abgearbeitet werden können. Es bietet sich daher an, derartige Reports für den ersten Einstieg in die Programmierung und die Demonstration der wichtigsten ABAP-Anweisungen zu benutzen.

3.1.1 Das erste Programm: Hello World!

Fast alle Programmierhandbücher beginnen mit demselben Programm. Seine Aufgabe besteht darin, eine einfache Textzeile auf den Bildschirm zu schreiben. Meist lautet diese „Hello World!“. Der praktische Nutzen eines derartigen Programms und der Lerneffekt bezüglich der Sprache halten sich natürlich in Grenzen. Bedeutsam an dieser Art des Einstiegs ist vielmehr der erste Kontakt mit der Entwicklungsumgebung oder den Programmierwerkzeugen. Da diese unter R/3 recht komplex sind, spricht nichts dagegen, auf eben diese Weise zu beginnen.

Erzeugen eines Programms

Alle für die Anwendungsentwicklung erforderlichen Werkzeuge sind in einer so genannten Workbench zusammengefasst. Diese stellt sich dem Anwender als Bereichsmenü dar, das in älteren Versionen der R/3-Software vom R/3-Hauptmenü aus mittels der Menüfunktion WERKZEUGE | ABAP-WORKBENCH oder von beliebiger anderer Stelle mit dem Transaktionscode S001 erreichbar ist. Abbildung 3.2 zeigt Ihnen dieses Bereichsmenü in einem aktuellen System. Zwecks besserer Navigationsmöglichkeit im Gesamtsystem wird es nicht mehr als herkömmliches Pulldown-Menü im Kopf des Fensters angezeigt sondern als Baum im Navigationsbereich der Arbeitsfläche.

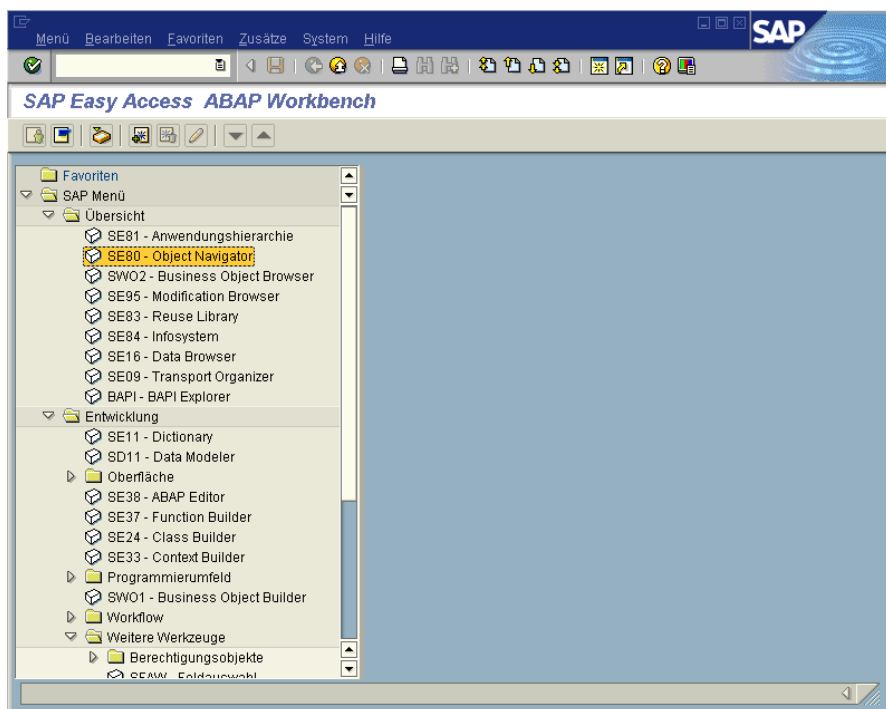


Abbildung 3.2
Die Workbench in der Version 6.x

© SAP AG

In der Workbench finden Sie alle Werkzeuge, die in irgendeiner Form mit der Programmierung von Anwendungen zu tun haben. Sie stellt somit eine funktionsorientierte Zugriffsmöglichkeit dar. Für die Entwicklungstätigkeit besser geeignet ist allerdings der *Object Navigator* (vergleichbar mit dem früheren Repository Browser). Er bietet eine Navigationsmöglichkeit über komplette Entwicklungsprojekte mit allen Bestandteilen. Die für jedes Entwicklungsobjekt erforderlichen Werkzeuge werden vom Object Navigator kontextsensitiv

bereitgestellt. Der Object Navigator ist über den entsprechenden Zweig der Workbench oder den Transaktionscode SE80 erreichbar. Die Oberfläche des Object Navigator (siehe Abbildung 3.3) bietet Ihnen wiederum einen zweigeteilten Arbeitsbereich. Die linke Seite dient zur Navigation innerhalb der unterschiedlichen Bereiche der Anwendung sowie der Auswahl des konkreten Entwicklungsobjekts. Im rechten Teil der Arbeitsfläche wird das Werkzeug zur Bearbeitung eines ausgewählten Entwicklungsobjekts eingeblendet.

Auf Grund der Komplexität des R/3-Systems sowie der Zahl der unterschiedlichen Elemente, Werkzeuge und entwicklungsbegleitenden Aufgaben muss im Object Navigator zunächst eine Auswahl aus einigen Rubriken getroffen werden. Im oberen Teil des Navigationsbereichs finden Sie fünf Drucktasten, von denen bis auf weiteres nur die Drucktaste REPOSITORY BROWSER betätigt werden sollte.

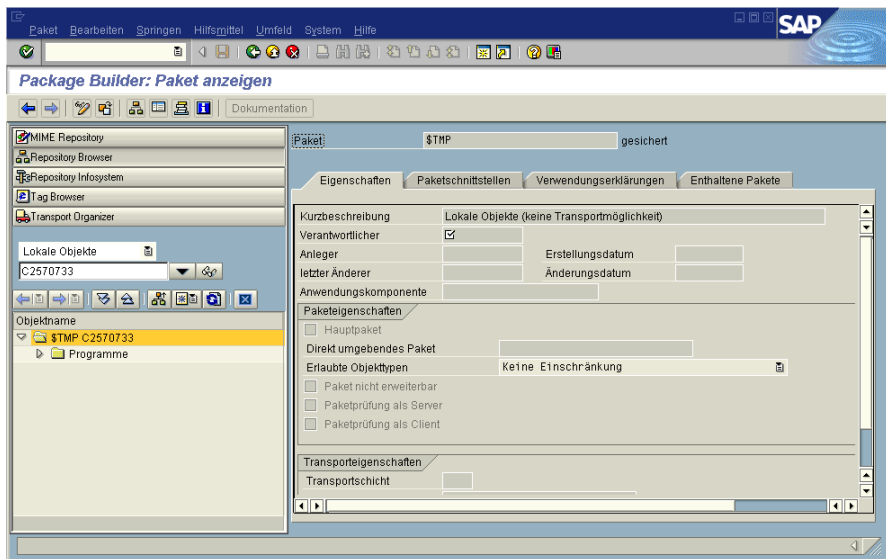


Abbildung 3.3
Grundbild des Object Navigator

© SAP AG

Zum Erzeugen eines neuen Programms ist noch eine weitere Vorgabe notwendig, die für alle Beispiele in diesem Kapitel gilt, sofern nichts anderes erwähnt wird. In dem direkt unter den Drucktasten liegenden Drop-Down-Auswahlfeld sollten Sie die Auswahl markieren, sofern sie nicht bereits aktiv ist (vergleiche Abbildung 3.4). Die Erläuterung der einzelnen Begriffe erfolgt später.

Daraufhin sollte im zweiten Eingabefeld Ihr Benutzername erscheinen. Falls nicht, sollten Sie ihn eintragen und mit der -Taste bestätigen. Die Betätigung der -Taste bzw. die des Refresh-Symbols führt zum Neuaufbau der Objektliste im unteren Teil des Navigationsbereiches. Die Oberfläche des Object Navigator sollte nun der Abbildung 3.3 entsprechen.

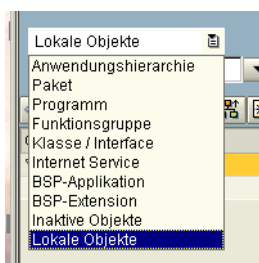


Abbildung 3.4
Auswahlmöglichkeiten für Entwicklungsbereiche
im Object Navigator, © SAP AG

Um ein neues Programm anzulegen, markieren Sie durch einen einfachen Mausklick den obersten Eintrag der Objektliste. Dieser besteht aus dem Kürzel „\$TMP“ als Kennzeichen für lokale Objekte und Ihrem Benutzernamen. Dann rufen Sie durch einen Klick mit der rechten Maustaste das hierarchische Kontextmenü auf und selektieren den Menüeintrag ANLEGEN | PROGRAMM. Abbildung 3.5 zeigt den Object Navigator mit dem aufgeklappten Kontextmenü.

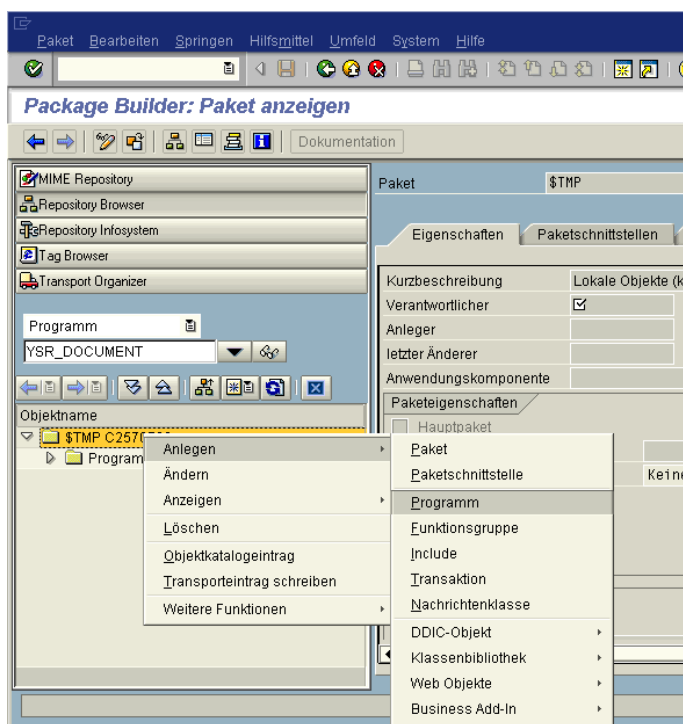


Abbildung 3.5
Kontextmenü zum Anlegen eines neuen Programms

© SAP AG

Das System blendet daraufhin ein kleines Popup ein (Abbildung 3.6), in dem Sie den Namen des Programms eintragen müssen.

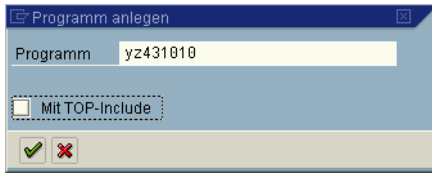


Abbildung 3.6 © SAP AG
Festlegen des Namens des neuen Programms

An dieser Stelle ist vor allem auf einen korrekten Programmnamen zu achten. Das SAP-System generiert an vielen Stellen selbsttätig Programme, wobei deren Namen nach feststehenden Regeln gebildet werden. Weiterhin unterscheidet die Entwicklungsumgebung zwischen Entwicklungssystemen bei SAP und Kundensystemen. Für beide Arten von Systemen existieren unterschiedliche Vorgaben bezüglich der verwendbaren Namen. Für alle Elemente (nicht nur Reports) prüft die Entwicklungsumgebung einige Namenskonventionen. Auf so genannten Kundensystemen dürfen die Namen neuer Programme nur mit den Zeichen Y oder Z beginnen. Auf diese Weise sollen eigene Entwicklungen des Kunden eindeutig von denen des Hauses SAP unterschieden werden. Dementsprechend können in den Entwicklungssystemen direkt bei SAP keine Objekte angelegt werden (zumindest nicht ohne eindrückliche Warnung), die mit diesen beiden Buchstaben beginnen. Eine weitere Differenzierung ist durch einen optionalen Präfix vor dem eigentlichen Namen möglich. Dieser Präfix stellt die Bezeichnung eines Namensraums dar. Darauf soll an dieser Stelle allerdings noch nicht eingegangen werden.

Um Konflikte mit den automatisch generierten Namen zu vermeiden und mehreren Anwendern in Ihrem System die Abarbeitung der Demoprogramme zu ermöglichen, sollten Sie einige Konventionen beachten. Für die Beispiele in diesem Buch werden Namen der Form Yiikknnm benutzt. Für ii setzen Sie bitte Ihre eindeutigen Initialen ein. Damit ist sichergestellt, dass mehrere Benutzer diese Beispiele an einem einzigen System nachvollziehen können. In diesem Buch werden als Initialen die Zeichen Z4 verwendet. Die Zeichenfolge kk steht als Platzhalter für die Kapitelnummer (der zweiten Gliederungsebene), nn ist eine laufende Nummer innerhalb des Kapitels. Zu einigen der vorgestellten Programme können bzw. sollen Sie eigene Modifikationen anlegen. Diese Varianten zum Programm werden mittels der letzten Stelle des Namens unterschieden. Die Ursprungsversion erhält immer das Zeichen 0. Das erste Programm bekommt daher den Namen YZ431010.

Umfangreiche Programme können Sie in mehrere Teildateien (Includes) aufspalten. Die Entwicklungsumgebung unterstützt diese Programmierweise durch automatisches Anlegen eines Rahmenprogramms, das nur aus Include-

Anweisungen besteht. An dieser Stelle ist diese komplexe Programmstruktur allerdings noch nicht notwendig. Sie sollten daher das Flag MIT TOP INCLUDE deaktivieren. Danach können Sie dieses Popup mit der -Taste oder durch Betätigen des entsprechenden Symbols verlassen. Gegebenenfalls erscheint nun ein Popup, dass Sie darüber informiert, dass das neue Programm im Kundennamensraum liegt. Dies ist allerdings von der konkreten Systemeinstellung abhängig. Sie können es ohne weiteres beenden.

Das neue Programm wird nun vom System angelegt. Dazu benötigt es aber eine Reihe zusätzlicher Informationen, die in einer Eingabemaske abgefragt werden. Diese erscheint wiederum als Popup (Abbildung 3.7). Viele Elemente einer Anwendung, so auch ein einfacher Report, besitzen einige Verwaltungsinformationen. Von besonderer Bedeutung sind dabei zwei Felder, in die auf jeden Fall eine Eingabe erfolgen muss. Bei diesen Feldern handelt es sich um eine Kurzbeschreibung des Programms und den Programmtyp.

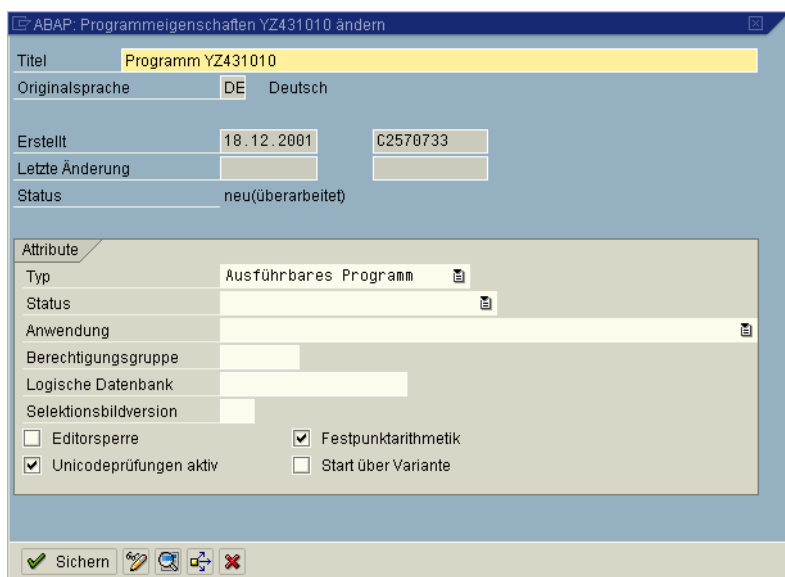


Abbildung 3.7
Pflege der Attribute eines neuen Programms

© SAP AG

Die Kurzbeschreibung im Feld TITEL wird von neueren Versionen der R/3-Software automatisch vorgelegt. Sie können diese Vorgabe jederzeit überschreiben, z.B. mit „Hello-World“. Diese Kurzbeschreibung ist in der Praxis unentbehrlich. Wegen der teilweise vorgeschriebenen Programmnamen, deren Länge zudem begrenzt ist und der großen Anzahl von Programmen im SAP-System gibt nur dieser Titel einen echten Hinweis auf die Funktion des Programms. Der Titel erscheint übrigens auch in den automatisch generierten Standardüberschriften der vom Report erzeugten Liste.

In Kapitel 2 wurden bereits verschiedene Programmtypen erwähnt, z.B. Modul-Pools oder die hier verwendeten Reports. Im Feld TYP wird eben dieser Typ des Programms festgelegt. Diese Typangabe ist zwingend erforderlich. Reports besitzen immer den Typ AUSFÜHRBARES PROGRAMM. Nur Programme mit diesem Typ können direkt abgearbeitet werden! Mit dem hier benutzten Werkzeug können durch Wahl des entsprechenden Typs neben Online-Reports auch andere Programmtypen angelegt werden. Abbildung 3.8 zeigt den zur Verfügung stehenden Wertevorrat in Form einer Eingabehilfe.

The screenshot shows a dialog box titled 'Attribute' with a tabbed interface. The 'Typ' tab is active, displaying a list of program types. The 'Ausführbares Programm' option is selected and highlighted in blue. Other options in the list include 'Include-Programm', 'Modulpool', 'Funktionsgruppe', 'Subroutinenpool', 'Interface-Pool', and 'Class-Pool'. Below the list, there are checkboxes for 'Editorsperre' (unchecked) and 'Unicodeprüfungen aktiv' (checked). A 'Start über Variante' checkbox is also present but unchecked.

Abbildung 3.8 © SAP AG
Eingabehilfe für den Programmtyp

Alle anderen Eingabefelder nehmen optionale Informationen auf. Sie können die vom System getroffenen Vorgaben übernehmen. Nach Eintragen der Attribute für das Programm müssen Sie diese Werte abspeichern. Das geschieht entweder mit der -Taste oder dem entsprechenden Icon. Beim erstmaligen Speichern eines neuen Objekts erfragt das System in einem zum Korrektur- und Transportwesen gehörenden Popup (Abbildung 3.9) den Namen eines so genannten Pakets.

The screenshot shows a dialog box titled 'Objektkatalogeintrag anlegen'. It contains several input fields: 'Objekt' with the value 'R3TR PROG YZ431010', 'Paket' with the value '\$TMP', 'Verantwortlicher' with the value 'C2570733', 'Originalsystem' with the value 'B20', and 'Originalsprache' with the value 'DE Deutsch'. At the bottom, there are three icons: a floppy disk icon labeled 'Lokales Objekt', a person icon labeled 'Sperrübersicht', and a red X icon.

Abbildung 3.9 © SAP AG
Popup des Korrektur- und Transportwesens

Pakete dienen zur Gruppierung der verschiedenen Entwicklungsobjekte innerhalb des R/3-Systems. Logisch zusammengehörende Elemente werden einem gemeinsamen Paket zugeordnet. Ein Paket ist eine Weiterentwicklung der aus älteren Systemen bekannten Entwicklungsklasse. Die Eigenschaften eines Pakets bestimmen beispielsweise auch, ob Objekte aus dem aktuellen System heraus in andere Systeme transportiert werden können oder nicht. Üblicherweise werden im R/3-System Programme in einem eigenständigen Testsystem entwickelt. Nach der Entwicklung und einem Programmtest werden die Programme dann mittels des Korrektur- und Transportwesens auf das gewünschte Zielsystem übertragen. Die korrekte Pflege der Eigenschaften der diversen Pakete und die Zuordnung der Anwendungen zu diesen Paketen ist somit von entscheidender Bedeutung für die Weitergabe von Anwendungen an nachfolgende Systeme. Zu dieser Problematik folgt später ein etwas ausführlicherer Abschnitt.

Damit nicht jedes kleine Test- oder Hilfsprogramm in die offizielle Entwicklung eingeht, können Entwicklungsobjekte auch als so genannte *Lokale Objekte* gekennzeichnet werden. Im dargestellten Popup dient dazu eine spezielle Drucktaste. So deklarierte Objekte werden durch das Transportwesen nicht erfasst. Sie können daher auch nicht in andere Systeme übertragen werden. Alle Demoprogramme sollten als lokale Objekte angelegt werden. Sie können das Popup also ohne weitere Eingabe durch Betätigen der Schaltfläche **LOKALES OBJEKT** beenden. Gleichbedeutend mit diesem Schalter ist die Eingabe des Pakets „\$TMP“ im dafür vorgesehenen Eingabefeld. Da im Grundbild des Object Navigator die lokalen Objekte als Arbeitsvorrat ausgewählt wurden, erscheint im Popup der entsprechende Paketname \$TMP bereits als Vorgabe.

Dieses Popup wird im weiteren Verlauf der Programmierung für jedes neue Objekt erscheinen. Es ist immer auf die erwähnte Weise zu behandeln. Im weiteren Verlauf des Buches wird nicht mehr auf diesen Arbeitsschritt eingegangen.

Nach Auswahl des Pakets befinden Sie sich wieder im Object Navigator. Das neue Programm wurde in die Objektliste aufgenommen. Im rechten Teil des Bildschirms erscheint nun der Programmeditor (Abbildung 3.10). Er zeigt das momentan nur aus einer einzigen echten Anweisung und einigen Kommentarzeilen bestehende Programm an. Bei Online-Reports generiert das System automatisch eine Programmzeile, die aus dem Schlüsselwort **REPORT** und dem Programmnamen besteht.

Beim Editor handelt es sich um den so genannten Frontend-Editor. Dieser wird durch ein Control im SAPGUI gebildet, die gesamte Programmlogik liegt somit auf dem lokalen Rechner. Dieses vorgehen stellt eine Abkehr vom Editor der älteren R/3-Versionen dar, der auf den Dialogelementen des SAPGUI und der herkömmlichen Dynpro-Technik beruhte. Der optisch überarbeitete, Serverbasierte Editor ist weiterhin verfügbar. Für die Abarbeitung dieses Beispiels ist es bedeutungslos, welcher Editor benutzt wird.

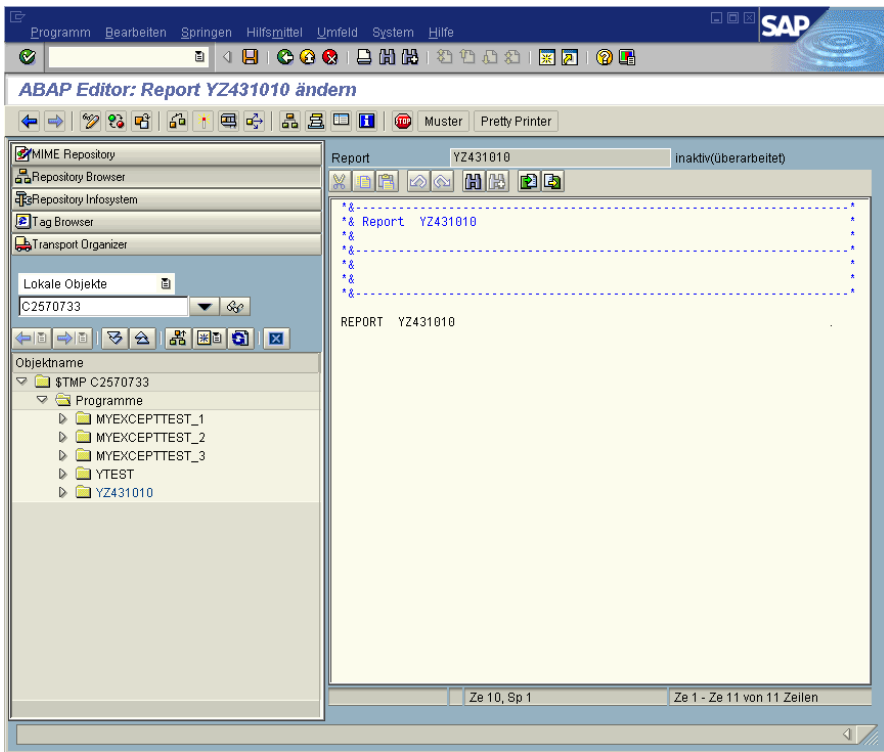


Abbildung 3.10
Editor mit Programmkopf

© SAP AG

An dieser Stelle können Sie das Programm erfassen. Dazu müssen Sie nur eine einzige Zeile ergänzen:

```
REPORT YZ331010.  
WRITE 'Hello world!'.  

```

Durch einen einfachen Mausklick können Sie den Cursor auf die Zeile unter der REPORT-Anweisung platzieren. Dann tragen Sie die zusätzliche Anweisung ein. Innerhalb des Editors können Sie mit der -Taste Zeilenumbrüche einfügen. In der ersten Zeile wird dem System mittels des Schlüsselwortes REPORT mitgeteilt, dass es sich bei den folgenden Anweisungen um ein eigenständiges Programm handelt. Völlig äquivalent zu REPORT ist übrigens die alternativ verfügbare Anweisung PROGRAM. Die zweite Zeile schreibt einen Text in das Ausgabefenster.

ABAP-Anweisungen können formatfrei eingegeben werden, also über mehrere Zeilen reichen oder beliebig eingerückt werden. Dies erfordert, das Ende eines ABAP-Kommandos eindeutig zu kennzeichnen. Im ABAP dient dazu ein Punkt.