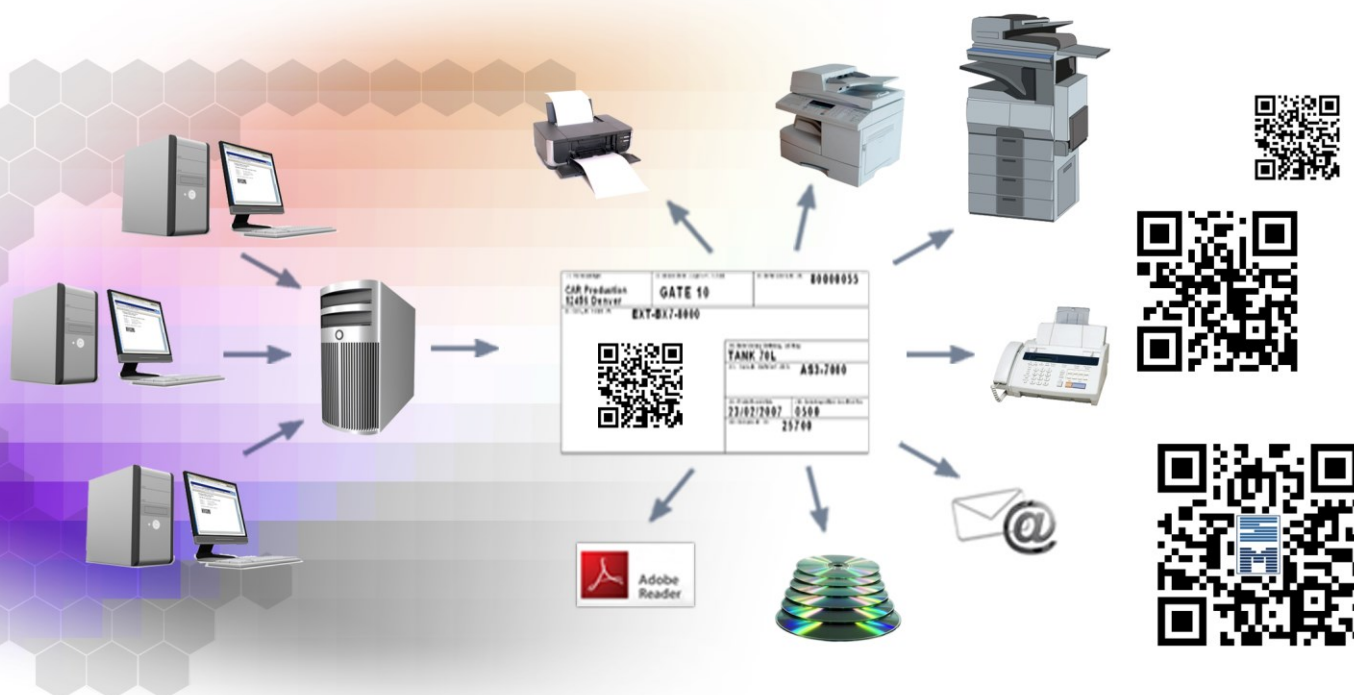




SUCHY MIPS



RBarc/QRCode Für SAP® Systeme Version 2 Handbuch

Stand Dezember 2021

© Copyright 2007-2021 Suchy MIPS GmbH, München. Alle Rechte vorbehalten.

Diese Dokumentation dient ausschließlich Informationszwecken und kann jederzeit ohne vorherige Ankündigung verändert werden. Suchy MIPS übernimmt keine Gewährleistung oder Garantie hinsichtlich der Richtigkeit und Genauigkeit der Angaben in dieser Dokumentation.

Diese Dokumentation oder Teile daraus dürfen nicht vervielfältigt, in Datenbanken gespeichert oder in irgendeiner Form elektronisch, fotomechanisch, auf Datenträgern oder auf irgendeine andere Weise übertragen werden, ohne dass vorher die schriftliche Zustimmung von Suchy MIPS eingeholt wurde. Verstöße gegen das Copyright ziehen zivil- und strafrechtliche Konsequenzen nach den internationalen Copyright Gesetzen nach sich.

Die Dokumentation beschreibt Installation und Anwendung einer durch Copyright geschützten Software. Die Benutzer sind verantwortlich für die Einhaltung aller einschlägigen Urheberrechtsgesetze und der jeweiligen Lizenzvereinbarung.

RBarc/QRCode ist geistiges Eigentum der Fa. Suchy MIPS GmbH.

MS-Windows ist eine eingetragene Marke der Microsoft Corporation, Inc.

R/3, SAP, mySAP and Netweaver sind eingetragene Marken der SAP AG.

Alle verwendete Firmen-, Markennamen und Warenzeichen sind Eigentum der jeweiligen Inhaber und dienen lediglich zur Identifikation und Beschreibung der Produkte und Dienstleistungen. Eine Haftung für Richtigkeit, Vollständigkeit und Aktualität wird nicht übernommen.

Suchy MIPS GmbH
Prinzregentenstrasse 128

81677 München
Deutschland

Tel.: +49 (0) 89 - 944 19 77 - 0
Fax: +49 (0) 89 - 944 19 77 - 13
E-Mail: info@suchymips.de
Internet: www.suchymips.de

Inhaltsverzeichnis

Nr.	Titel	Seite
1	Einführung	3
2	Systemvoraussetzungen	4
3	Installation	5
4	Unterschiede zwischen der Demo- und Vollversion von RBarc/QRCode	6
5	Testen der Installation von RBarc/QRCode	7
5.1	Testen mit SAPscript	7
5.2	Testen mit Smart Forms	8
5.3	Testen mit Interactive Forms	9
6	QR-Code in Formulare einbinden	10
6.1	Das Funktionsprinzip von RBarc/QRCode	10
6.2	Typisierung der Variablen	11
6.3	Optionale Parameter zur Definition der Barcode Eigenschaften.	12
6.4	Drucken aus einem SAPscript Formular	15
6.4.1	Barcode Definition im SAPscript Formular	15
6.5	QR-Code Druck mit Smart Forms	19
6.5.1	Barcode Definition im Smart Forms Formular	19
6.6	QR-Code Druck mit Interactive Forms	24
6.6.1	Erweiterung der Interactive Forms Schnittstelle	24
6.6.2	Erweiterung des Interactive Forms Formulars	29
6.7	Definition der Barcode Eigenschaften	32
7	Beispiel für eine Aufgabenstellung in einem SAPscript Formular	46
7.1	Aufgabe	46
8	Beispiel für Aufgabenstellung in einem Smart Forms Formular.	48
8.1	Aufgabe	48
9	Anhang 1: Neues Paket (Entwickungsklasse) erstellen	51
10	Anhang 2: Ein Include in der ABAP Workbench anlegen.	53
11	Anhang 3: Ein Programm in der ABAP Workbench anlegen.	56
12	Anhang 4: Inhalt der Installationsdateien in ABAP Programme einfügen.....	59
13	Anhang 5: Aktivieren von Includes und Programmen	60
14	Anhang 6: Ein ABAP Programm ausführen.	62
15	Anhang 7: Ein SAPscript Formular importieren (hochladen).	63
16	Anhang 8: Ein Smart Forms Formular hochladen	65
17	Anhang 9: Smart Forms Formular testen	67
18	Anhang 10: Eine Interactive Forms Schnittstelle hochladen.	70
19	Anhang 11: Ein Interactive Forms Formular hochladen	71
20	Anhang 12: Ein Interactive Forms-Formular testen	72
21	Anhang 13: Swiss QRCode	74

1 Einführung

RBarc/QRCode ist ein ABAP Programm zur on-the-fly Erzeugung des 2D Barcodes QR-Code auf SAP Systemen (R/3, MySAP, ERP, Netweaver usw.). Die entsprechenden Funktionen stehen in **SAPscript**-, **Smart Forms**- und **Interactive Forms** Formularen zur Verfügung.

RBarc/QRCode ist kompatibel zur ISO QR-Code Model 2 Spezifikation

Die mit **RBarc/QRCode** generierten 2D Barcodes werden innerhalb des SAP Systems so erzeugt, dass sie integraler Bestandteil des auszugebenden Dokumentes werden. Die daraus resultierenden Vorteile sind, dass Dokumente mit Barcodes auf beliebigen Druckern ausgegeben können - unabhängig vom Hersteller und der verwendeten Druckersprache. Hardwareerweiterungen werden nicht benötigt. Ein weiterer Vorteil ist, dass Barcodes auf den Dokumenten immer sichtbar sind, auch wenn das Dokument gefaxt, gemailt, in ein PDF-Dokument umgewandelt oder archiviert wurde.

Wichtig:

RBarc/QRCode ändert den SAP Standard nicht. Die zu **RBarc/QRCode** gehörenden Programme beginnen mit dem Buchstaben "Z" und werden im sog. Kundenraum installiert. Somit werden keine wichtigen Programmteile bei eventuellen Updates überschrieben.

2 Systemvoraussetzungen

RBarc/QRCode setzt ein installiertes SAP® System R/3 vers. 3.x oder höher voraus (auch MySAP ERP und NetWeaver).

Wichtig:

RBarc/QRCode ist für Administratoren (Installation) und Entwickler (Barcode Gestaltung und Einbindung ins Formular) bestimmt. Konsequenterweise sind gute Kenntnisse in Basis bei der Installation Voraussetzung. Für das Einbinden von Barcodes in **SAPscript**-, **Smart Forms**- oder **Interactive Forms**-Formulare sollten mindestens Grundkenntnisse in dem jeweiligen Formularverfahren als auch in ABAP Programmierung vorhanden sein.

Alle Anwender, die Dokumente mit von **RBarc/QRCode** erzeugten Barcodes drucken, müssen folgende Berechtigung für das Objekt **S_BDS_DS** besitzen:

ACTVT = 01, 02, 03 und 06
CLASSNAME = DEVC_STXD_BITMAP
CLASSTYPE = OT

Die entsprechenden Einstellungen können ggf. mit der Transaktion "**PFCG**" durchgeführt werden. Die User Rolle muss die Transaktion SE78 und die Autorisierung für **BC-SRV-KPR-BDS** (Technischer Name **S_BDS_DS**) enthalten. **S_BDS_DS** gehört zu den **Basis-Central-Funktionen**.

Bei fehlenden Rechten des Benutzers erscheint kein Barcode auf dem Ausdruck oder die generierten Barcodes werden aus dem System nicht gelöscht.

3 Installation

Die Installationsschritte werden hier in Kurzform beschrieben. Falls Sie mit manchen Vorgängen nicht vertraut sind (z.B. Anlegen eines Programms), finden Sie bei jedem Schritt einen Verweis auf die Seite, auf der der Vorgang genauer beschrieben wird. Um zu detaillierten Beschreibung automatisch zu springen, klicken Sie mit der linken Maustaste auf den Querverweis.

Im Verzeichnis **Install** der gelieferten CD befinden sich 4 Dateien: **ZQRCODE.INC**, **Z_SET_QRCODE.PRG**, **ZSS_QR_PRINT.PRG** und **ZAF_QR_PRINT.PRG** die zur Installation benötigt werden.

1. Starten Sie das **SAP GUI** und loggen sich auf dem SAP System als Administrator an.
2. Starten Sie den **Objekt Navigator** (Transaktion **SE80**) und erstellen ein neues Paket (Entwickungsklasse) mit dem Namen **ZQRCODE** (empfohlen).
(Gehe zu Seite 51: [Anhang 1: Neues Paket \(Entwickungsklasse\) erstellen](#))
3. Legen Sie in dem Paket (Entwickungsklasse) **ZQRCODE** ein Include mit dem Namen **ZQRCODE** an und löschen den automatisch erzeugten Inhalt.
(Gehe zu Seite 53: [Anhang 2: Ein Include in der ABAP Workbench anlegen](#))
4. Kopieren Sie in das neu erstellte Include **ZQRCODE** den Inhalt der Datei **ZQRCODE.INC** aus dem Verzeichnis **Install**, sichern und aktivieren es.
(Gehe zu Seite 59: [Anhang 4: Inhalt der Installationsdateien in ABAP Programme einfügen](#))
5. Legen Sie in dem Paket **ZQRCODE** ein Programm (Report) mit dem Namen **Z_SET_QRCODE** an und löschen den ev. automatisch erzeugten Inhalt.
(Gehe zu Seite 51: [Anhang 3: Ein Programm in der ABAP Workbench anlegen](#))
6. Kopieren Sie in das neu erstellte Programm **Z_SET_QRCODE** den Inhalt der Datei **Z_SET_QRCODE.PRG** aus dem Verzeichnis **Install**, sichern und aktivieren es.

Damit ist die Installation von RBarc/QRCode abgeschlossen.

Jetzt können noch Testobjekte installiert werden, mit denen je ein Smartforms, Interactiveforms und Sapscrip Formular mit einem QRCode Barcode gedruckt werden können.

7. Legen Sie in dem Paket **ZQRCODE** ein Programm mit dem Namen **ZSF_QR_PRINT** an und löschen Sie den ev. automatisch erzeugten Inhalt.
8. Kopieren Sie in das neu erstellte Programm **ZSF_QR_PRINT** den Inhalt der Datei **ZSF_QR_PRINT.PRG** aus dem Verzeichnis **Test-Smartforms**, sichern und aktivieren es.
9. Legen Sie in dem Paket **ZQRCODE** ein Programm mit dem Namen **ZIF_QR_PRINT** an und löschen Sie den ev. automatisch erzeugten Inhalt.
10. Kopieren Sie in das neu erstellte Programm **ZIF_QR_PRINT** den Inhalt der Datei **ZIF_QR_PRINT.PRG** aus dem Verzeichnis **Test-Interactiveforms**, sichern und aktivieren es.
11. Legen Sie in dem Paket **ZQRCODE** ein Programm mit dem Namen **ZSS_QR_PRINT** an und löschen Sie den ev. automatisch erzeugten Inhalt.
12. Kopieren Sie in das neu erstellte Programm **ZSS_QR_PRINT** den Inhalt der Datei **ZSS_QR_PRINT.PRG** aus dem Verzeichnis **Test-Sapscrip**, sichern und aktivieren es.

(Gehe zu Seite 60: [Anhang 5: Aktivieren von Includes und Programmen](#))

4 Unterschiede zwischen der Demo- und Vollversion von RBarc/QRCode

- Die einzige, in der Demoversion unterstützte QRVersion ist die Version 1 mit ECC_Level "L" mit der etwas 25 alphanumerischem 41 numerische Zeichen oder 17 Bytes kodiert werden können. Im gemischten Modus oder bei Verwendung von Sonderzeichen, wie z.B. GS (Group Separator) ist die kodierbare Datenmenge nicht einfach vorhersehbar, da beim Wechsel von Alphabet zu Alphabet ein gewisser Datenoverhead entsteht, deren Menge nicht nur von den verwendeten Alphabeten sondern auch von der Verteilung der Zeichen abhängt.
- Die Ziffer „1“ wird durch eine „0“ ersetzt und umgekehrt.

Wichtig:

Falls Sie Tests mit echten Daten durchführen möchten, kontaktieren Sie uns um eine Teststellung zu vereinbaren. Sie haben die Möglichkeit die Software 4 Wochen lang zu testen und genießen in dieser Zeit den vollen technischen Support.

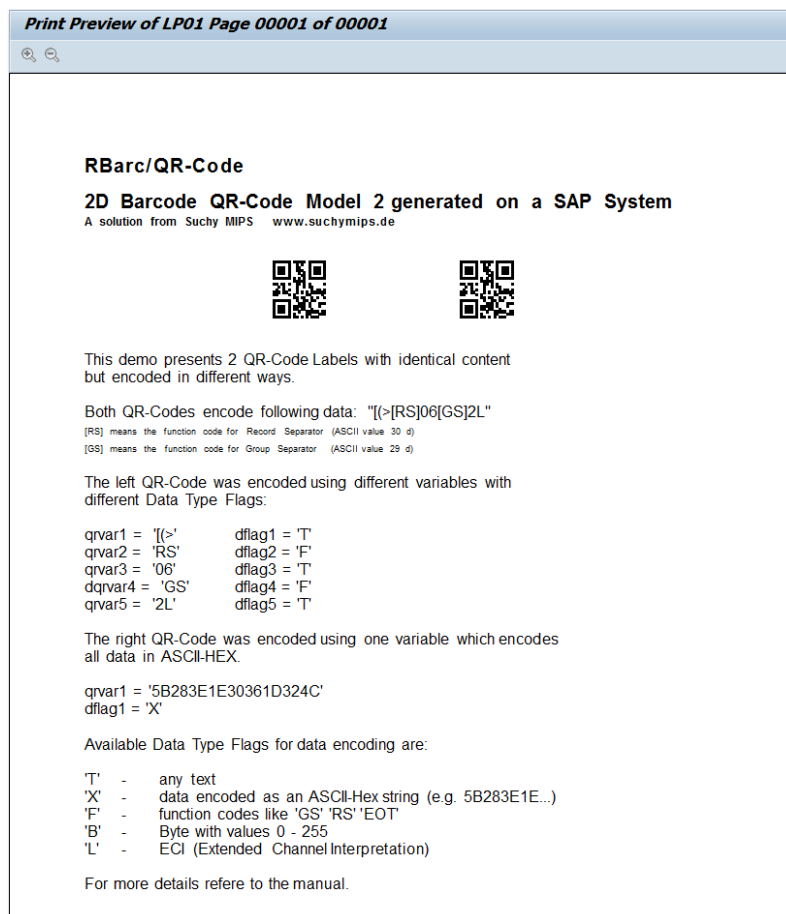
5 Testen der Installation von RBarc/QRCode

Im Verzeichnis **“TestForms”** der Installations-CD befinden sich jeweils ein **SAPscript**-, ein **Smart Forms**- und ein **Interactive Forms**- Formular.

5.1 Testen mit SAPscript

1. Starten Sie das SAP Standardprogramm **RSTXSCR**P und importieren das **SAPscript** Formular **ZSS_QR_FORM** aus der Datei **ZSS_QR_FORM.FOR**, die sich in dem Verzeichnis **TestForms** befindet. (Gehe zu Seite 63: [Anhang 7: Ein SAPscript Formular importieren \(hochladen\)](#))
2. Um das **SAPscript** Testformular **ZSS_BC_FORM** zu drucken, führen Sie das Programm **ZSS_QR_PRINT** aus. (Gehe zu Seite 62: [Anhang 6: Ein ABAP Programm ausführen](#)).

Das Programm **ZSS_QR_PRINT** druckt das **SAPscript** Formular **ZSS_QR_FORM**. Sie können das Formular auf dem Drucker ausgeben, oder in der Druckvorschau ansehen. Das Ergebnis sollte in etwa wie folgt aussehen:



Um einen Barcode in ein eigenes SAPscript Formular einzubinden, lesen Sie weiter auf Seite 15 unter [Barcode Definition im SAPscript Formular](#)

5.2 Testen mit Smart Forms

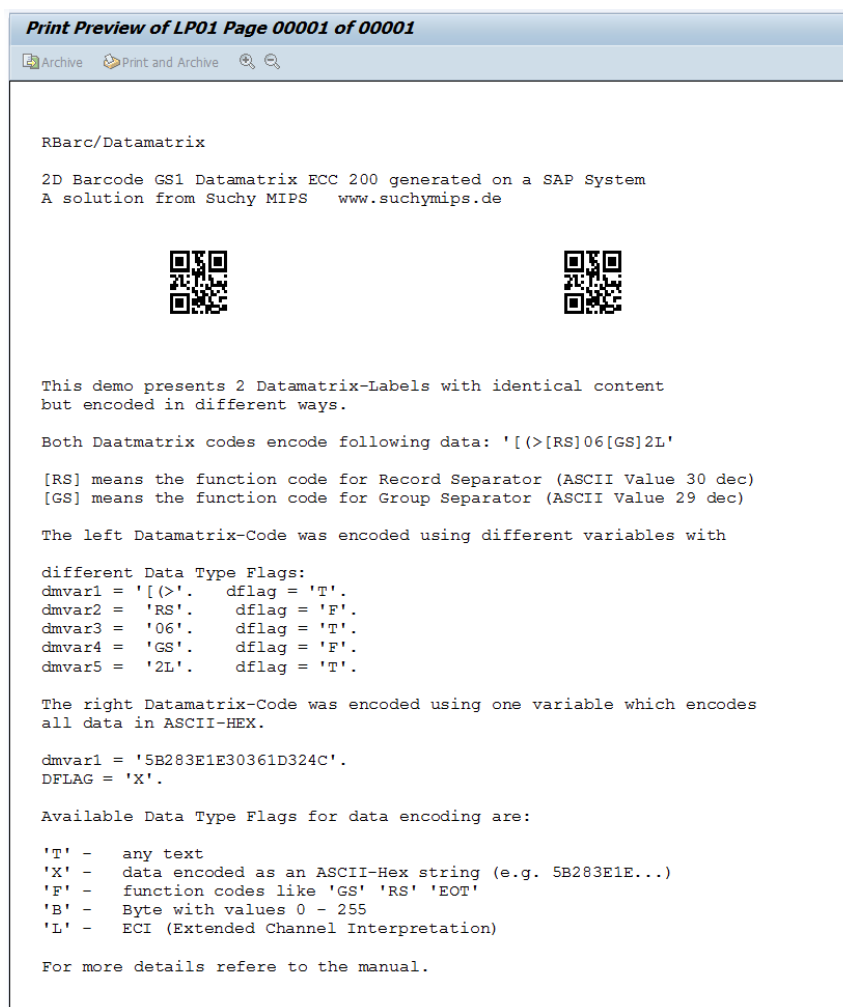
1. Starten Sie die Transaktion „**Smartforms**“ und laden Sie das **Smart Forms** Formular **ZSF_QR_FORM** aus der Datei **zsf_qr_form.xml**, die sich im Verzeichnis **TestForms** befindet, hoch.
(Gehe zu Seite 65: [Anhang 8: Ein Smart Forms Formular hochladen](#)).

Achtung:

Diese Funktion ist ab R/3 Vers. 4.7 verfügbar. Bei R/3 Vers. 4.6x müssen Sie ein **Smart Forms** Formular von Hand erstellen, siehe dazu: Gehe zu Seite 19: [QR-Code Druck mit Smart Forms](#).

2. Um das **Smart Forms** Testformular **ZSF_QR_FORM** zu drucken, öffnen Sie das Formular und wählen Sie aus dem Menü „**Formular / Testen**“.
(Gehe zu Seite 67: [Anhang 9: Smart Forms Formular testen](#)).

Sie können das Formular auf dem Drucker ausgeben, oder in der Druckvorschau ansehen. Das Ergebnis sollte in etwa wie folgt aussehen:



Um einen Barcode in ein eigenes **Smart Forms** Formular einzubinden, lesen Sie weiter unten, unter: „Drucken von Barcodes aus **Smart Forms**“ (Gehe zu Seite 19: [QR-Code Druck mit Smart Forms](#))

5.3 Testen mit Interactive Forms

1. Starten Sie die Transaktion „**SFP**“ und laden Sie die **Interactive Forms**-Schnittstelle **ZIF_QR_INTERFACE** aus der Datei **zif_qr_interface.xml**, die sich im Verzeichnis **TestForms** befindet, hoch
(Gehe zu Seite 70: [Anhang 10: Eine Interactive Forms Schnittstelle hochladen](#))
2. Starten Sie die Transaktion „**SFP**“ und laden Sie das **Interactive Forms**-Formular **ZIF_QR_FORM** aus der Datei **zif_qr_form.xml**, die sich im Verzeichnis **TestForms** befindet, hoch
(Gehe zu Seite 71: [Anhang 11: Ein Interactive Forms Formular hochladen](#))
3. Um das **Interactive Forms** Testformular **ZIF_QR_FORM** zu drucken, öffnen Sie das Formular und wählen Sie aus dem Menü „**Formular / Testen**“.
(Gehe zu Seite 67: [Anhang 12: Ein Interactive Forms-Formular testen](#)).

Sie können das Formular auf dem Drucker ausgeben, oder in der Druckvorschau ansehen. Das Ergebnis sollte in etwa wie folgt aussehen:



Um einen Barcode in ein eigenes **Interactive Forms** Formular einzubinden, lesen Sie weiter unten, unter: „Drucken von Barcodes aus **Interactive Forms**“ (Gehe zu Seite 24: [QR-Code Druck mit Interactive Forms](#))

6 QR-Code in Formulare einbinden

QR-Code ist ein sog. 2D Barcode, mit dem eine große Anzahl von Zeichen kodiert werden kann. (bis zu 7089 numerische oder 4296 alfanumerische Zeichen oder 2953 Bytes). In ERP Anwendungen handelt es sich dabei meistens nicht um fließenden Text, sondern um Inhalte bestimmter Datenfelder aus der Datenbank, die in einem Barcode Label kodiert werden sollen, womöglich mit Feldseparatoren voneinander getrennt und mit einem Daten Vor- und Nachspann versehen.. Im Folgenden wird das Funktionsprinzip von **RBarc/QRCode** als auch die Regeln für die Implementierung von QR-Code Barcodes in den verschiedenen SAP Formulartechnologien: **SAPscript**, **Smart Forms** und **Interactive Forms** beschrieben.

Hilfreiche Beispiele mit Aufgabenstellung aus der Praxis und Implementierungslösungen finden Sie in den Kapiteln:

[Beispiel für eine Aufgabenstellung in einem SAPscript Formular](#)

[Beispiel für Aufgabenstellung in einem Smart Forms Formular](#)

6.1 Das Funktionsprinzip von RBarc/QRCode.

Das Funktionsprinzip von **RBarc/QRCode** sieht wie folgt aus:

- Alle Variablen, die im QR-Code kodiert werden sollen, werden typisiert und in eine interne Schnittstellentabelle paarweise (Variable / Typ) geschrieben. Die Typisierung der Variablen wird im nächsten Abschnitt beschrieben.
- Optional werden Parameter für Barcode Eigenschaften, wie z.B. **Modulgröße**, **Versionsnummer** und **Grafikauflösung** in die Schnittstellentabelle geschrieben. Die optionalen Parameter werden im übernächsten Abschnitt beschrieben. Wenn optionale Parameter nicht definiert werden, dann werden für diese Parameter Standardwerte automatisch von **RBarc/QRCode** gesetzt.
- Die Schnittstellentabelle mit den zu verschlüsselnden Variablen, deren Typen als auch eventuell definierten optionalen Parametern wird an eine Formroutine des Programms **Z_SET_QRCODE** übergeben. Der Namen der Formroutine ist anders für jede SAP-Formulartechnologie und lautet:

für SAPscript : **GEN_QRCODE_SS**

für Smart Forms: **GEN_QRCODE_SF**

für Interactive Forms: **GEN_QRCODE_IF**

- Während der Laufzeit erzeugt das Programm **Z_SET_QRCODE** den Barcode dynamisch und übergibt dessen einmaligen, dynamisch erzeugten Namen bzw. den Barcode als binäres Objekt im Fall von Interactive Forms an das aufrufende Formular.
- Im Formular wird der Barcode dynamisch eingebunden.
- Anschließend wird der Barcode aus dem System gelöscht.

Bemerkung:

Die Übergabe aller Parameter über eine Schnittstellentabelle erlaubt dass Kodieren einer beliebigen Anzahl von Variablen, ohne dass jedes Mal die Schnittstelle angepasst werden müsste.

Die Schnittstelle zwischen dem Formular und dem Programm **Z_SET_QRCODE** ist im Prinzip für alle SAP Formulartechnologien gleich. Technologiebedingt gibt es jedoch Implementierungsunterschiede zwischen **SAPscript**, **Smart Forms** und **Interactive Forms**. Deshalb werden Einzelheiten der Implementierung in den entsprechenden Kapiteln für die jeweilige Formulartechnologien beschrieben.

6.2 Typisierung der Variablen

Der Barcode QR-Code kann alle ASCII Werte im Bereich von 0 – 255 (dezimal) kodieren. Da nicht all diese Werte in einem SAP System als Zeichen darstellbar sind, entwickelten wir ein spezielles Verfahren, das Ihnen erlaubt alle Zeichenwerte zu kodieren. Dazu müssen die Variablen typisiert werden, bevor sie in die Schnittstellentabelle geschrieben werden. Dies geschieht mit dem Parameter "**dflag**". Folgende Indikatoren (Werte für "**dflag**") sind gültig:

'T' jede alfanumerische Variable, wie z.B. 'ABCDabcd12345AbCd'.

'X' – Variablen im ASCII HEX Format 0x00 – 0xFF. Beispiel: die Variable '**123ABC**' mit **dflag = 'T'** könnte in Form von '**303132414243**' mit **dflag = 'X'** übergeben werden. Auf diese Weise lassen sich auch Zeichen kodieren, die sonst nicht darstellbar sind, wie z.B. eine binäre 0 als ASCII-HEX '00'.

'F' – Variable ist ein Funktionszeichen. Es kann nur eine Variable dieses Typs pro Tabellenzeile übergeben werden. Wenn mehrere Zeichen dieses Typs nebeneinander kodiert werden müssen, muss jedes Zeichen einzeln mit dem dazugehörigen Typ (dflag) in die Schnittstellentabelle eingetragen werden. Folgende Funktionszeichen werden unterstützt:

- ASCII Zeichen 0 bis 31:
'NUL','SOH','STX','ETX','EOT','ENQ','ACK','BEL','BS','HT',
'LF','VT','FF','CR','SO','SI','DLE','DC1','DC2','DC3','DC4',
'NAK','SYN','ETB','CAN','EM','SUB','ESC','FS','GS','RS','US'.
- Das EAN Funktionszeichen: '**FNC1**'
- Das EAN Funktionszeichen: '**FNC2**'. Dem Bezeichner **FNC2** muss unmittelbar die zweistellige, alfanumerische Applikationsnummer folgen (z.B. '**FNC2A7**'
- Das "Extended Channel Interpretation" Zeichen: '**ECI**'

'B' – Variable stellt einen Binärwert in Dezimalformat dar. Es können Werte von 0 bis 255 übergeben werden. Es darf nur ein Wert pro Schritt übergeben werden. Wenn mehrere Zeichen dieses Typs nebeneinander kodiert werden müssen, muss jedes Zeichen einzeln, zusammen mit dem dazugehörigen Typ (dflag) übergeben werden. Auch mit dieser Methode lassen sich - ähnlich wie beim Typ 'X' alle ASCII Werte von 0 bis 255 verschlüsseln.

'L' – Die Variable stellt einen sog. "Extended Channel Interpretation" Wert da. Jede Variable dieser Art muss mit '**ECI**' beginnen, gefolgt von einer 6-stelligen Nummer. Der ECI bleibt gültig, bis nicht ein anderer ECI gesendet wird oder bis zum Ende der Daten. Gültige ECI Nummern sind 0 bis 999999. Der Standard ECI Wert beträgt 000003 (ASCII für Zeichen 0 bis 127 und ISO 8859-1 für Zeichen 128 bis 255. Mit dem ECI wird eine bestimmte Zeicheninterpretation erzwungen (Standard = Westeuropäisch). Bitte beachten Sie, dass ECI nur von speziellen Scannern verarbeitet werden kann. Eine genaue Beschreibung des ECI Protokolls finden Sie in dem Dokument "*Extended Channel Interpretation (ECI) Assignments*". Beispiel: ECI000978.

6.3 Optionale Parameter zur Definition der Barcode Eigenschaften.

Optionale Parameter dienen der Steuerung von äußeren Barcode Eigenschaften. Dazu gehören die **Größe des Barcodes**, die **Versionsnummer**, die **Fehlerkorrekturlevel (ECC_LEVEL)**, die **Auflösung der Grafik** und die Einstellung der Wechselwirkung der Grafik mit anderen Objekten auf der Seite. Wird ein Parameter nicht definiert, dann wird er automatisch auf den Standardwert gesetzt.

QRVersion Die Versionsnummer des QR-Codes gemäß ISO-Spezifikation. Die ISO-Norm spezifiziert 40 Ausprägungen des QR-Codes, die hier mit **qrversion** gewählt werden können. Normalerweise versucht man die zu kodierenden Daten in einer möglichst kleinen Matrix unter zu bringen. In manchen Anwendungen ist es jedoch erforderlich, eine Konstante Barcode Größe zu erhalten. In diesem Fall muss ein Wert für **qrversion** vorgegeben werden. Der Wert sollte so gewählt werden, dass möglichst alle in Frage kommenden Daten darin Platz haben. Erweist sich die gewählte Versionsnummer während der Laufzeit als zu klein wird eine Fehlermeldung ausgegeben und der Barcode wird nicht erzeugt.

Zulässige Werte: 0 bis 40.

Standardwert: 0 (automatische Berechnung der kleinstmöglichen Version).



Beide Barcodes verschlüsseln den gleichen Text "Suchy MIPS"

Eine vollständige Auflistung aller QR Versionen finden Sie im Kapitel [Definition der Barcode Eigenschaften](#)

RES

Angabe der Auflösung, mit der die Barcode Grafik generiert werden soll. Je kleiner die Auflösung, desto schneller wird der Barcode generiert und desto weniger Daten werden erzeugt.

Eine Verdoppelung der Auflösung bedeutet eine Vervierfachung der Datenmenge bei gleich bleibender Barcodegröße oder eine Verkleinerung des Barcodegröße um das Vierfache bei gleich bleibendem **X_DIM**. Da die Module eines QR-Code Barcodes quadratisch sind, hat die Auflösung keinen entscheidenden Einfluss auf seine Qualität. Höhere Auflösungen sollten nur dann gewählt werden, wenn eine gewünschte Barcode Größe mit der Standardauflösung nicht erreicht werden kann. Falls **RES** erhöht wird, ohne dass **X_DIM** geändert wird, dann verkleinert sich der Barcode und seine Größe kann genauer mit **X_DIM** gesteuert werden.

Falls eine Höhere Auflösung bei gleich bleibender Barcodegröße gewünscht wird, dann muss der Parameter **X_DIM** um den gleichen Faktor wie die Auflösung erhöht werden.

Zulässige Werte: >0 (Pixel pro Zoll)

Standardwert: 150



RES = 150 dpi

RES = 300 dpi

Beide Barcodes verschlüsseln den gleichen Text "Suchy MIPS"

X_DIM

Horizontale Größe des kleinsten Moduls. Je größer **X_DIM** desto größer der Barcode. Die Einheit ist **1/RES** Zoll. Je höher die Auflösung desto kleiner ist die **X_DIM** Einheit und desto feiner kann die Barcodegröße bestimmt werden.

Zulässige Werte: 1 bis 100.

Standardwert: 4



X_DIM = 5 bei 150 dpi



X_DIM = 10 bei 150 dpi

Beide Barcodes verschlüsseln den gleichen Text "Suchy MIPS"

Eine detaillierte Beschreibung des Parameters **X_DIM** finden Sie im Kapitel [Definition der Barcode Eigenschaften](#)

Um den **Swiss QRCode** zu drucken, muss der **X_DIM** Parameter den Wert 1046 haben. An diesem Wert erkennt das Programm, dass anstatt des "normalen" QR Codes der Swiss QR Code generiert werden soll. Der Swiss QR Code wird immer in der Größe 46 x 46 mm gedruckt und die Modulgröße wird dabei automatisch berechnet.

Y_DIM

Sollte nur in Ausnahmefällen definiert werden.

Zulässige Werte: 1 bis 100

Standardwert: X_DIM

XPOS

XPOS gilt nur für SAPscript. Der Parameter bestimmt die horizontale Position der Barcode Grafik relativ zum linken Fensterrand des Fensters, in dem der Barcode eingefügt wird. Der Wert darf nicht negativ sein.

Zulässige Werte: 0 bis Seitenbreite

Standardwert: 0

YPOS

YPOS gilt nur für SAPscript. Der Parameter bestimmt die vertikale Position der Barcode Grafik relativ zur Zeile, in der der Barcode eingefügt wird. Der Wert darf nicht negativ sein.

Zulässige Werte: 0 bis Seitelänge

Standardwert: 0

ECC_LEVEL Fehlerkorrekturlevel des Barcodes. Die ISO-Norm definiert 4 Fehlerkorrekturlevel: 'L', 'M', 'Q' und 'H'. Je nach Level werden in den Barcode spezielle Prüfziffern, sog. Reed Solomon Zahlen eingebaut, die es erlauben einen Barcode zu lesen, der teilweise zerstört wurde. Es gelten folgende Regeln für die Lesbarkeit eines QR-Codes:

L ca. 7% des Barcodes kann zerstört sein.

M ca 15% des Barcodes kann zerstört sein

H ca. 25% des Barcodes kann zerstört sein

Q ca. 30% des Barcodes kann zerstört sein.

Je höher der Fehlerkorrekturlevel, desto mehr RSS-Codewords werden berechnet und desto wenige Nutzdaten können in einem QRCode der gegebenen Versionsgröße kodiert werden.

Zulässige Werte: 'L' , 'M', 'Q', 'H'

Standardwert: 'L'

Autoheight Art der Platzreservierung für die Barcodegrafik in SAPscript Formularen. Wenn Autoheight auf 'Y' gesetzt wird, dann wird der ganze Raum links und rechts daneben für die Grafik beansprucht und es können keine Texte daneben platziert werden. Wird der Wert dagegen auf 'N' gesetzt, dann beansprucht die Grafik keinen Platz.

Zulässige Werte: 'Y' 'N'

Standardwert: 'N'

6.4 Drucken aus einem SAPscript Formular

6.4.1 Barcode Definition im SAPscript Formular

Das folgende Beispiel zeigt die Implementierung eines QR-Code Labes mit 5 verschiedenen Variablen in einem **SAPscript** Formular.:

```
/E ITEM_LINE
* <HL>RBarc/QR-Code
*
* <HL>2D Barcode QR-Code Model 2 generated on a SAP System
* <HN>A solution from Suchy MIPS www.suchymips.de
*
*
/* ***** RBarc/QR-Code 1 *****
/* ***** definition of encoding data and data types *****
/: DEFINE &QVAR1& = '(>'
/: DEFINE &DFLAG1& = 'T'
/: DEFINE &QVAR2& = 'RS'
/: DEFINE &DFLAG2& = 'F'
/: DEFINE &QVAR3& = '06'
/: DEFINE &DFLAG3& = 'T'
/: DEFINE &QVAR4& = 'GS'
/: DEFINE &DFLAG4& = 'F'
/: DEFINE &QVAR5& = '2L'
/: DEFINE &DFLAG5& = 'T'
/*
/* ***** RBarc/QR-Code 1 *****
/* ***** definition of optional barcode parameters *****
/: DEFINE &RES& = 150
/: DEFINE &QVERSION& = 0
/: DEFINE &ECC_LEVEL& = 'L'
/: DEFINE &X_DIM& = 4
/: DEFINE &XPOS& = '50.00'
/: DEFINE &YPOS& = '0.00'
/: DEFINE &AUTOHEIGHT& = 'N'
/*
/* ***** RBarc/QR-Code 1 *****
/* ***** performing barcode generation *****
/: PERFORM GEN_QRCODE_SS IN PROGRAM Z_SET_QRCODE
/: USING &QVAR1&
/: USING &DFLAG1&
/: USING &QVAR2&
/: USING &DFLAG2&
/: USING &QVAR3&
/: USING &DFLAG3&
/: USING &QVAR4&
/: USING &DFLAG4&
/: USING &QVAR5&
/: USING &DFLAG5&
/: USING &RES&
/: USING &QVERSION&
/: USING &ECC_LEVEL&
/: USING &X_DIM&
/: USING &XPOS&
/: USING &YPOS&
/: USING &AUTOHEIGHT&
/: CHANGING &QRNAME&
/: CHANGING &QRWIDTH&
/: CHANGING &RESULT&
/: ENDPERFORM
/*
/* ***** RBarc/QR-Code 1 *****
/* ***** including barcode bitmap *****
/: BITMAP &QRNAME& OBJECT GRAPHICS ID BMAP TYPE BMON XPOS &XPOS& MM
/*
/* ***** RBarc/QR-Code 1 *****
/* ***** deleting barcode bitmap *****
/: PERFORM DEL_QR_SS IN PROGRAM Z_SET_QRCODE
/: USING &QRNAME&
/: CHANGING &RESULT&
/: ENDPERFORM
/* ***** END of QR-Code 1 *****
```


Erläuterungen:

```
DEFINE &QVAR1& ='[(>Ä'  
DEFINE &DFLAG1& = 'T'
```

...

Definition der zu kodierenden Variablen. Es gibt keine Einschränkungen in der Anzahl der Variablen. Es können jederzeit weitere Variablen hinzugefügt werden. Die Namen der Variablen sind vorgeben und dürfen nicht geändert werden. Die zu verschlüsselnden Daten müssen in **QVARx** gespeichert werden, wobei **x** die Nummer der Variable in aufsteigender Reihenfolge, beginnend mit **1** ist.

```
DEFINE &RES& = 150
```

Definition der Grafikauflösung, wie unter 6.3 beschrieben. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

```
DEFINE &QVERSION& = 0
```

Definition der QR-Code Versionsnummer, wie unter 6.3 beschrieben. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

```
DEFINE &ECC_LEVEL& = 0
```

Definition Fehlerkorrekturlevel, wie unter 6.3 beschrieben. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

```
DEFINE &X_DIM& = 4
```

Definition der horizontalen Größe des kleinsten Moduls, wie unter 6.3 beschrieben. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden

```
DEFINE &XPOS& = '50.00'
```

Definition der horizontalen Position der Barcode Grafik relativ zum linken Rand des Fensters. Die Einheit ist Millimeter. Der Wert muss in Hochkommata angegeben werden und muss größer oder gleich Null sein. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

```
DEFINE &YPOS& = '0.00'
```

Definition der vertikalen Position der Barcode Grafik zur aktuellen Zeile, in der der Barcode ausgegeben wird. Die Einheit ist Millimeter. Der Wert muss in Hochkommata angegeben werden und muss größer oder gleich Null sein. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

```
DEFINE &AUTOHEIGHT& = 0
```

Definition der Raumbehandlung, den der Barcode einnimmt, wie unter 6.3 beschrieben. Wird der Parameter weggelassen, dann muss auch die entsprechende **USING** Variable aus dem **PERFORM GEN_QRCODE_SS** Befehl entfernt werden.

PERFORM GEN_QRCODE_SS IN PROGRAM Z_SET_QRCODE

```
USING &QVAR1&
USING &DFLAG1&
USING &QVAR2&
USING &DFLAG2&
USING &QVAR3&
USING &DFLAG3&
USING &QVAR4&
USING &DFLAG4&
USING &QVAR5&
USING &DFLAG5&
USING &RES&
USING &QVERSION&
USING &ECC_LEVEL&
USING &X_DIM&
USING &XPOS&
USING &YPOS&
USING &AUTOHEIGHT&
CHANGING &QRNAME&
CHANGING &QRWIDTH&
CHANGING &RESULT&
```

ENDPERFORM

Übergabe der zu kodierenden Variablen "QVAR...", der dazugehörigen Variablentypen "DFLAG..." und der optionalen Barcodeigenschaften "RES" (Resolution) "QVERSION" (Versionsnummer) "ECC_LEVEL" (Error Correction Level), "X_DIM" (Breite des kleinsten Moduls), "XPOS" (horizontale Position relativ zum linken Rand des Fensters), "YPOS" (vertikale Position relativ zur aktuellen Zeile) "AUTOHEIGHT" (Raumbehandlung) an die **Formroutine GEN_QRCODE_SS** im Programm **Z_SET_QRCODE**. Die Anzahl der Variablen ist nicht limitiert und kann vom Formularersteller beliebig erweitert werden - unter Berücksichtigung der oben beschriebenen Nummerierungsregel. Als Rückgabeparameter wird der Name des Barcodes geliefert (Parameter "QRNAME"), die Barcodebreite in Pixeln der aktuellen Auflösung ("QRWIDTH") und das Resultat ("RESULT"). Wenn keine Fehler auftauchen, dann gilt **RESULT = 'No errors'**.

IBITMAP &QRNAME& OBJECT GRAPHICS ID BMAP TYPE BMON XPOS &XPOS& MM

Der Barcode wird dynamisch in das Formular eingebunden. Mit dem Parameter "XPOS" kann der Barcode innerhalb des Fensters, in dem er eingefügt wird, horizontal positioniert werden.

PERFORM DEL_QR_SS IN PROGRAM Z_SET_QRCODE

```
USING &QRNAME&
CHANGING &RESULT&
```

ENDPERFORM

Nach dem Einbinden in das Formular wird der Barcode aus dem System gelöscht. Der Parameter **&RESULT&** übergibt das Ergebnis. Im positiven Fall enthält die Variable **RESULT** den Wert **"No errors"**. Es ist sehr wichtig den Barcode zu löschen. Im anderen Fall bleibt er im System erhalten und muss nachträglich gelöscht werden. Prüfen Sie nach erfolgten Tests mit der Transaktion **"SE78"** ob Grafiken, die mit **"ZQR"** beginnen vorhanden sind. Sie können die Grafiken nachträglich mit dem Programm **"ZDELBMP"**, das im Verzeichnis **"UTILITY"** zu finden ist, löschen.

Merke:

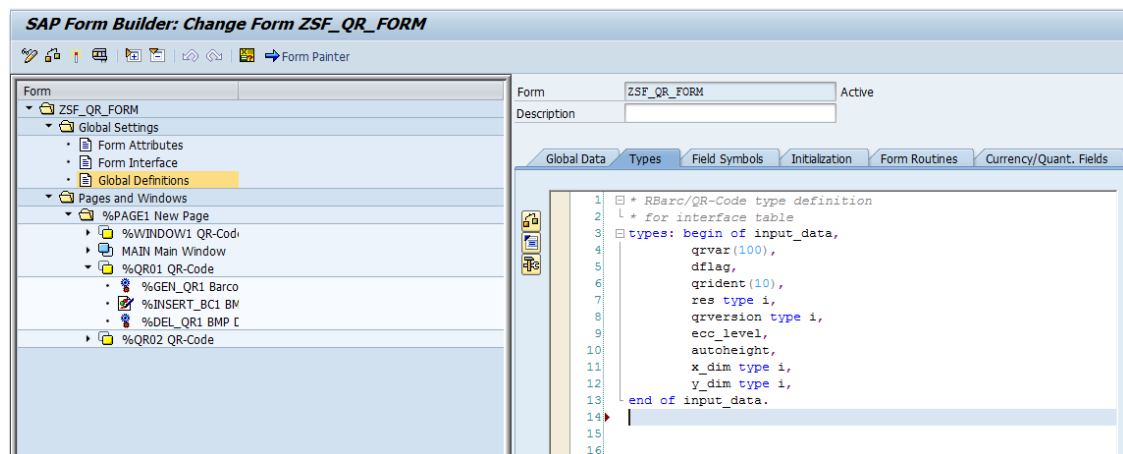
- Die Anzahl der zu kodierenden Variablen (**QVAR...**) ist nicht limitiert.
- Jeder Variable muss der dazugehörige Typ (**DFLAG...**) zugeordnet werden.
- Die zu kodierenden Variablen müssen lückenlos durchnummeriert werden, beginnend mit 1.
- Der zu Variable **QVARx** gehörende Typ **DFLAGx** muss die gleiche Nummer tragen.
- Variable **QVARx** und ihr Typ **DFLAGx** müssen hintereinander als **USING** Parameter übergeben werden.
- Es dürfen nur tatsächlich vorhanden oder explizit mit **DEFINE** deklarierte Variablen als **USING** Parameter übergeben werden. Wenn ein optionaler Parameter (z.B. **RES**) nicht definiert wird, dann darf er auch nicht als **USING** Parameter verwendet werden, sonst kommt es zu einem Fehler.
- Die Parameter "**RES**", "**QVERSION**", "**ECC_LEVEL**", "**X_DIM**", "**XPOS**", "**YPOS**" und **AUTOHEIGHT** sind optional. Werden Sie nicht verwendet, dann kommen Standardwerte automatisch zum Einsatz.
- **XPOS** und **YPOS** dürfen keine negativen Werte annehmen

6.5 .QR-Code Druck mit Smart Forms

6.5.1 Barcode Definition im Smart Forms Formular

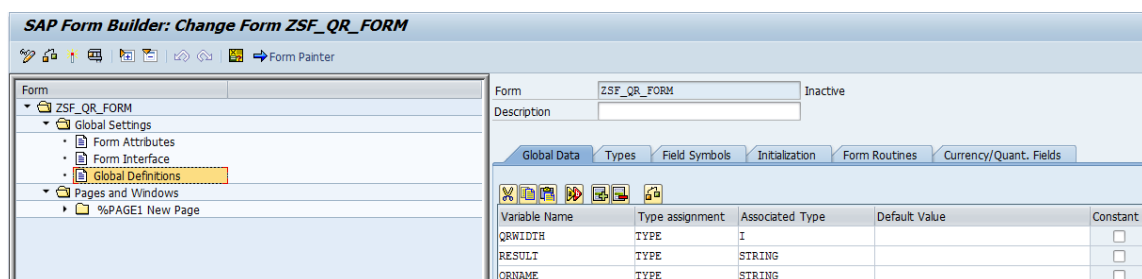
1. Definieren Sie im Reiter "Typen" aus dem Baum "Globale Definitionen" den type **input_data** wie folgt:

```
types: begin of input_data,  
      qrvar(100),  
      dflag,  
      qrident(10),  
      res type i,  
      qrversion type i,  
      ecc_level,  
      autoheight,  
      x_dim type i,  
      y_dim type i,  
end of input_data.
```

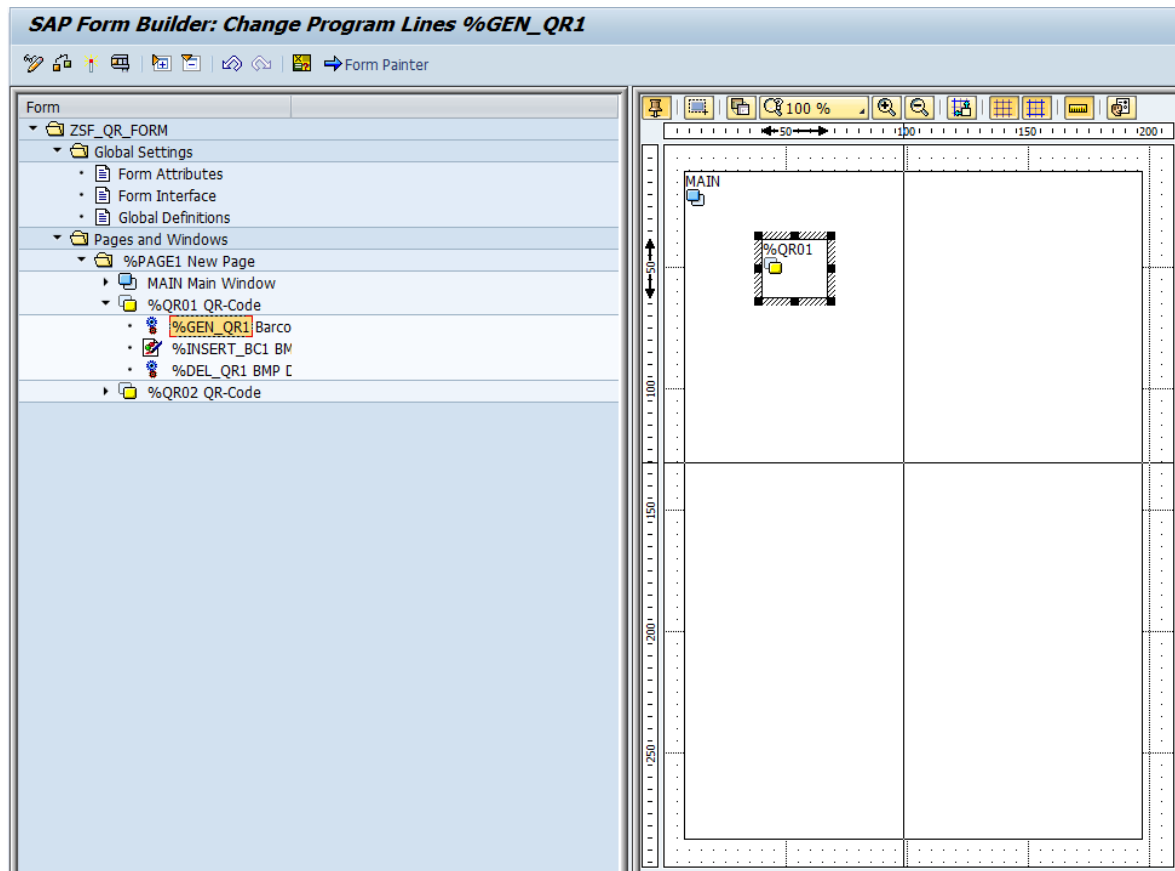


2. Definieren Sie im Reiter "Globale Daten" aus dem Baum "Globale Definitionen" die folgenden Variablen:

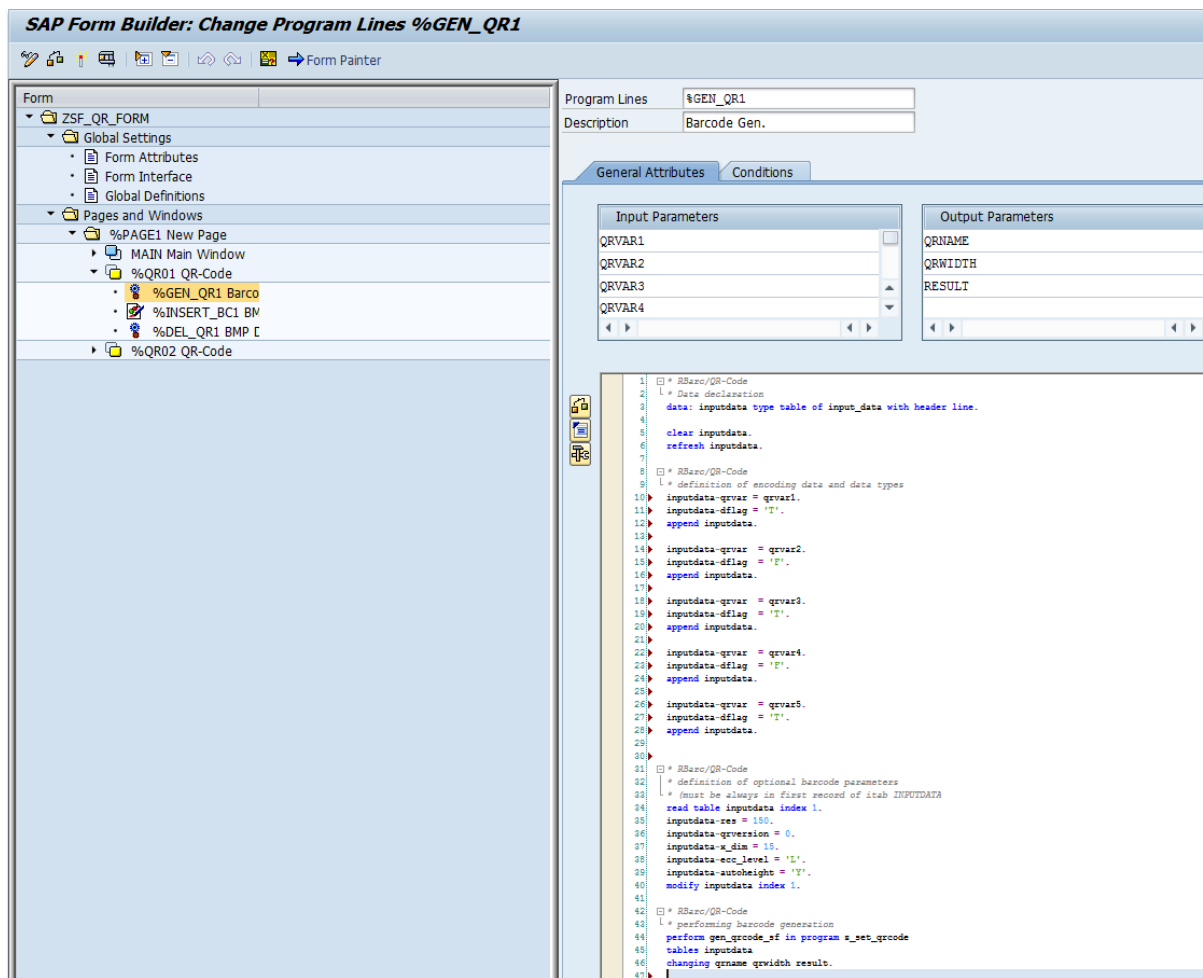
QRNAME	Type	String
QRWIDTH	Type	I
RESULT	Type	String



3. Erstellen Sie ein Fenster, das groß genug ist, um den Barcode aufzunehmen.
Im folgenden Beispiel ist das das Fenster "QR01 QR-Code".



- Erstellen Sie in dem gerade erzeugten Fenster einen Knoten vom Typ **"Programmzeilen"**.



- Geben Sie als **Input Parameter** alle zu kodierenden Variablen ein.
- Geben Sie als **Output Parameter** die Variable **QRNAME** und optional **QRWIDTH** und **RESULT** an (**QRWIDTH** und **RESULT** dienen der eigenen Verwendung).
- Deklariieren Sie die **interne Tabelle INPUTDATA**
`data: inputdata type table of input_data with header line.`
- Löschen Sie den Inhalt der Tabelle **INPUTDATA** (wichtig, wenn z.B. das Formular mehrere Seiten hat, auf denen die gleichen Programmknoten verwendet wird).
`clear inputdata.`
`refresh inputdata.`

- Schreiben Sie die jeweilige Variable in das Tabellenfeld **QRVAR** und den dazugehörigen Datentypen in das Tabellenfeld **DFLAG** und hängen den jeweiligen Datensatz an die Tabelle **INPUTDATA** an

```
inputdata-qrvar = qrvar1.  
inputdata-dflag = 'T'.  
append inputdata.
```

... usw. für jede zu kodierende Variable.

Die Anzahl der kodierbaren Variablen ist nicht limitiert. Die Variablen mit ihrem dazugehörigen Typen müssen lückenlos hintereinander in die Tabelle geschrieben werden. Jede Variable bildet also einen Datensatz (=Record) in der Tabelle.

- Nachdem alle zu kodierenden Variablen in die Tabelle eingetragen wurden, lesen Sie den ersten Datensatz der Tabelle und tragen dort die gewünschten Werte für **"RES"** (Resolution) **"QRVERSION"** (Versionsnummer) **ECC_LEVEL** (Error Correction Level), **"X_DIM"** (Breite des kleinsten Moduls) ein

```
read table inputdata index 1.  
inputdata-res = 150.  
inputdata-qrversion = 0.  
inputdata-x_dim = 15.  
inputdata-ecc_level = 'L'.  
modify inputdata index 1.
```

Dieser Teil ist optional. Falls er nicht verwendet wird, dann werden für die Parameter **RES**, **QRVERSION**, **ECC_LEVEL** und **X_DIM** und Standardwerte verwendet.

- Übergeben Sie die Tabelle **INPUTDATA** an die Formroutine **GEN_QRCODE_SF** im Programm **Z_SET_QRCODE**

```
perform gen_qrcode_sf in program z_set_qrcode  
tables inputdata  
changing qrcode qrwidth result.
```

- Als Rückgabeparameter wird der Name des Barcodes geliefert (Parameter **"QRNAME"**), die Barcodebreite in Pixeln der aktuellen Auflösung (**"QRWIDTH"**) und das Resultat (**"RESULT"**). Wenn keine Fehler auftreten, dann gilt **RESULT = 'No errors'**.

.

5. Erstellen Sie im Fenster für den QR-Code Barcode einen zweiten Knoten vom Typ "**Grafik**" und füllen Sie die Felder **Name**, **Objekt** und **ID**, wie im folgenden Bild dargestellt:

SAP Form Builder: Change Graphic %INSERT_BC1

Form: ZSF_QR_FORM

- Global Settings
 - Form Attributes
 - Form Interface
 - Global Definitions
- Pages and Windows
 - %PAGE1 New Page
 - MAIN Main Window
 - %QR01 QR-Code
 - %GEN_QR1 Barco
 - %INSERT_BC1 BMP**
 - %DEL_QR1 BMP I
 - %QR02 QR-Code

Graphic: %INSERT_BC1
Description: BMP Instert

General Attributes | Output Options | Conditions

Name: &QRNAME&
Object: GRAPHICS
ID: BMAP

☒ Black and White Bitmap Image (BMON)
☐ Color Bitmap Image (BCOL)
☐ Determine Dynamically (BMON, BCOL)

Technical Attributes
Resolution: DPI

Name = &QRNAME&

Objekt = GRAPHICS

ID = BMAP

Auflösung = Leer lassen

6. Erstellen Sie im Fenster für den QR-Code Barcode einen dritten Knoten vom Typ "**Programmzeilen**" und fügen Sie dort folgendes Statement ein:

```
perform del_qr_sf in program z_set_qrcode  
using qname  
changing result.
```

Geben Sie als **Input Parameter** alle **QRNAME** ein.

Geben Sie als **Output Parameter** **RESULT** ein (optional, falls Sie es verwenden möchten).

SAP Form Builder: Change Program Lines %DEL_QR1

Form: ZSF_QR_FORM

- Global Settings
 - Form Attributes
 - Form Interface
 - Global Definitions
- Pages and Windows
 - %PAGE1 New Page
 - MAIN Main Window
 - %QR01 QR-Code
 - %GEN_QR1 Barco
 - %INSERT_BC1 BMP
 - %DEL_QR1 BMP I**
 - %QR02 QR-Code
 - %GEN_QR2 Barco
 - %INSERT_BMP2 E
 - %DEL_QR2 BMP I

Program Lines: %DEL_QR1
Description: BMP Delete

General Attributes | Conditions

Input Parameters
QRNAME

Output Parameters
RESULT

1 * RBarco/QR-Code
2 * Deleting barcode from the system
3
4 perform del_qr_sf in program z_set_qrcode
5 using qname
6 changing result.

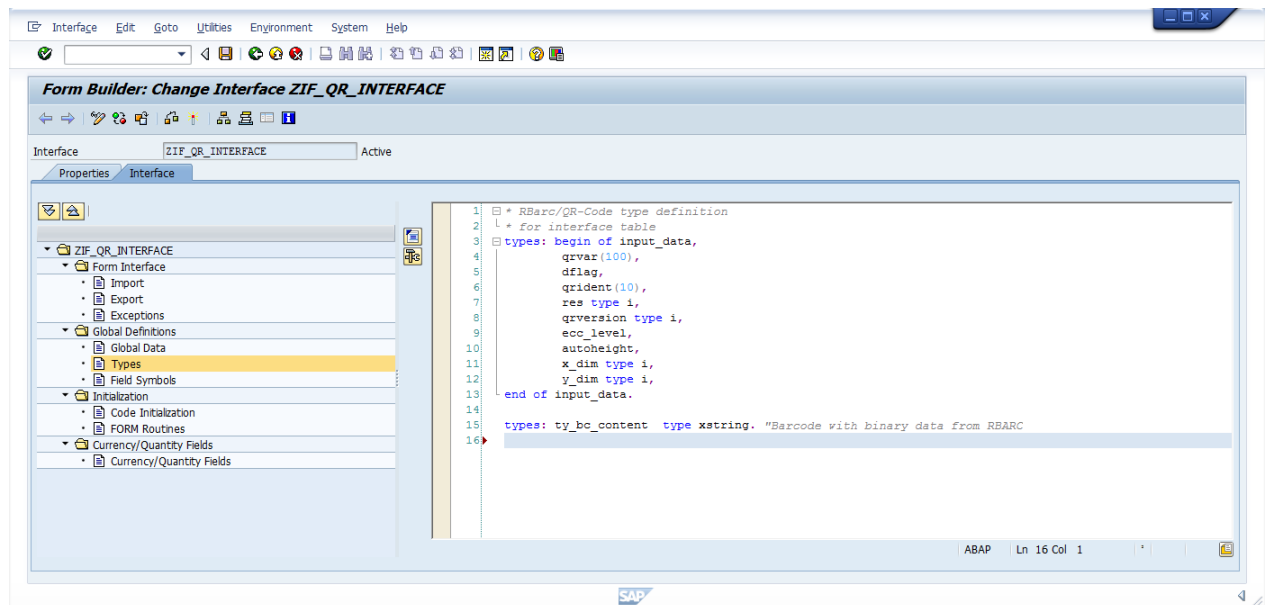
6.6 QR-Code Druck mit Interactive Forms

6.6.1 Erweiterung der Interactive Forms Schnittstelle

1. Legen Sie im Knoten „Globale Definition→ Typen“ die Typen **input_data** und **ty_bc_content** wie folgt an:

```
types: begin of input_data,  
        qrvar(100),  
        dflag,  
        qrident(10),  
        res type i,  
        qrversion type i,  
        ecc_level,  
        autoheight,  
        x_dim type i,  
        y_dim type i,  
end of input_data.
```

```
types: ty_bc_content type xstring.
```

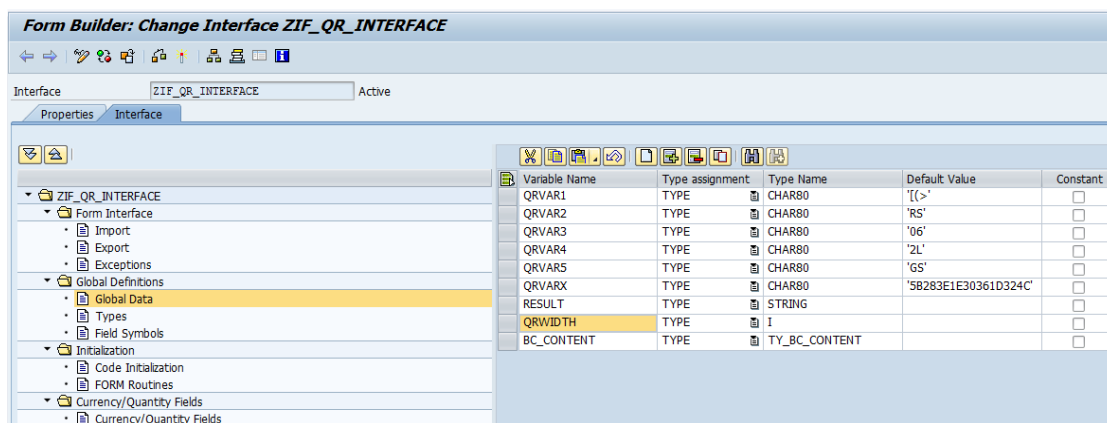


2. Definieren Sie im Knoten „**Globale Definitionen→ Globale Daten**“ folgende globale Variablen.

```

QRWIDTH TYPE      I
RESULT  TYPE      STRING
BC_CONTENT          TYPE    TY_BC_CONTENT

```



Bemerkung:

im obigen Beispiel wurden auch zu kodierende Variablen **QRVAR1** bis **QRVAR5** als globale Variablen in der Schnittstelle angelegt. In der Regel werden das jedoch Variablen aus der Workflow der Datenverarbeitung sein. Sie müssen dann mit geeigneten Mitteln dafür sorgen, dass die benötigten Variablen in der Schnittstelle auch sichtbar sind. In der Regel ist es dann nicht notwendig, solche Variablen als globale Variablen zu definieren, wie das z.B. der Fall bei dem Feld **MATNR** aus der Tabelle **MARD** wäre.

Für mehrere Barcodes in einem Formular können mehrere Globale variablen vom Type **TY_BC_CONTENT** angelegt werden.

3. Fügen Sie folgendes Coding im Knoten „**Initialisierung→Code Initialisierung**“ ein (hier ein Beispiel mit 5 verschiedenen Variablen):

** RBarc/QRCode*

** Data declaration*

data: inputdata **type table of** input_data **with header line.**

clear inputdata.

refresh inputdata.

**RBarc/QRCode*

**definition of encoding data and data types*

inputdata-qrvar = qrvar1.

inputdata-dflag = 'T'.

append inputdata.

inputdata-qrvar = qrvar2.

inputdata-dflag = 'F'.

append inputdata.

inputdata-qrvar = qrvar3.

inputdata-dflag = 'T'.

append inputdata.

inputdata-qrvar = qrvar4.

inputdata-dflag = 'F'.

append inputdata.

inputdata-qrvar = qrvar5.

inputdata-dflag = 'T'.

append inputdata.

** RBarc/QRCode*

** definition of optional barcode parameters*

** (must be always in first record of itab INPUTDATA*

read table inputdata **index 1.**

inputdata-res = 150.

inputdata-qrversion = 0.

inputdata-x_dim = 10.

inputdata-ecc_level = 'L'.

modify inputdata **index 1.**

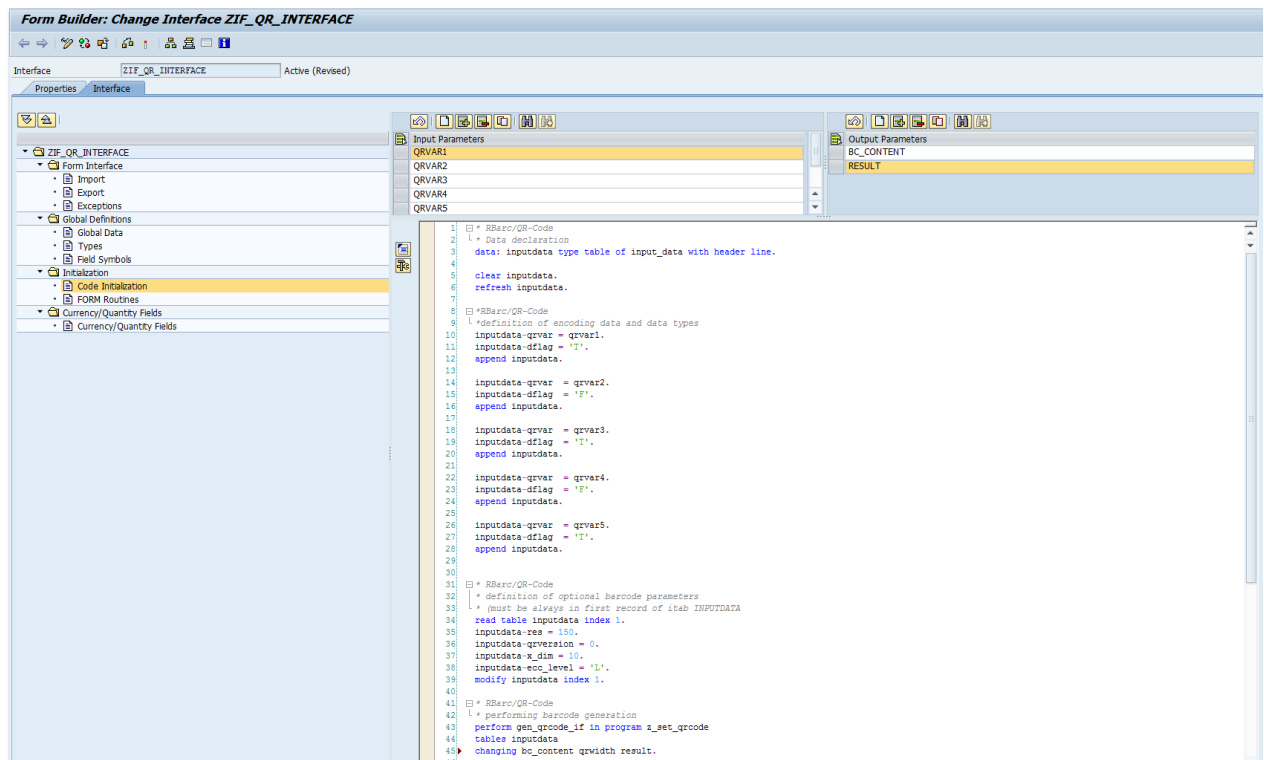
** RBarc/QRCode*

** performing barcode generation*

perform gen_qrcode_if **in program** z_set_qrcode

tables inputdata

changing bc_content qrwidth result.



Erläuterungen:

- Geben Sie als **Input Parameter** alle zu kodierenden Variablen und die Variablen ein.
- Geben Sie als **Output Parameter** die Variable **BC_CONTENT** und **RESULT** an
- Deklarieren Sie die interne Tabelle **INPUTDATA**

`data: inputdata type table of input_data with header line.`

- Löschen Sie den Inhalt der Tabelle **INPUTDATA** (wichtig, wenn z.B. das Formular mehrere Seiten hat auf denen die gleichen Programmknotten verwendet wird).

`clear inputdata.`
`refresh inputdata.`

- Schreiben Sie die jeweilige Variable in das Tabellenfeld **QRVAR** und den dazugehörigen Datentypen in das Tabellenfeld **DFLAG** und hängen den jeweiligen Datensatz an die Tabelle **INPUTDATA** an

`inputdata-qrvar = qrvar1.`
`inputdata-dflag = 'T'.`
`append inputdata.`

... usw. für jede zu kodierende Variable.

Die Anzahl der kodierbaren Variablen ist nicht limitiert. Die Variablen mit ihrem dazugehörigen Typen müssen lückenlos hintereinander in die Tabelle geschrieben werden. Jede Variable bildet also einen Datensatz (=Record) in der Tabelle.

- Nachdem alle zu kodierenden Variablen in die Tabelle eingetragen wurden, lesen Sie den ersten Datensatz der Tabelle und tragen dort die gewünschten Werte für **RES** (Resolution), **QRVERSION**, **ECC_LEVEL** und **X_DIM** (Größe des kleinsten Moduls) ein

```
read table inputdata index 1.
inputdata-res = 150.
inputdata-qrversion = 0.
inputdata-ecc_level = 'L'.
inputdata-x_dim = 4.
modify inputdata index 1.
```

Dieser Teil ist optional. Falls er nicht verwendet wird, dann werden für die Parameter **RES**, **QRVERSION**, **ECC_LEVEL** und **X_DIM** Standardwerte verwendet.

- Übergeben Sie die Tabelle **INPUTDATA** an die Formroutine **GEN_QRCODE_IF** im Programm **Z_SET_QRCODE**

```
perform gen_qrcode_if in program z_set_qrcode
tables inputdata
changing bc_content qrwidth result.
```

Als Rückgabe Parameter werden folgende Variablen übergeben: **bc_content** (das ist die Barcode Graphic), **qrwidth** (die Barcodebreite in Pixel der aktuellen Auflösung) und **result** (das Ergebnis). Ist die Barcodegenerierung fehlerfrei verlaufen, dann ist **result = 'No errors'**.

Wichtig:

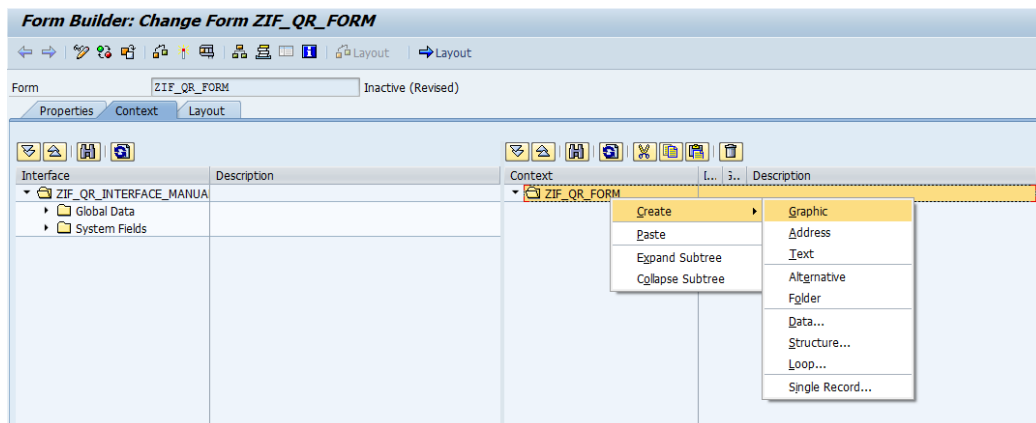
Bitte beachten Sie, dass die verwendeten Variablen im Knoten „Initialisierung“ bekannt gemacht werden müssen, d.h. Sie müssen immer als Eingabeparameter neben den anderen Parametern eingetragen werden.

6.6.2 Erweiterung des Interactive Forms Formulars

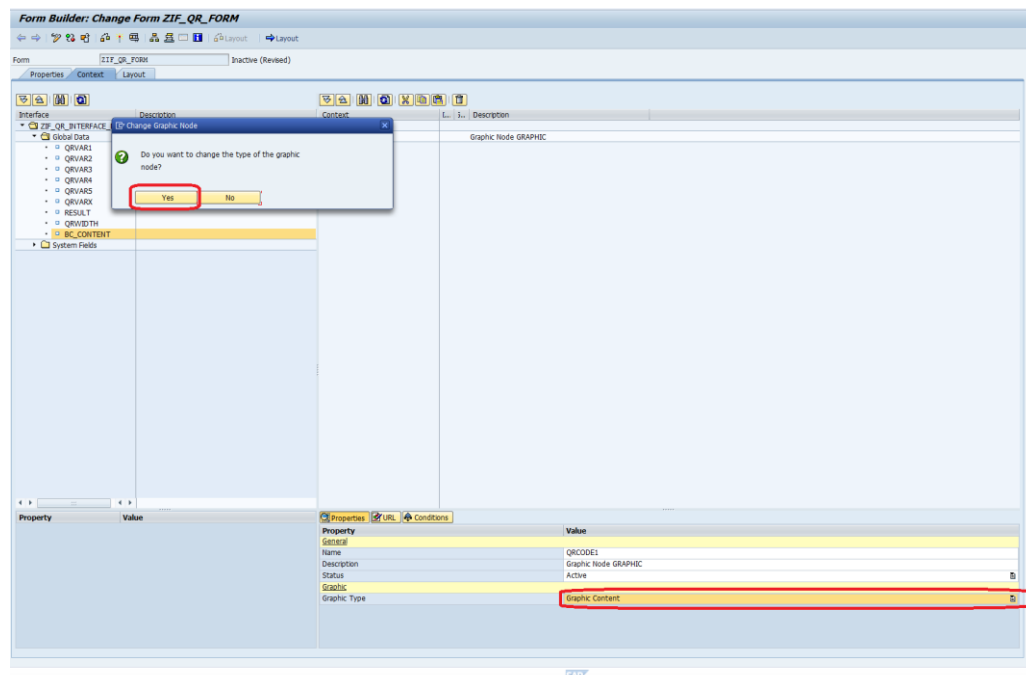
Im **Interactive Forms** Formular wird der mit **RBarc/QRCode** erzeugte Barcode als Bildobjekt dargestellt. Zunächst muss daher der Kontext um die notwendigen Elemente für einen Barcode erweitert werden.

1. Markieren Sie im Formular (Transaktion **SFP**) im Kontextfenster (rechts) den Formularnamen, klicken auf die rechte Maustaste und wählen aus dem Kontextmenü **Anlegen→Grafik** um eine Grafik anzulegen.

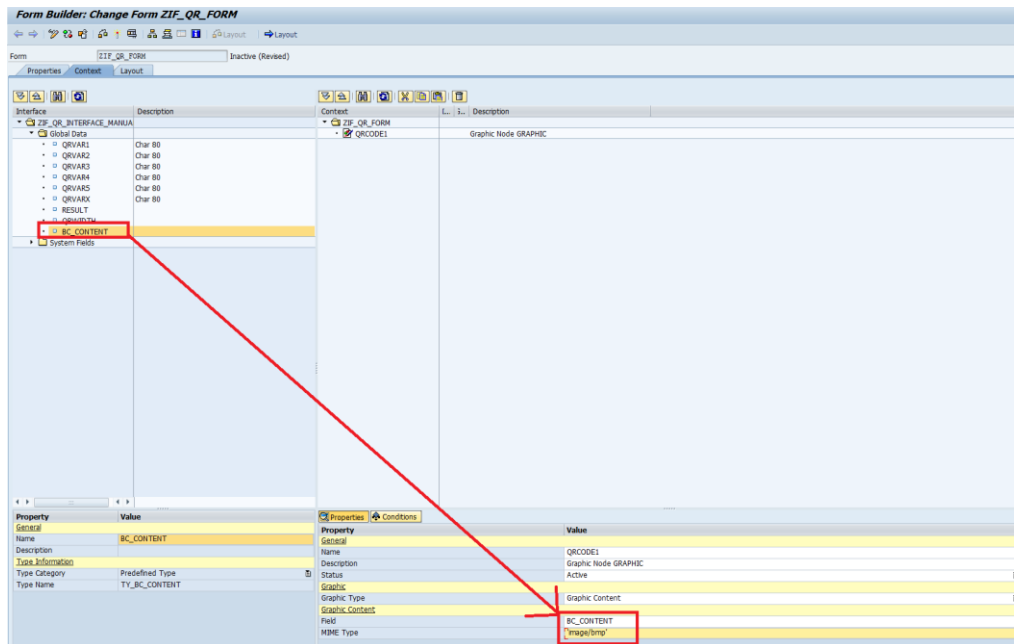
Geben Sie der Grafik einen Namen und eine Beschreibung. Im Beispiel lautet der Name der Grafik **QRCODE1**.



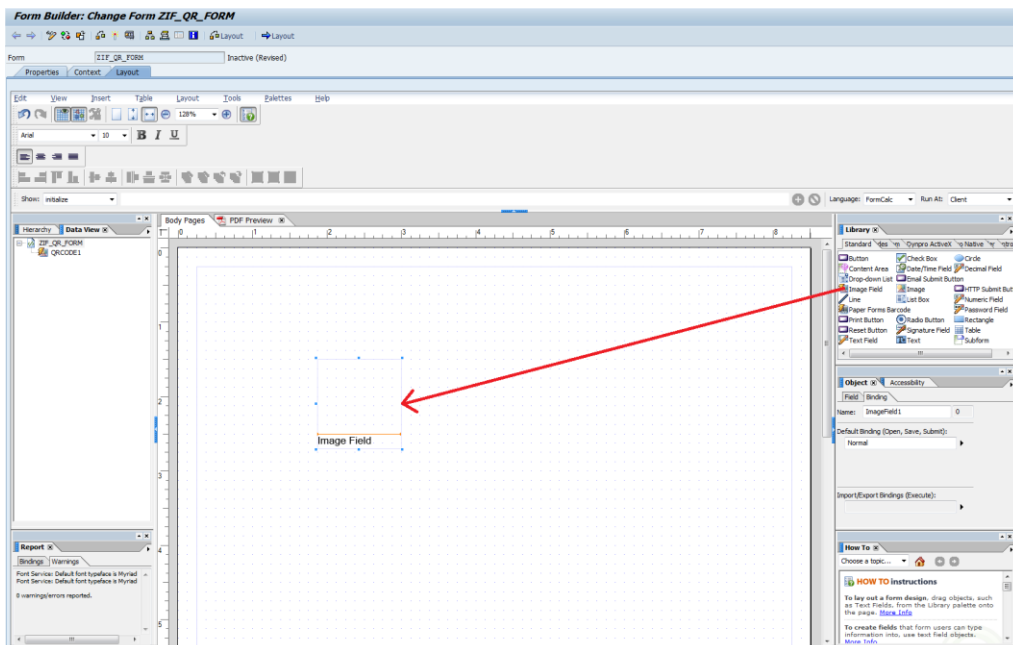
2. Wechseln Sie jetzt den Grafiktyp auf „**Grafikinhalt**“ und bestätigen Sie das Popup mit „**Ja**“.



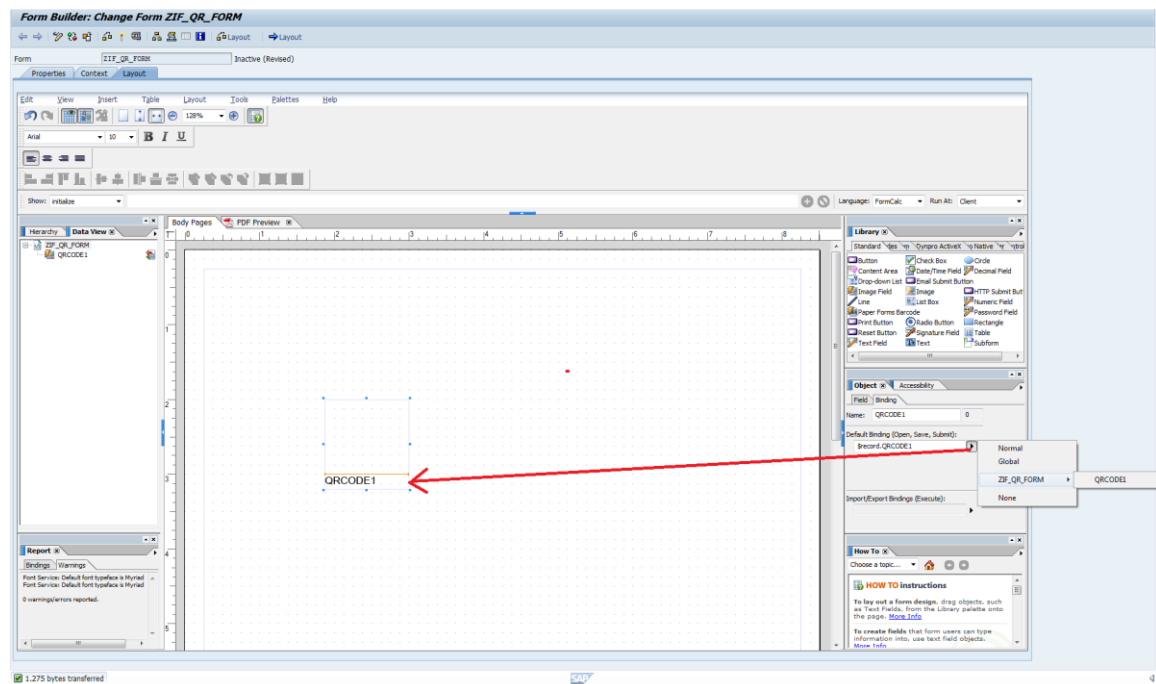
- Verknüpfen Sie jetzt die Grafik **QRCODE1** mit dem Datenfeld, der den Inhalt des Barcodes enthält, indem Sie mit gedrückter linken Maustaste aus der Schnittstelle das Feld **BC_CONTENT** in das passende Feld auf der rechten Seite ziehen. Geben Sie als **MIME-Typ** 'image/bmp' an.



- Wechseln Sie nun in die Layoutansicht des Form Builders und legen Sie ein Element vom Typ „Bildfeld“ an.



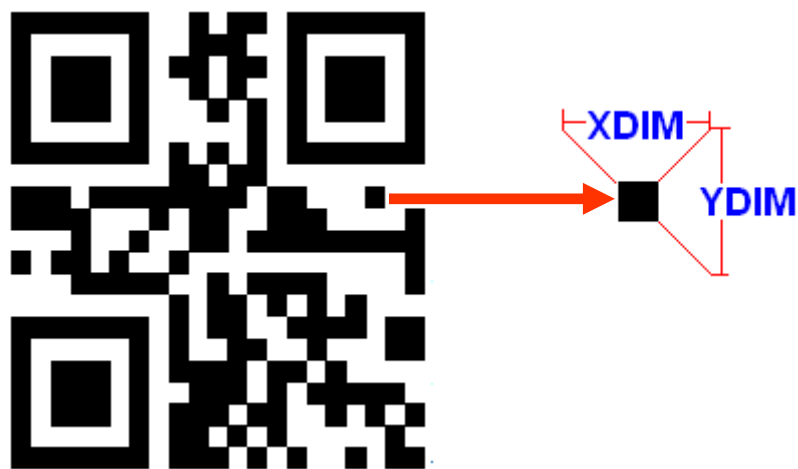
5. Geben Sie dem Bildfeld als Datenbindung den im Kontext definierten Grafikknoten mit (im Beispiel **QRCODE1**)..



Fertig!

Durch die soeben erzeugte Datenbindung wird bei der Verarbeitung des Interactive Forms an der ausgewählten Stelle der von RBarc/QRCode ON-THE-FLY erzeugte QR-Barcode Barcode eingefügt.

6.7 Definition der Barcode Eigenschaften.



Ein QR-Barcode besteht aus weißen und schwarzen Modulen, deren Größe variabel ist.

Bemerkung:

Wird Swiss QR Code gedruckt, dann hat der X_DIM Parameter eine andere Bedeutung.

Um den Swiss QR Code zu drucken, muss der X_DIM Parameter den Wert **1046** haben. An diesem Wert erkennt das Programm, dass anstatt des "normalen" QR Codes der Swiss QR Code generiert werden soll. Der Swiss QR Code wird immer in der Größe 46 x 46 mm gedruckt und die Modulgröße wird dabei automatisch berechnet.

Folgende Versionen des QR-Codes können definiert werden:









QRVersion	Anzahl Module	Datencodewords
0	auto	Je nach Datenmenge
1	21x21	26
2	25x25	44
3	29x29	70
4	33x33	100
5	37x37	134
6	41x41	172
7	45x45	196
8	49x49	242
9	53x53	192
10	57x57	346
11	61x61	404
12	65x65	466
13	69x69	632
14	73x73	581
15	77x77	655
16	81x81	733
17	85x85	815
18	89x89	901
19	93x93	991
20	97x97	1085
21	101x101	1156
22	105x105	1258
23	109x109	1364
24	113x113	1474
25	117x117	1588
26	121x121	1706
27	125x125	1828
28	129x129	1921
29	133x133	2051
30	137x137	2185
31	141x141	2323
32	145x145	2465
33	149x149	2611
34	153x153	2761
35	157x157	2876
36	161x161	3034
37	165x165	3196
38	169x169	3362
39	173x173	3532
40	177x177	3706






In der obigen Tabelle wurde die Anzahl der Datencodewords angegeben, die je nach gewählter Versionsnummer (Parameter **qrversion**) kodiert werden kann. Bitte bedenken Sie, dass dieser Wert nur ein Anhaltspunkt für die Menge der Daten ist, die wirklich kodiert werden kann. In manchen Fällen werden die zu kodierenden Daten so komprimiert, dass mehr davon kodiert werden können, als die jeweils angegebene Anzahl von Codewords. In manchen, ungünstigen Fällen kann es jedoch passieren, dass die Anzahl der Daten, die kodiert werden können, geringer ist, als die angegebene Anzahl von Datencodewords für eine bestimmte Barcode Größe.





Bemerkung:




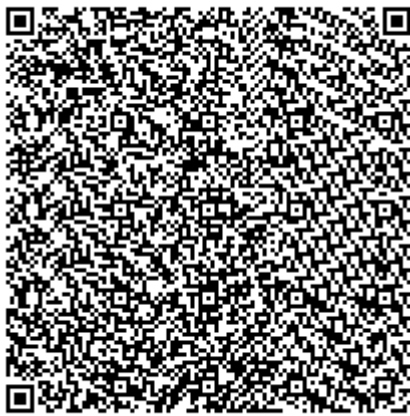
Bei Angabe des Wertes **QRVERSION = 0**, wird die Versionsnummer automatisch berechnet und zwar so, dass die Daten in einer kleinstmöglichen Matrix kodiert werden. Ein wichtiger Aspekt dieses Verfahrens ist, dass bei Seriendruckern die Barcodes verschiedene Größen haben können, je nach Anzahl der Daten, die zum Kodieren anfallen.

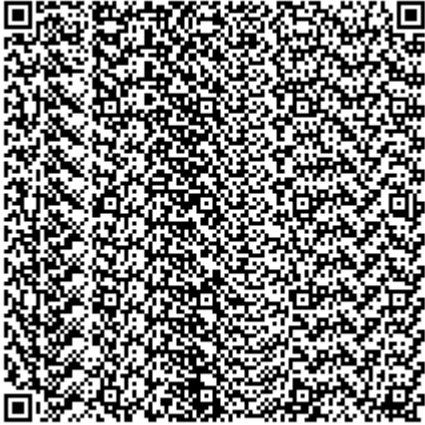
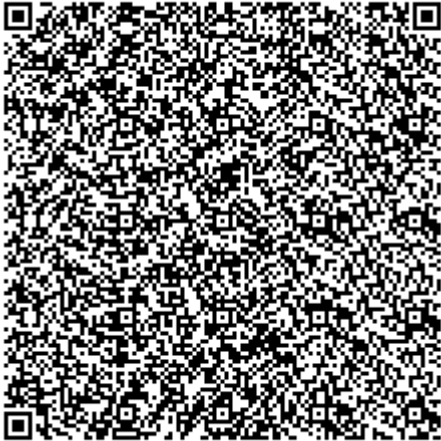
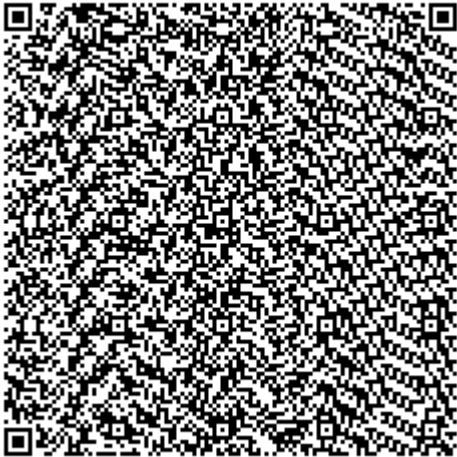
Demgegenüber kann die Größe mit dem Parameter **QRVERSION** fest definiert werden, sodass alle Barcodes in gleicher Größe erscheinen. In diesem Fall droht jedoch die Gefahr, dass bei zu großer Datenmenge der Barcode mit vorgegebener Größe nicht erzeugt werden kann und eine Fehlermeldung erscheint. Sie sollten also in einem solchen Fall den Barcode entsprechend überdimensionieren und in Ihren Tests nicht nur Ziffern (beste Datenkompression), sondern wechselnd Klein-, Großbuchstaben, Sonderzeichen und eine kleine Anzahl von Ziffern verwenden.

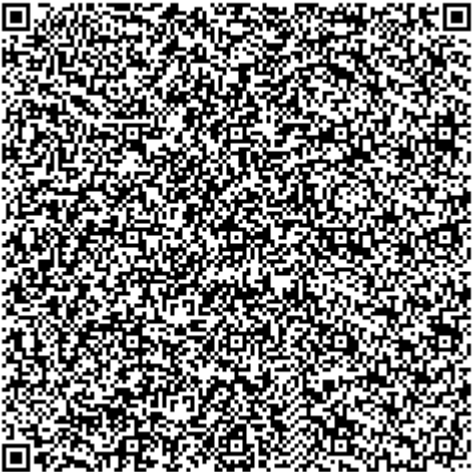
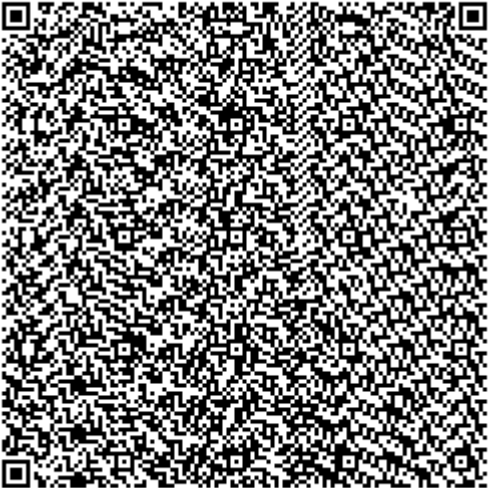
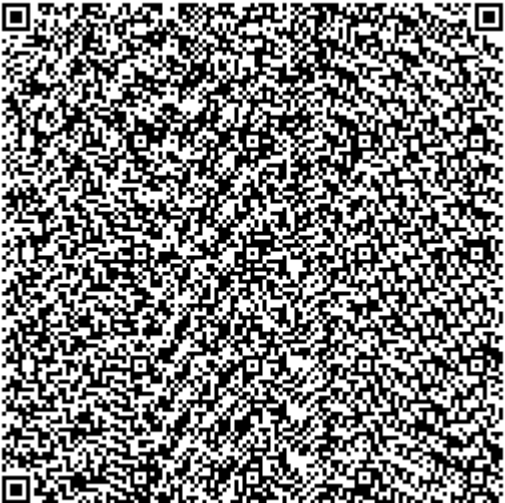
Version	Module	Barcode Beispiel
1	21x21	
2	25x25	
3	29x29	
4	33x33	
5	37x37	
6	41x41	
7	45x45	
8	49x49	

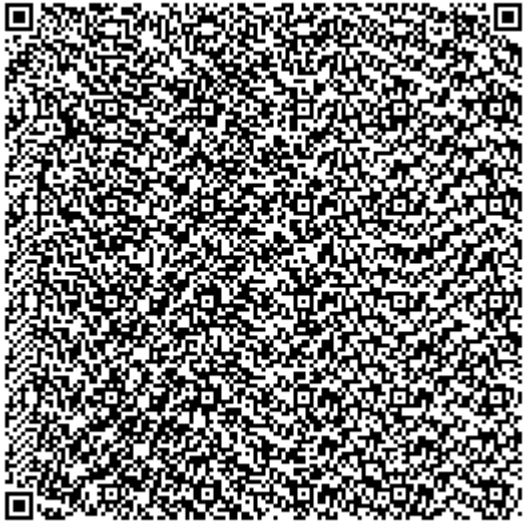
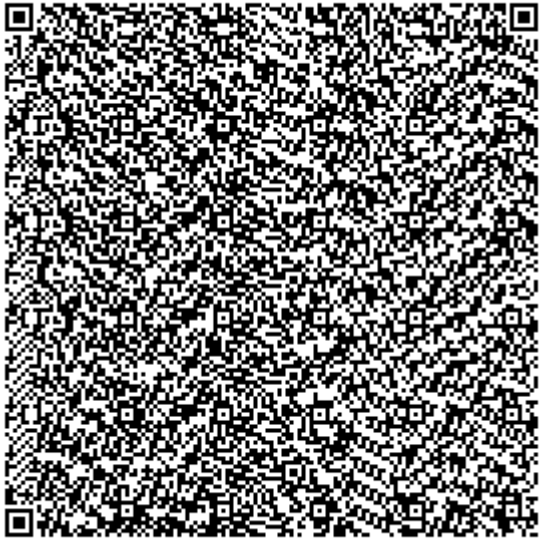
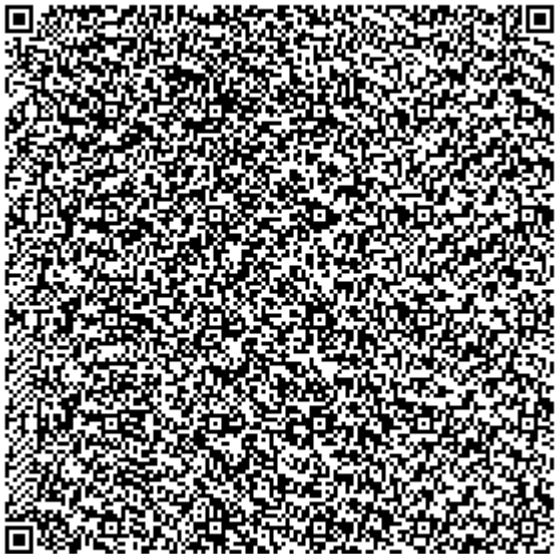
9	53x53	
10	57x57	
11	61x61	
12	65x65	
13	69x69	

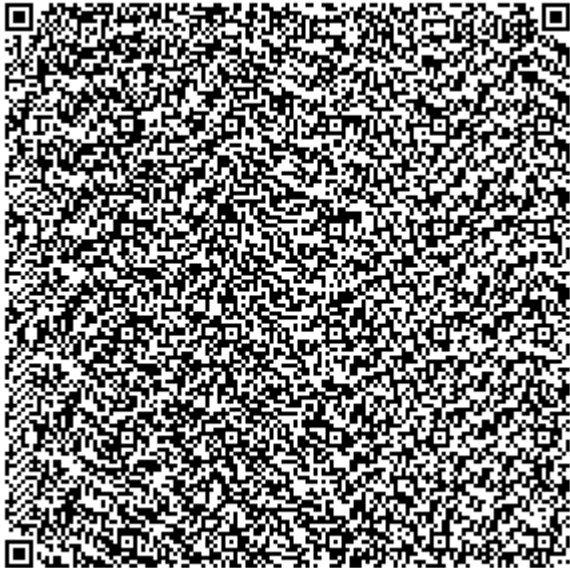
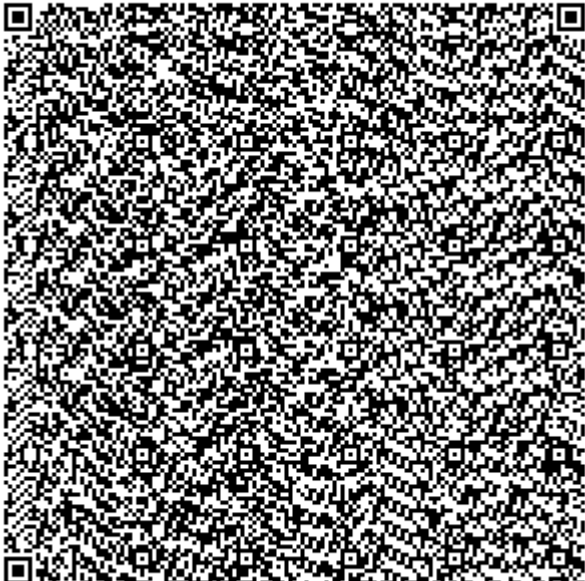
14	73x73	
15	77x77	
16	81x81	
17	85x85	

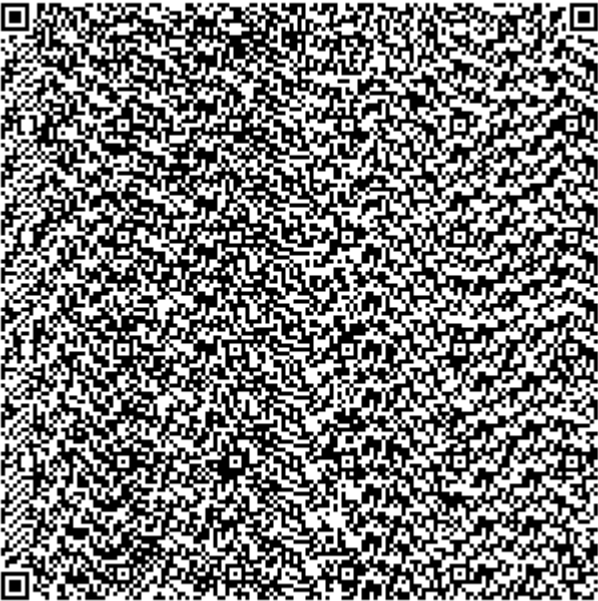
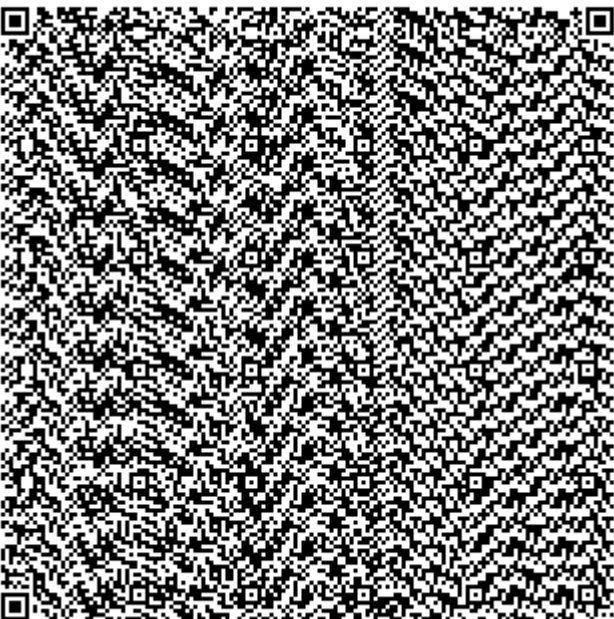
18	89x89	
19	93x93	
20	97x97	
21	101x101	

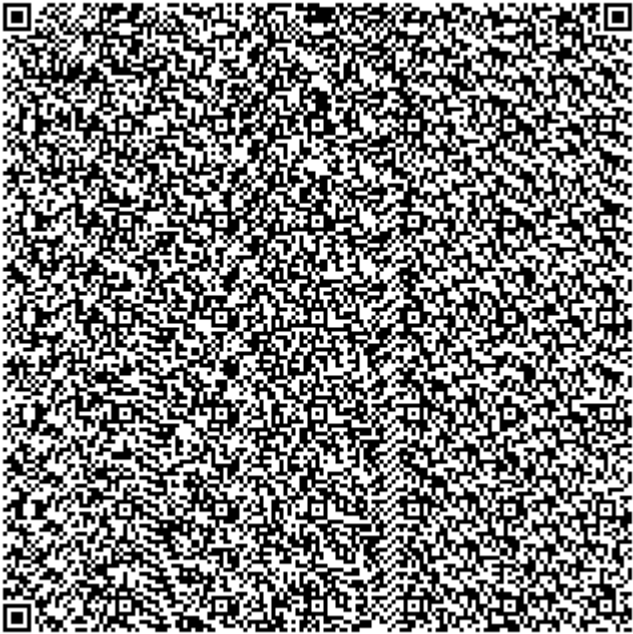
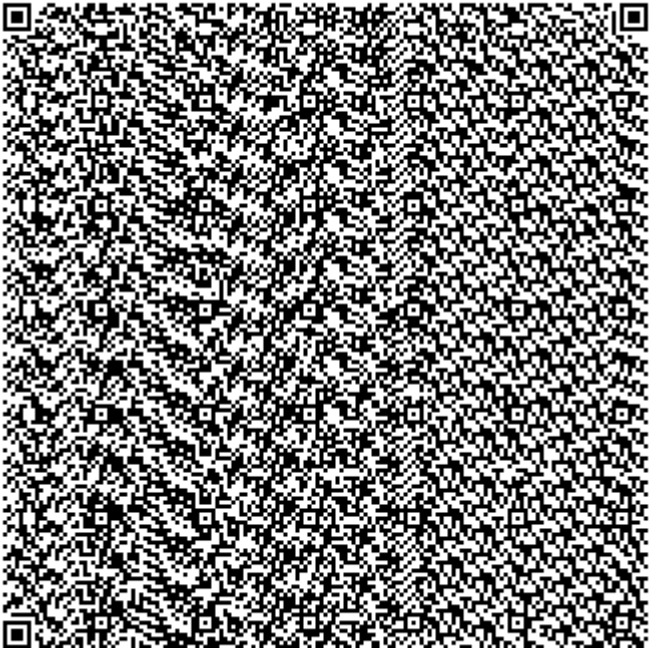
22	104x104	
23	109x109	
24	113x113	

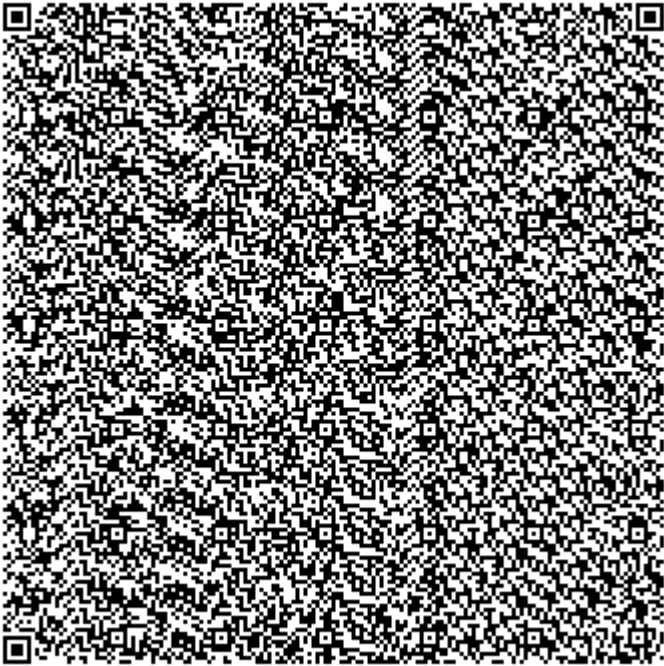
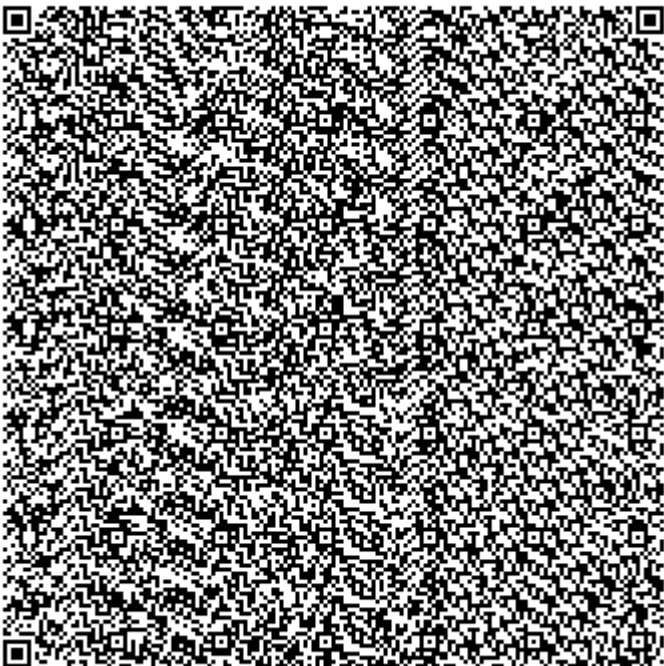
25	117x117	
26	121x121	
27	125x125	

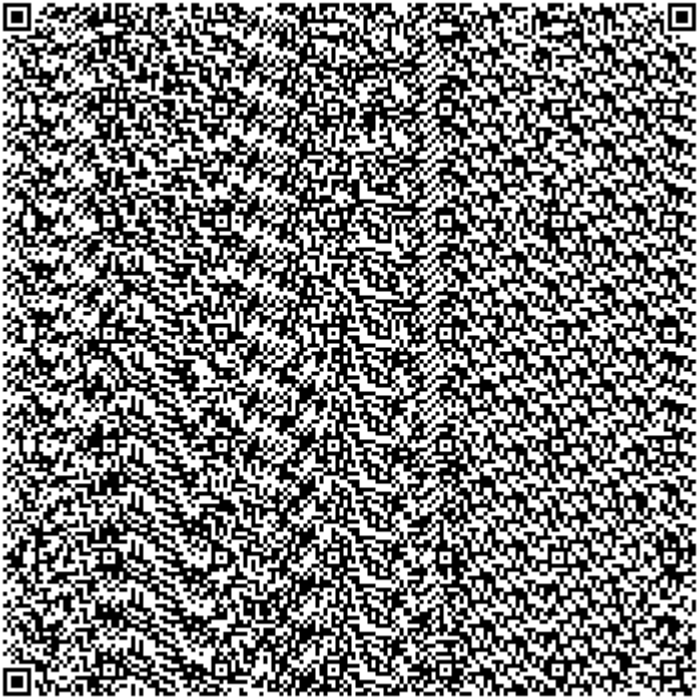
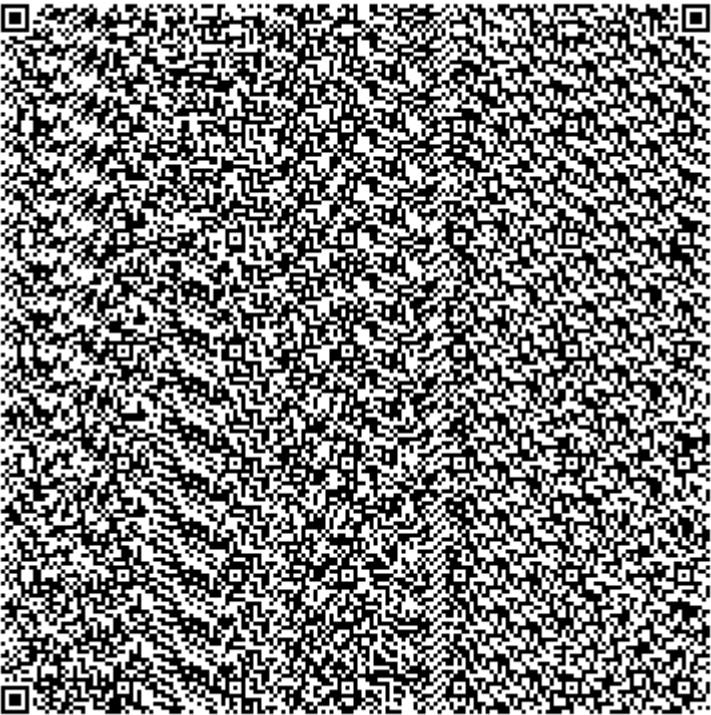
28	129x129	
29	133x133	
30	.137x137	

31	141x141	
32	145x145	

33	149x149	
34	153x153	

35	157x157	
36	161x161	

37	165x165	
38	169x169	

39	173x173	
40	177x177	

7 Beispiel für eine Aufgabenstellung in einem SAPscript Formular.

7.1 Aufgabe

Es sollen folgende Felder kodiert werden:

VBELN aus der Tabelle **LIKP**,
VBELN aus der Tabelle **VBRK** und
ADDRNUMBER aus der Tabelle **ADRT**.

Die Variablen sollen voneinander mit einem **GS** (Group Separator) als Feldseparator getrennt werden. Der Barcode soll **50 Mm** vom linken Rand des Fensters gedruckt werden. Das Modul soll **1 Mm** breit und **1 Mm** hoch sein. Die Versionsnummer soll automatisch berechnet werden

Lösung:

Definieren Sie im **SAPscript** Formular die 3 Variablen, deren Typen (dflag) und die Variablen mit den Barcode Eigenschaften (**RES**, **QRVERSION**, **ECC_LEVEL**, **AUTOHEIGHT** und **X_DIM**). Übergeben Sie alle definierten Variablen an die Formroutine **GEN_QRCODE_SS** im Report **Z_SET_QRCODE**. Anschließend lassen Sie die von **RBarc/QRCode** erzeugte Grafik dynamisch in das Formular einbinden und vom System löschen.

```
/: DEFINE &QVAR1& = &LIKP-VBELN&
/: DEFINE &DFLAG1& = 'T'
/: DEFINE &QVAR2& = 'GS'
/: DEFINE &DFLAG2& = 'F'
/: DEFINE &QVAR3& = &VBRK-VBELN&
/: DEFINE &DFLAG3& = 'T'
/: DEFINE &QVAR4& = 'GS'
/: DEFINE &DFLAG4& = 'F'
/: DEFINE &QVAR5& = &ADRT-ADDRNUMBER&
/: DEFINE &DFLAG5& = 'T'
/: DEFINE &RES& = 150
/: DEFINE &QRVERSION& = 0
/: DEFINE &ECC_LEVEL& = 'L'
/: DEFINE &X_DIM& = 6
/: DEFINE &AUTOHEIGHT& = 'N'
```

(bitte Seite wechseln)

```
/: PERFORM GEN_QRCODE_SS IN PROGRAM Z_SET_QRCODE
```

```
/: USING &QVAR1&
```

```
/: USING &DFLAG1&
```

```
/: USING &QVAR2&
```

```
/: USING &DFLAG2&
```

```
/: USING &QVAR3&
```

```
/: USING &DFLAG3&
```

```
/: USING &QVAR4&
```

```
/: USING &DFLAG4&
```

```
/: USING &QVAR5&
```

```
/: USING &DFLAG5&
```

```
/: USING &RES&
```

```
/: USING &QVERSION&
```

```
/: USING &ECC_LEVEL&
```

```
/: USING &X_DIM&
```

```
/: CHANGING &QRNAME&
```

```
/: CHANGING &QRWIDTH&
```

```
/: CHANGING &RESULT&
```

```
/: ENDPERFORM
```

```
/: BITMAP &QRNAME& OBJECT GRAPHICS ID BMAP TYPE BMON XPOS &XPOS& MM
```

```
/: PERFORM DEL_QR_SS IN PROGRAM Z_SET_QRCODE
```

```
/: USING &QRNAME&
```

```
/: CHANGING RESULT
```

```
/: ENDPERFORM
```

In der oben dargestellten Implementierung werden zunächst alle Variablen Typen zugeordnet (z.B. wird **&LIKP-VBELN&** als Text definiert). Dabei wird der Separator "GS" als Funktionscode mit dem Typenbezeichnung **dflag = 'F'** kodiert.

Im nächsten Schritt werden Barcode Eigenschaften mit **RES**, **QVERSION**, **ECC_LEVEL**, **X_DIM** und **AUTOHEIDHT** definiert.

Alle definierten Variablen und Parameter werden in einem Schritt an die Formroutine **GEN_QRCODE_SS** im Programm **Z_SET_QRCODE** übergeben.

Die von **RBarc/QRCode** erzeugte Grafik mit dem Namen **&QRNAME&** wird in das Formular mit der Anweisung **BITMAP...** eingebunden.

Zum Schluss wird die Grafik durch den Aufruf der Formroutine **DEL_QR_SS** im Programm **Z_SET_QRCODE** aus dem System gelöscht.

8 Beispiel für Aufgabenstellung in einem Smart Forms Formular.

8.1 Aufgabe.

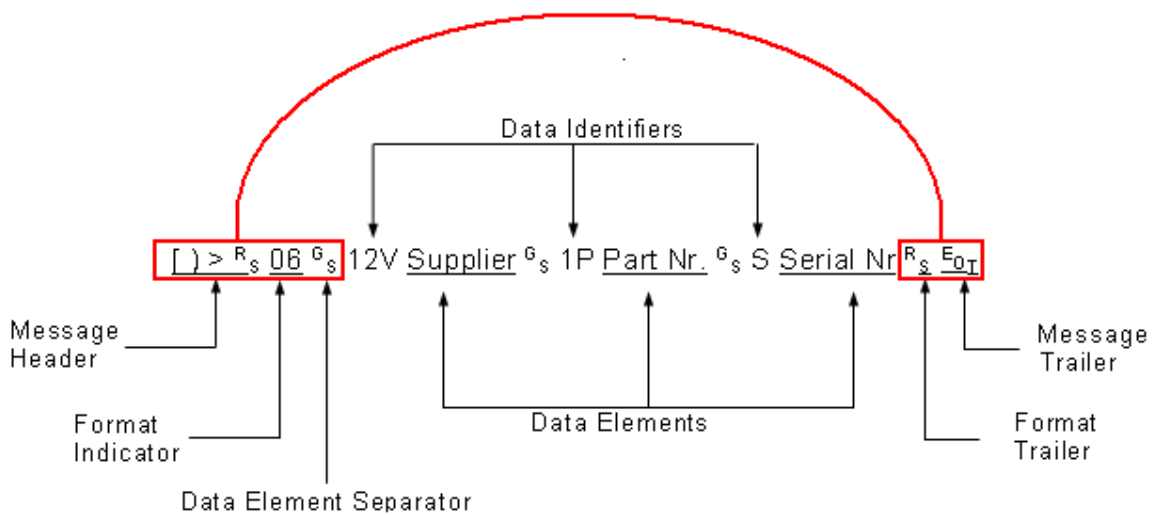
Es sollen 3 Variablen kodiert werden:

die Hersteller-Nummer "**SuplierNo**",

die Artikel-Nummer "**PartNo**"

die Seriennummer eines Artikels "**SerNo**".

Die Variablen sollen in ein sog. **Message Envelop** "verpackt" werden (siehe Abbildung). Das Matrixmodul soll **1 Mm** breit und **1 Mm** hoch sein. Die Matrixgröße soll automatisch berechnet werden.



Lösung:

1. Als erstes erzeugen Sie das Barcode Fenster mit den 3 dazugehörigen Knoten, wie im Kapitel **Drucken von Barcodes aus Smart Forms** beschrieben. Weitere Definitionen müssen nur im ersten Programmzeilen-Knoten durchgeführt werden.
2. Der Inhalt der Knoten 2 und 3 bleibt immer unverändert (wie unter **Drucken von Barcodes aus Smart Forms** beschrieben).

(bitte Seite wechseln)

3. Erstellen Sie im ersten Barcode-Programmknotten folgenden Code:.

Datendeklaration

data: inputdata **type table of** input_data **with header line.**

clear inputdata.

refresh inputdata.

Kodierung von '['(>' alsText

inputdata-qrvar = '['(>.

inputddata-dflag = 'T'.

append inputdata.

Kodierung von 'RS' als Funktionscode

inputdata-qrvar = 'RS'.

inputddata-dflag = 'F'.

append inputdata.

Kodierung von '06' alsText

inputdata-qrvar = '06'.

inputdata-dflag = 'T'.

append inputdata.

Kodierung von 'GS' als Funktionscode

inputdata-qrvar = 'GS'.

inputdata-dflag = 'F'.

append inputdata.

Kodierung von '12V' + suplierno alsText

concatenate '12V' suplierno **into** inputdata-qrvar.

inputdata-dflag = 'T'.

append inputdata.

Kodierung von 'GS' als Funktionscode

inputdata-qrvar = 'GS'.

inputddata-dflag = 'F'.

append inputdata.

Kodierung von '1P' + partno alsText

concatenate '1P' partno **into** qrvar.

inputdata-dflag = 'T'.

append inputdata.

Kodierung von 'GS' als Funktionscode

inputdata-qrvar = 'GS'.

inputddata-dflag = 'F'.

append inputdata.

Kodierung von 'S' + serialno alsText

concatenate 'S' serialno **into** qrvar.

inputdata-dflag = 'T'.

append inputdata.

Kodierung von 'RS' als Funktionscode

inputdata-qrvar = 'RS'.

inputdata-dflag = 'F'.

append inputdata.

Kodierung von 'EOT' als Funktionscode

inputdata-qrvar = 'EOT'.

inputdata-dflag = 'F'.

append inputdata.

Definition der Barcode Eigenschaften

```
read table inputdata index 1.  
inputdata-res = 150.  
inputdata-qrversion = 0.  
inputdata-ecc_level = 'L'.  
inputdata-x_dim = 4.  
modify inputdata index 1.  
  
perform gen_qrcode_sf in program z_set_qrcode  
tables inputdata  
changing qrcode qrcode result.
```

In der oben dargestellten Implementierung werden zunächst alle Variablen Typen zugeordnet (z.B. wird **12+SUPLIERNO** als Text definiert). Dabei wird der Separator **"GS"** als Funktionscode mit dem Typenbezeichnung dflag = **'F'** kodiert.

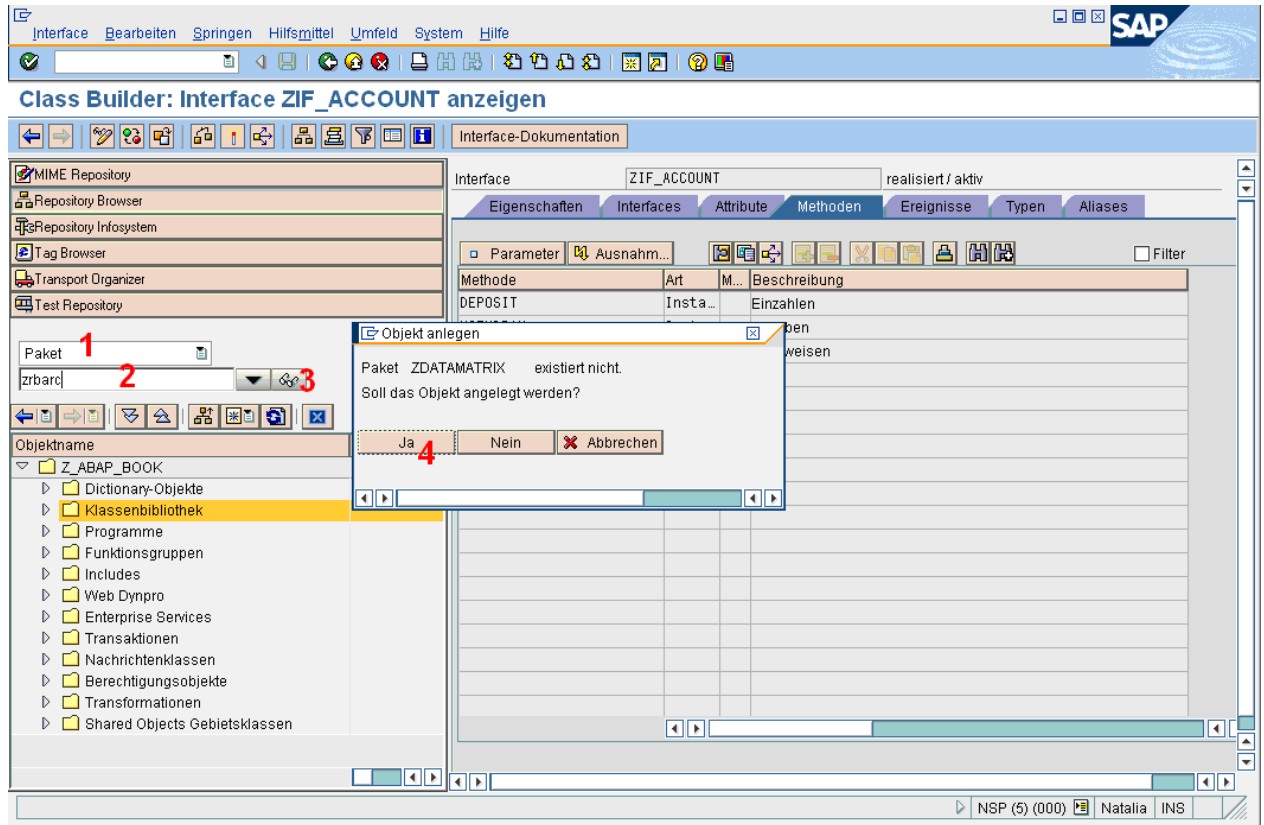
Im nächsten Schritt werden Barcode Eigenschaften mit **RES**, **QRVERSION**, **ECC_LEVEL** und **X_DIM** definiert und in den ersten Satz der Schnittstellentabelle **INPUTDATA** geschrieben.

Die Schnittstellentabelle **INPUTDATA** wird an die Formroutine **gen_qrcode_sf** im Programm **z_set_qrcode** übergeben.

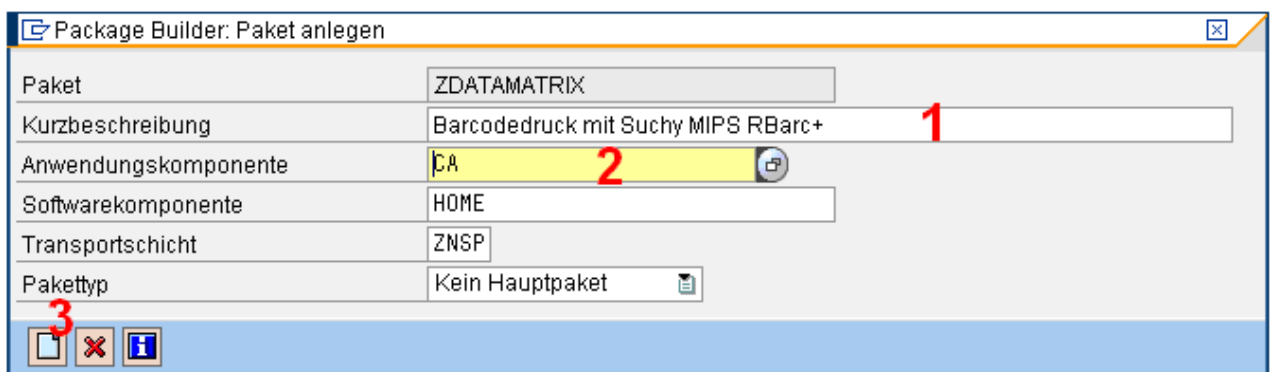
Als Rückgabeparameter wird der Name des Barcodes **QRNAME** geliefert. Der Barcode wird in der weiteren Formularverarbeitung dynamisch eingebunden.

9 Anhang 1: Neues Paket (Entwickungsklasse) erstellen.

1. Starten Sie die Transaktion **SE80 (Objekt Navigator)**.
2. Es erscheint folgender Dialog:

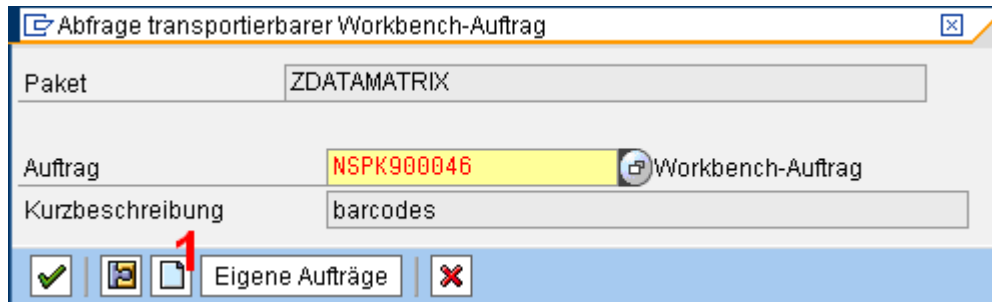



3. Wählen Sie als darzustellendes Objekt „Paket“ (1) (auf älteren Systemen „Entwickungsklasse“).
4. Geben Sie unter dem Objekttypfeld den Namen **ZQRCODE** ein (2).
5. Klicken Sie auf das Brillensymbol für Anzeige (3).
6. Es erscheint die Meldung, dass das Paket nicht vorhanden ist und Sie werden gefragt, ob Sie es anlegen möchten. Klicken Sie auf **Ja** (4).
7. Es erscheint folgender Dialog:

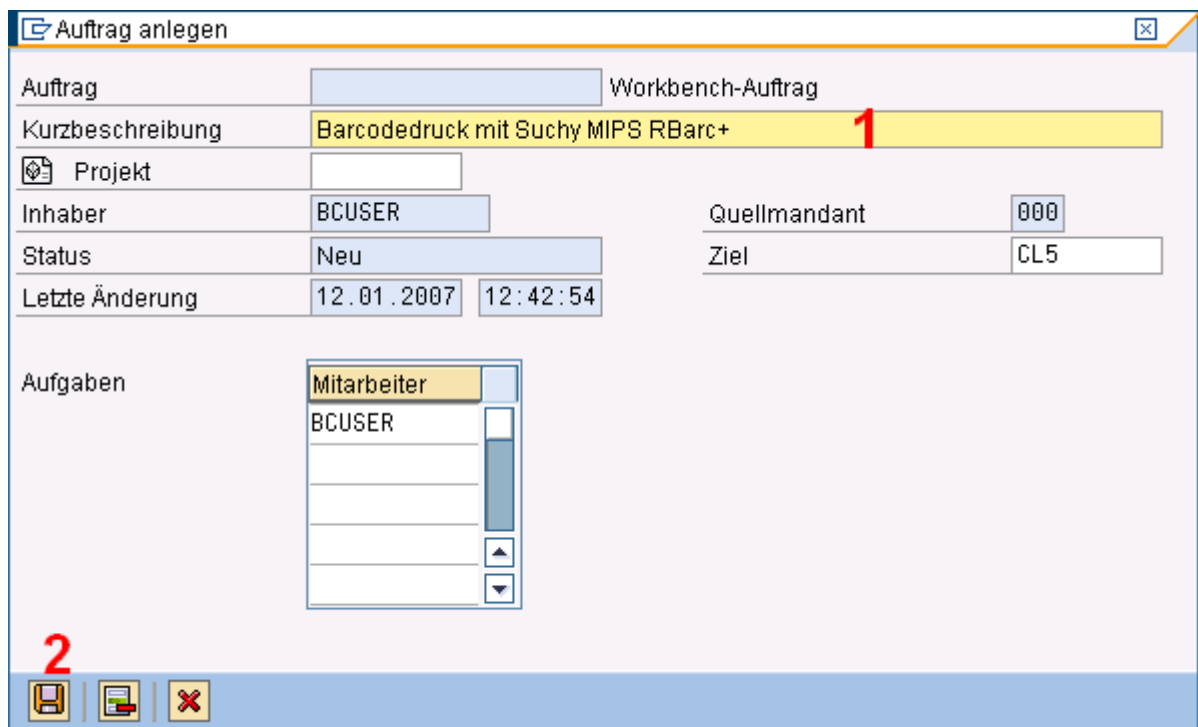


8. Fügen Sie eine Kurzbeschreibung ein (1).
9. Wählen Sie „CA“ als Anwendungskomponente (2).
10. Klicken Sie auf das Bestätigungssymbol (3) um das Paket anzulegen.

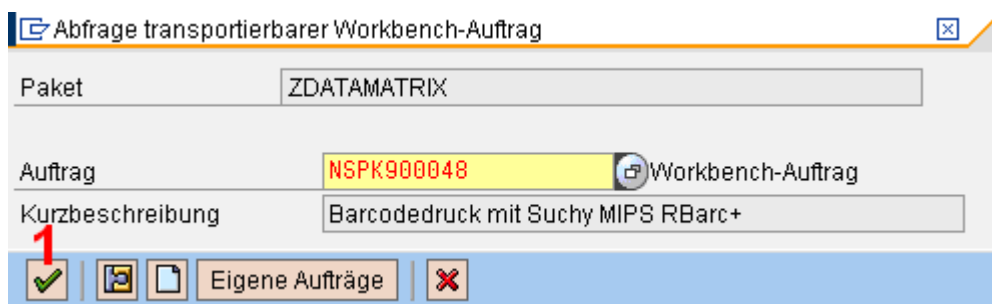
11. Es erscheint die Abfrage für transportierbaren Workbench-Auftrag.



12. Falls Sie für diese Installation noch keinen neuen Auftrag angelegt haben, legen Sie jetzt einen an. Sonst springen Sie weiter zum Pt. 16.
13. Klicken Sie auf das Symbol  um einen neuen Transportauftrag zu erstellen.



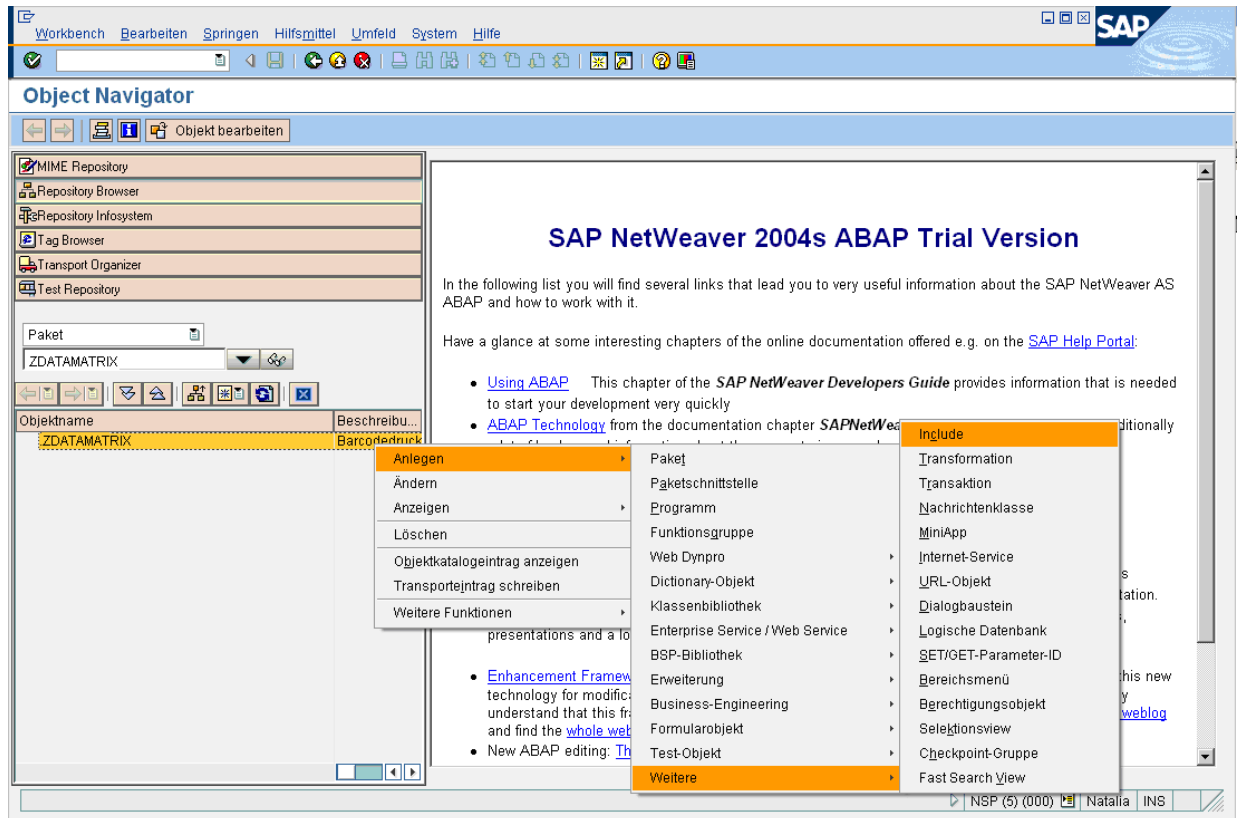
14. Tragen Sie eine Kurzbeschreibung für den Auftrag ein (1).
15. Klicken Sie auf das Diskettensymbol (2) um den Eintrag zu speichern.
16. Die Auftragsnummer erscheint in dem Feld „Auftrag“



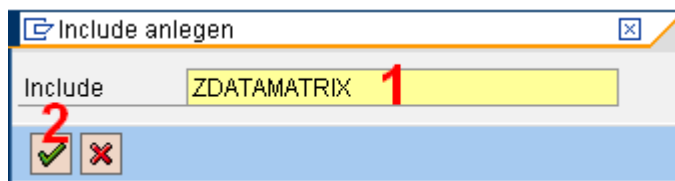
17. Klicken Sie auf das grüne Häkchen (1) um den Auftrag zu speichern.

10 Anhang 2: Ein Include in der ABAP Workbench anlegen.

1. Falls noch nicht geschehen, starten Sie den **Objekt Navigator** (Transaktion **SE80**) und wählen Sie das angelegte Paket (Entwickungsklasse) **ZQRCODE**.



2. Markieren Sie den Objektnamen **ZQRCODE** und klicken Sie auf die rechte Maustaste.
3. Wählen Sie aus dem Kontextmenü **Anlegen/Weitere/Include** (Bei älteren Systemen erscheint **Include** gleich an zweiter Stelle).
4. Geben Sie in dem folgenden Dialog einen Namen für das zu erstellende Include (**1**) und klicken auf das grüne Häkchen (**2**).



5. Klicken Sie in dem folgenden Dialog auf „Sichern“ (1) um das Include zu erstellen.

ABAP: Programmeigenschaften

Titel: ZDATAMATRIX

Originalsprache: DE Deutsch

Erstellt: 12.01.2007 BCUSER

Letzte Änderung:

Status: neu(überarbeitet)

Attribute

Typ: Include-Programm

Status:

Anwendung:

Berechtigungsgruppe:

☐ Editorsperre

1 Sichern

6. Achten Sie darauf, dass das Include dem neu angelegten Paket (Entwickungsklasse) **ZQRCODE** hinzugefügt wird (1) und klicken Sie auf das Diskettensymbol (2) um das Include zu sichern.

Objektkatalogeintrag anlegen

Objekt: R3TR PROG ZRBARC_2006X

Attribute

Paket: ZDATAMATRIX 1

Verantwortlicher: CUST

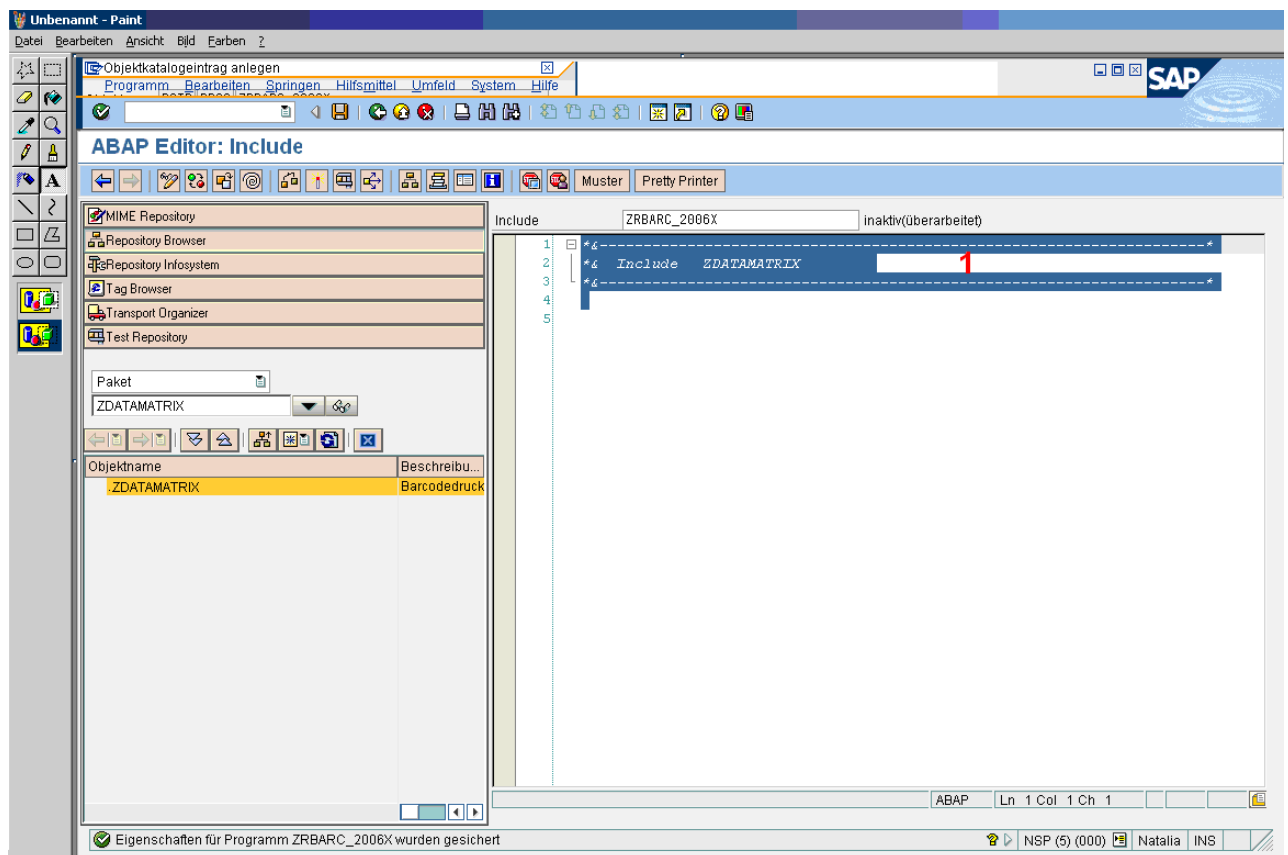
Originalsystem: NSP

Originalsprache: DE Deutsch

2 Lokales Objekt Sperrübersicht

7. Es erscheint der Dialog für das Anlegen eines Transportauftrages. Dieser sollte zu diesem Zeitpunkt schon vorhanden sein, da er beim Anlegen des Pakets (Entwicklungsklasse) **ZQRCODE** erstellt wurde. Die Auftragsnummer sollte also diesmal die gleiche sein, wie beim Anlegen des Paketes (**1**).

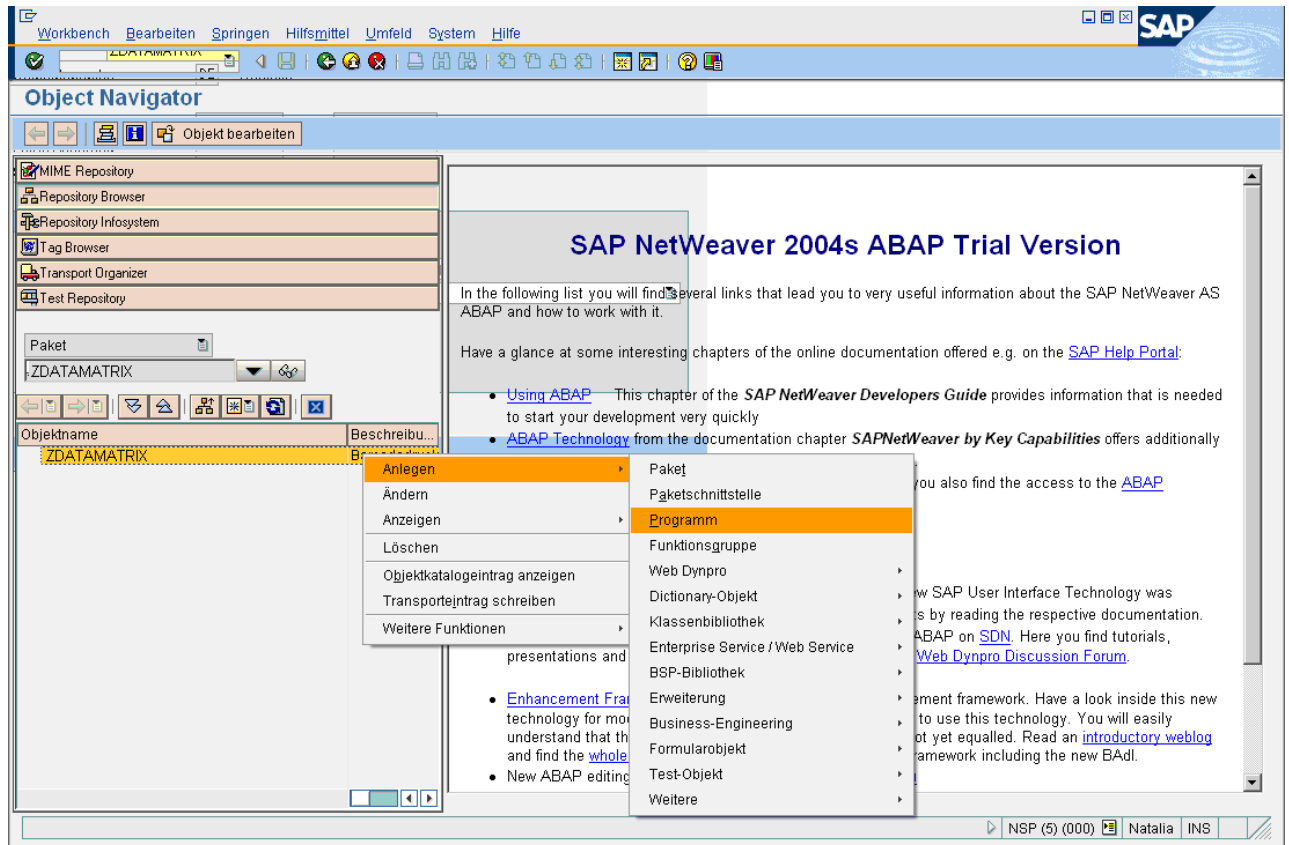
8. Klicken Sie auf das grüne Häkchen (**2**), um den Transportauftrag zu speichern.
9. Das Include wird gespeichert und der Quellcode (noch fast leer) erscheint auf der rechten Bildschirmseite.



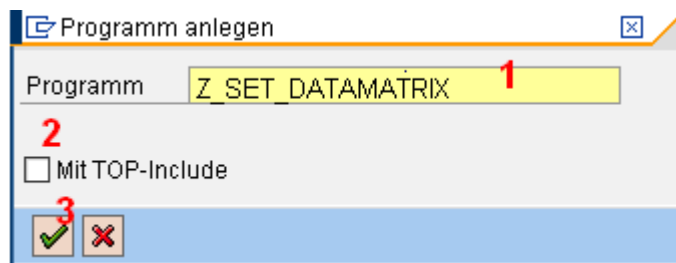
10. Löschen Sie den Inhalt des Quellcodes, der beim Anlegen des Includes automatisch erzeugt wurde.
11. Fügen Sie den Quellcode aus der entsprechenden Installationsdatei über die Zwischenablage ein und speichern Sie das Include. Damit ist die Installation des Includes abgeschlossen.
(Gehe zu Seite 59: [Anhang 4: Inhalt der Installationsdateien in ABAP](#) Programme einfügen.)

11 Anhang 3: Ein Programm in der ABAP Workbench anlegen.

1. Falls noch nicht geschehen, starten Sie den **Objekt Navigator** (Transaktion **SE80**) und wählen Sie das angelegte Paket (Entwickungsklasse) **ZQRCODE**.



2. Markieren Sie den Objektnamen **ZQRCODE** und klicken Sie auf die rechte Maustaste.
3. Wählen Sie aus dem Kontextmenü **Anlegen/Programm**.
4. Geben Sie in dem folgenden Dialog einen Namen für das anzulegende Programm (**1**) an, wählen Sie das Häkchen „Mit TOP-Include“ ab (**2**) und klicken Sie auf das grüne Häkchen (**3**).



5. Klicken Sie in dem folgenden Dialog auf „Sichern“ (1), um das Programm zu erstellen.

ABAP: Programmeigenschaften

Titel: Report Z_SET_DATAMATRIX

Originalsprache: DE Deutsch

Erstellt: 12.01.2007 BCUSER

Letzte Änderung:

Status: neu(überarbeitet)

Attribute

Typ: Ausführbares Programm

Status:

Anwendung:

Berechtigungsgruppe:

Logische Datenbank:

Selektionsbildversion:

☐ Editorsperre ☒ Festpunktarithmetik

☒ Unicodeprüfungen aktiv ☐ Start über Variante

1 Sichern

6. Achten Sie darauf, dass das Programm dem neu angelegten Paket (Entwicklungsklasse) **ZQRCODE** hinzugefügt wird (1) und klicken Sie auf das Diskettensymbol (2) um das Programm zu sichern.

Objektkatalogeintrag anlegen

Objekt: R3TR PROG ZDATAMATRIX

Attribute

Paket: ZDATAMATRIX 1

Verantwortlicher: CUST

Originalsystem: NSP

Originalsprache: DE Deutsch

2 Lokales Objekt Sperrübersicht

7. Es erscheint der Dialog für das Anlegen eines Transportauftrages. Dieser sollte zu diesem Zeitpunkt schon vorhanden sein, da er beim Anlegen des Paketes (Entwicklungsklasse) **Z_SET_QRCODE** erstellt wurde. Die Auftragsnummer sollte also diesmal die gleiche sein, wie beim Anlegen des Paketes (1).

Abfrage transportierbarer Workbench-Auftrag

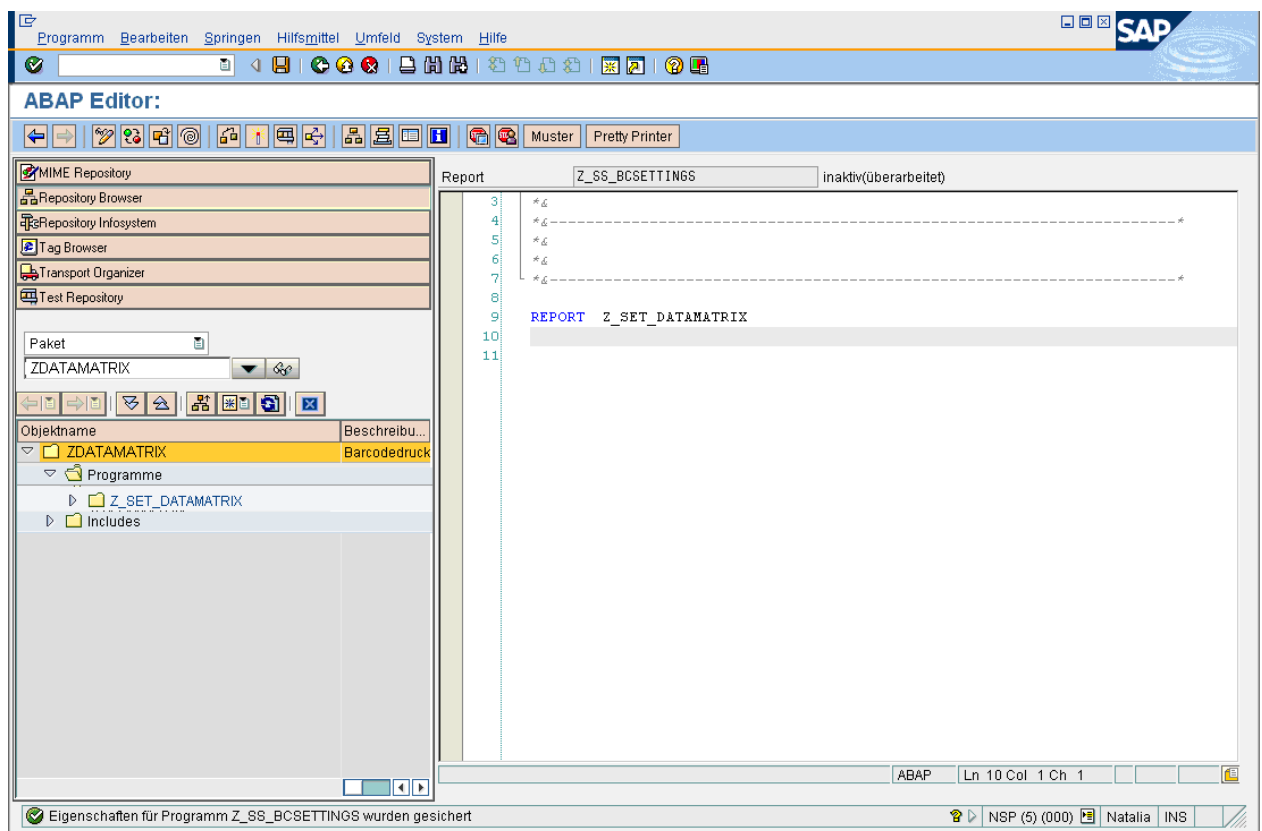
Programm: Z_SET_DATAMATRIX

Auftrag: NSPK900048 1 Workbench-Auftrag

Kurzbeschreibung: Barcodedruck mit Suchy MIPS RBarc+

2 [Green Checkmark Icon] Eigene Aufträge

8. Klicken Sie auf das grüne Häkchen (2), um den Transportauftrag zu speichern.
9. Das Programm wird gespeichert und der Quellcode (noch fast leer) erscheint auf der rechten Bildschirmseite.

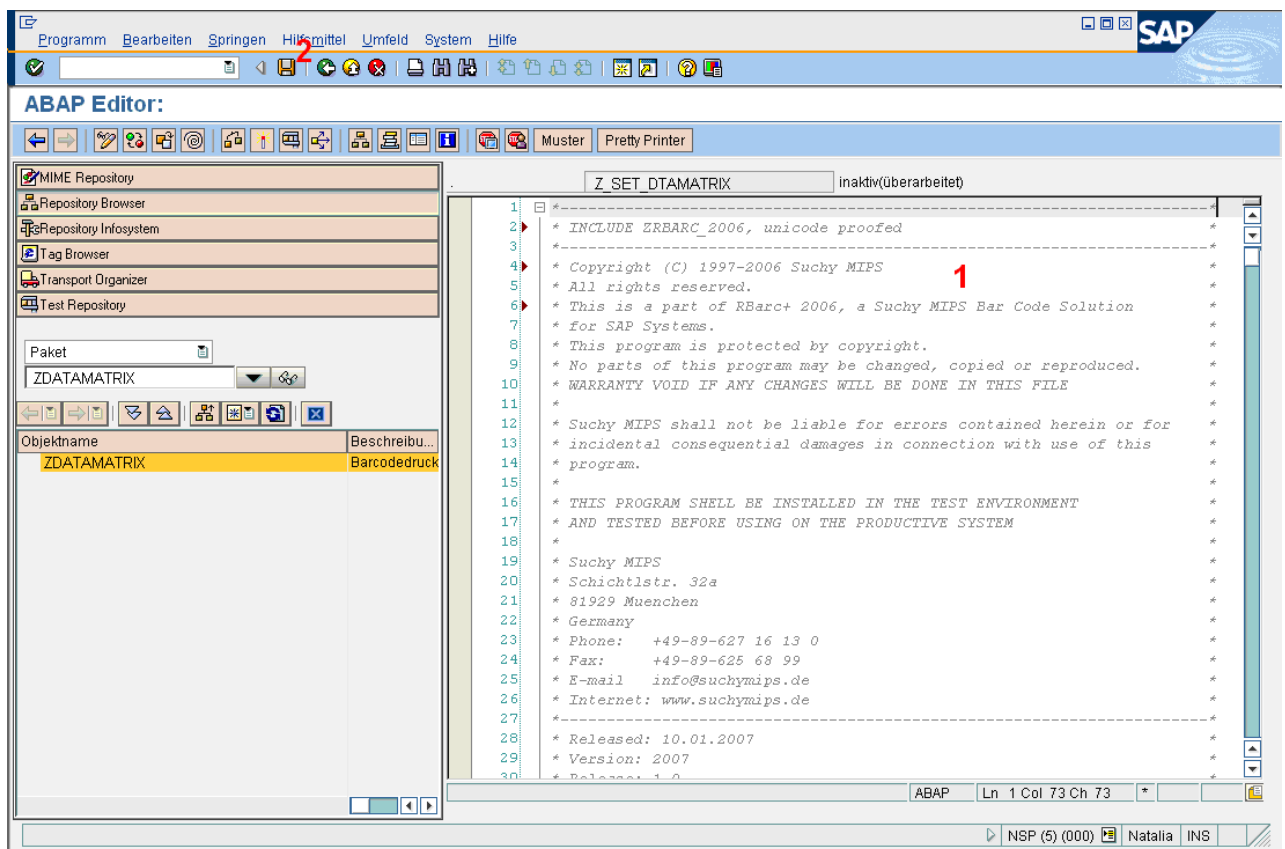


10. Löschen Sie den Inhalt des Quellcodes, der beim Anlegen des Includes automatisch erzeugt wurde.
11. Fügen Sie den Quellcode aus der entsprechenden Installationsdatei über die Zwischenablage ein und speichern Sie das Programm. Damit ist die Installation des Programms abgeschlossen. (Gehe zu Seite 59: [Anhang 4: Inhalt der Installationsdateien in ABAP Programme einfügen.](#))

12 Anhang 4: Inhalt der Installationsdateien in ABAP Programme einfügen.

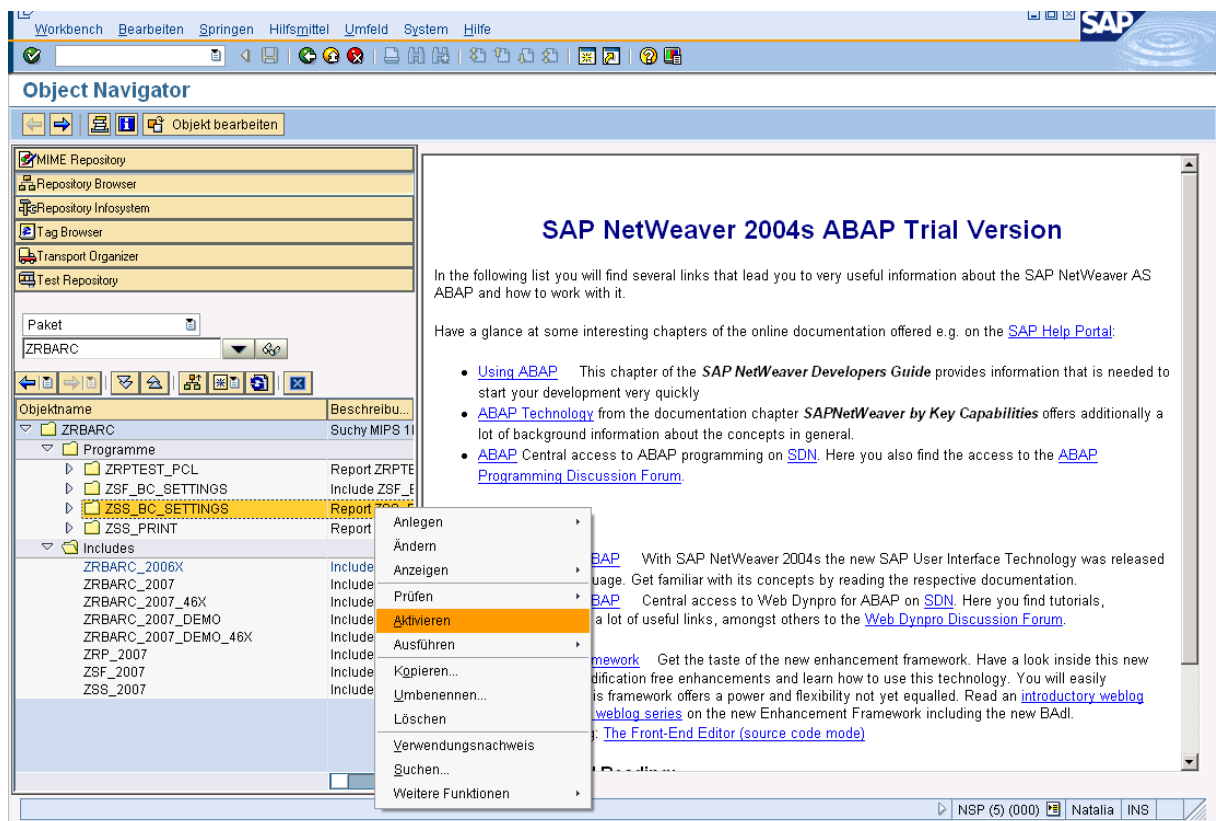
Dieser Vorgang betrifft sowohl Includes als auch Programme. Er wird anhand der Datei **Z_SET_QRCODE** erläutert, gilt aber grundsätzlich für alle Installationsdateien mit der Erweiterung **.PRG** und **.INC**.

1. Öffnen Sie die Datei **Z_SET_QRCODE.PRG** aus dem Verzeichnis **Install** mit dem Windows Editor (oder einem anderen Texteditor, falls Sie nicht unter Windows arbeiten).
2. Markieren Sie den gesamten Inhalt der Datei mit den Tasten **Strg+A**.
3. Kopieren Sie den Inhalt der Datei in die Zwischenablage mit den Tasten **Strg+C**.
4. Wechseln Sie im SAP GUI zum geöffneten Quellcode des Programms **Z_SET_QRCODE**.
5. Klicken Sie mit der Maus in das Fenster mit dem ABAP Quellcode (**1**) und drücken Sie die Tasten **Strg+V**, um den Quellcode aus der Zwischenablage einzufügen.
6. Klicken Sie anschließend auf das Diskettensymbol (**2**), um den Quellcode zu speichern.

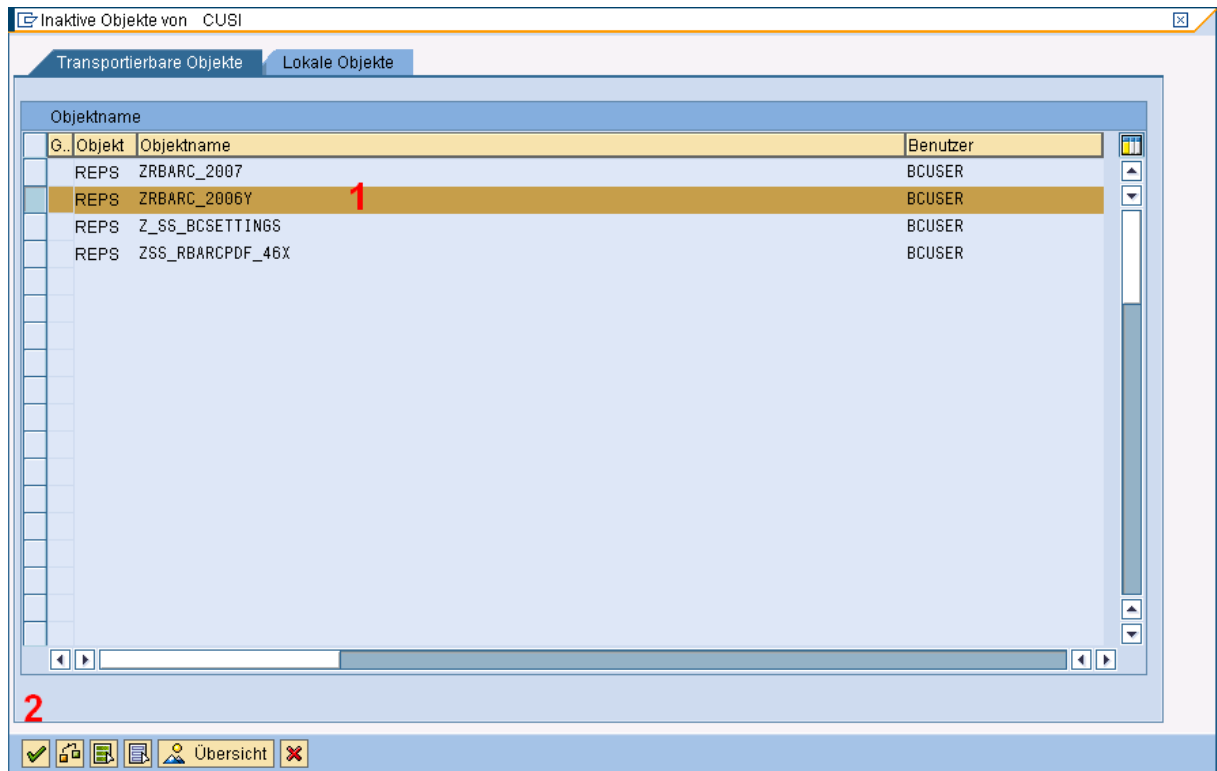


13 Anhang 5: Aktivieren von Includes und Programmen

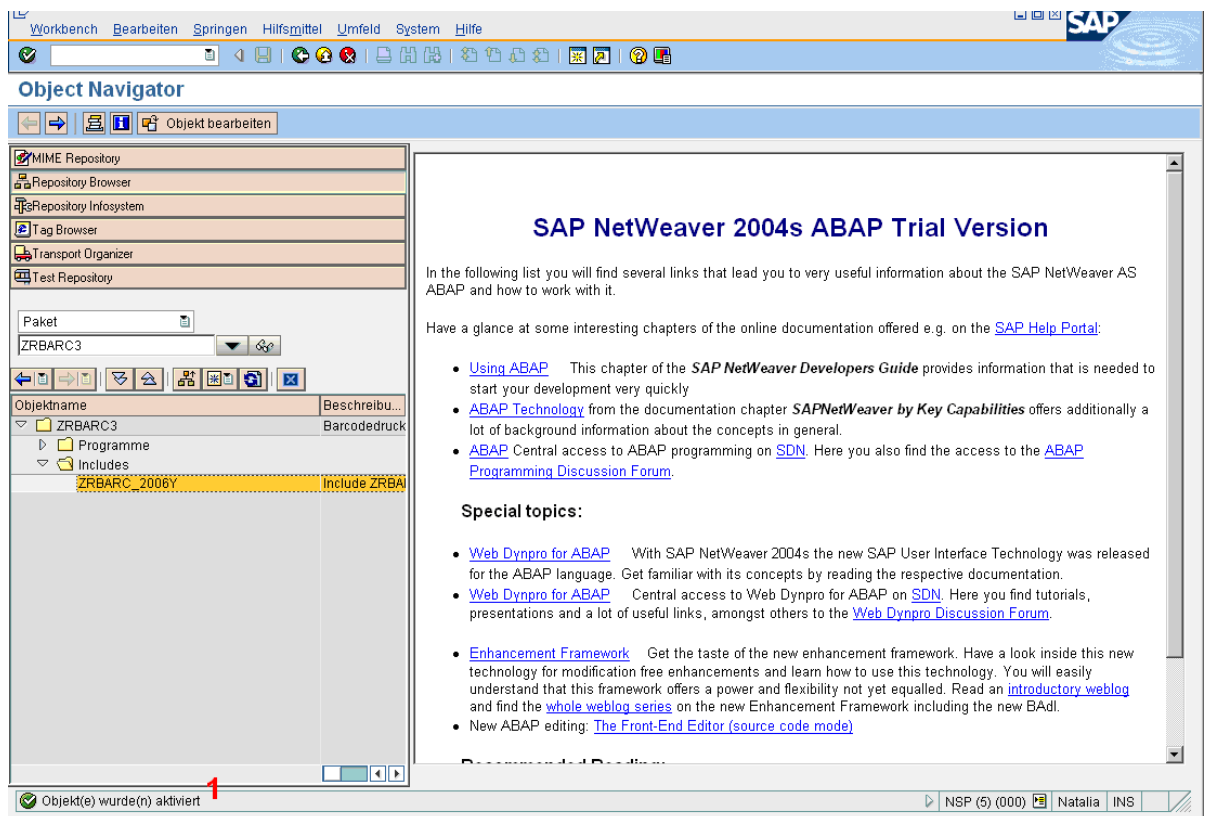
1. Öffnen Sie im SAP GUI den **Objekt Navigator** (Transaktion **SE80**).
2. Wählen Sie als Objekttyp „**Paket**“
3. Geben Sie den Paketnamen **ZQRCODE** ein.
4. Im Fenster Objektname erscheint der Name des Pakets mit den Baumknoten „**Programme**“ und „**Includes**“.
5. Expandieren Sie den Baumknoten **Programme** (1) und den Baumknoten **Includes** (2).
6. Wählen Sie das zu aktivierende Objekt, halten Sie die rechte Maustaste gedrückt und wählen Sie aus dem Kontextmenü „**Aktivieren**“



7. Es erscheint die Liste aller inaktiven Objekte. Das gewählte Objekt ist markiert (1). Klicken Sie auf das grüne Häkchen (2) um mit dem Vorgang fortzusetzen.

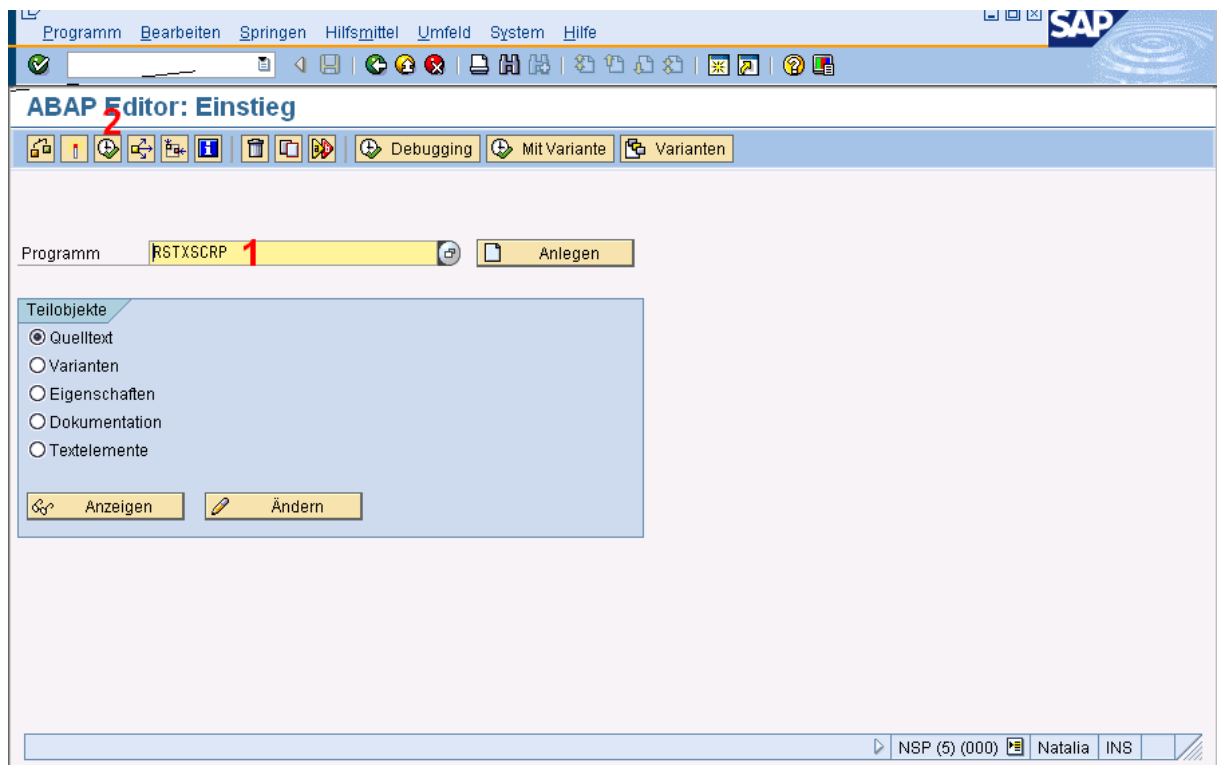


8. Nach der Aktivierung erfolgt eine Meldung in der Statusleiste (1). Nur bei erfolgreicher Aktivierung können die Programme einwandfrei arbeiten.



14 Anhang 6: Ein ABAP Programm ausführen.

1. Starten Sie die Transaktion **SE38**.
2. Geben Sie im Feld „**Programm**“ den Namen des auszuführenden Programms ein (**1**).
3. Klicken Sie im Menü auf das Startsymbol (**2**).



15 Anhang 7: Ein SAPscript Formular importieren (hochladen).

1. Starten Sie das SAP Standardprogramm **RSTXSCR**.
(Gehe zu: [Anhang 6: Ein ABAP Programm ausführen](#))
2. Wählen Sie den Objekttyp „**Formular**“ (1).
3. Tragen Sie im Feld „Objektname“ **ZSS_QR_FORM** ein (2).
4. Tragen Sie den Modus „**IMPORT**“ ein (3).
5. Klicken Sie auf das Startsymbol (4)

Programme Bearbeiten Springen System Hilfe

Upload/Download von SAPscript-Objekten

Objektselektion und Modussteuerung

☒ Formular 1
☐ Stil
☐ Standardtext
Text-ID: ST
Sprache: DE
☐ Gerätetyp
Objektname: ZSS_DM_FORM 2
Modus (EXPORT,IMPORT): IMPORT 3

Steuerparameter für Datei-Operation

☒ Von/auf Frontend
☐ Von/auf Applikationsserver
Dateiname: C:\temp*****&&&&
☐ Binäres Dateiformat
☐ Dateinhalt anzeigen

Kontrolle über Sprachversionen

Sprachenvektor:
☐ Nur Originalsprache export.

NSP (5) (MOM) Natalia INS

6. Es erscheint der Dialog für den Objektkatalogeintrag. Fügen Sie das Formular dem Paket (Entwicklungsgruppe) **ZQRCODE** hinzu (1). Klicken Sie auf das Diskettensymbol (2) um den Eintrag zu speichern.

Objektkatalogeintrag anlegen

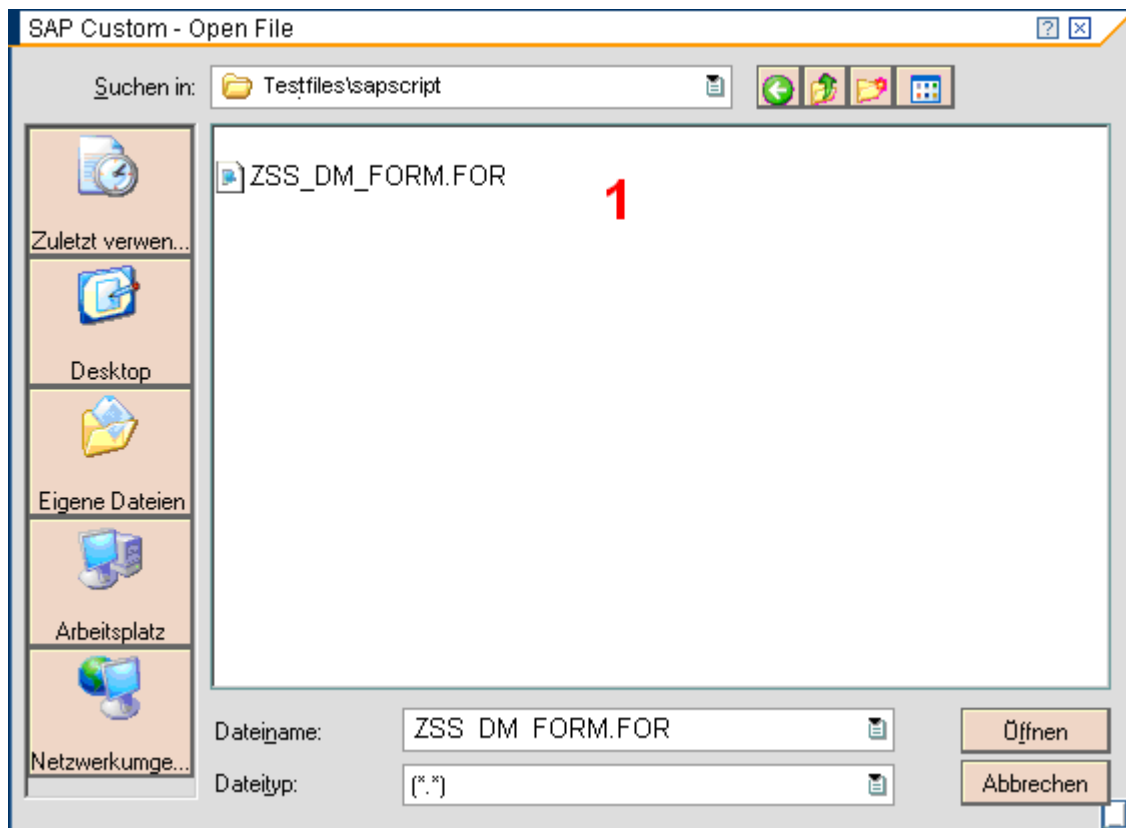
Objekt: R3TR FORM ZSS_DM_FORM

Attribute

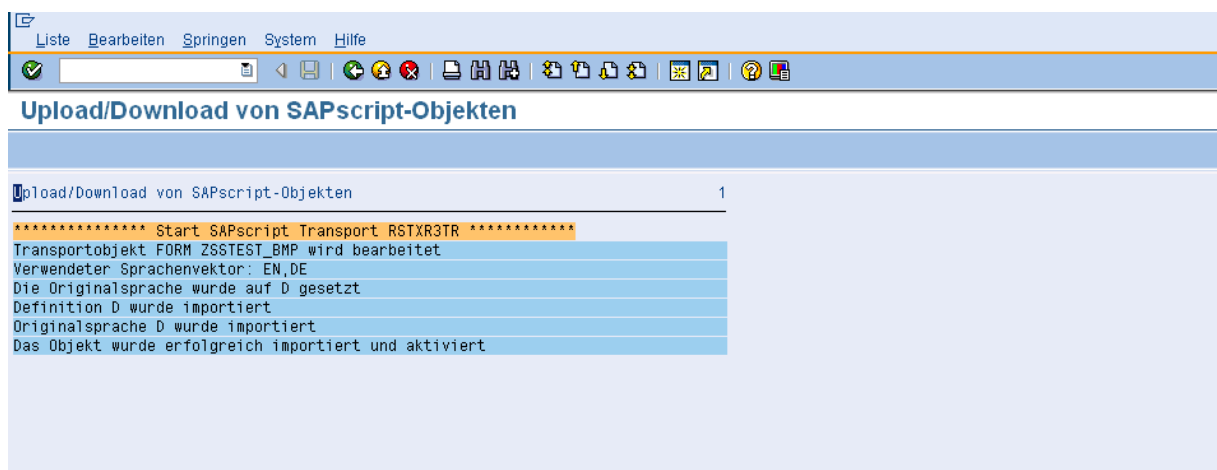
Paket: ZDATAMATRIX 1
Verantwortlicher: CUST
Originalsystem: NSP
Originalsprache:

2
Lokales Objekt Sperrübersicht

7. Es erscheint der Dialog für den Transportauftrag. Lassen Sie den bisherigen Transportauftrag bestehen (1) und klicken Sie auf das grüne Häkchen, um den Eintrag zu speichern (2).
8. Jetzt erscheint das Dateiauswahl Menü. Wählen Sie die Datei **ZSS_QR_FORM.FOR** aus dem Verzeichnis **TestForms** (1) und klicken Sie auf **Öffnen**.

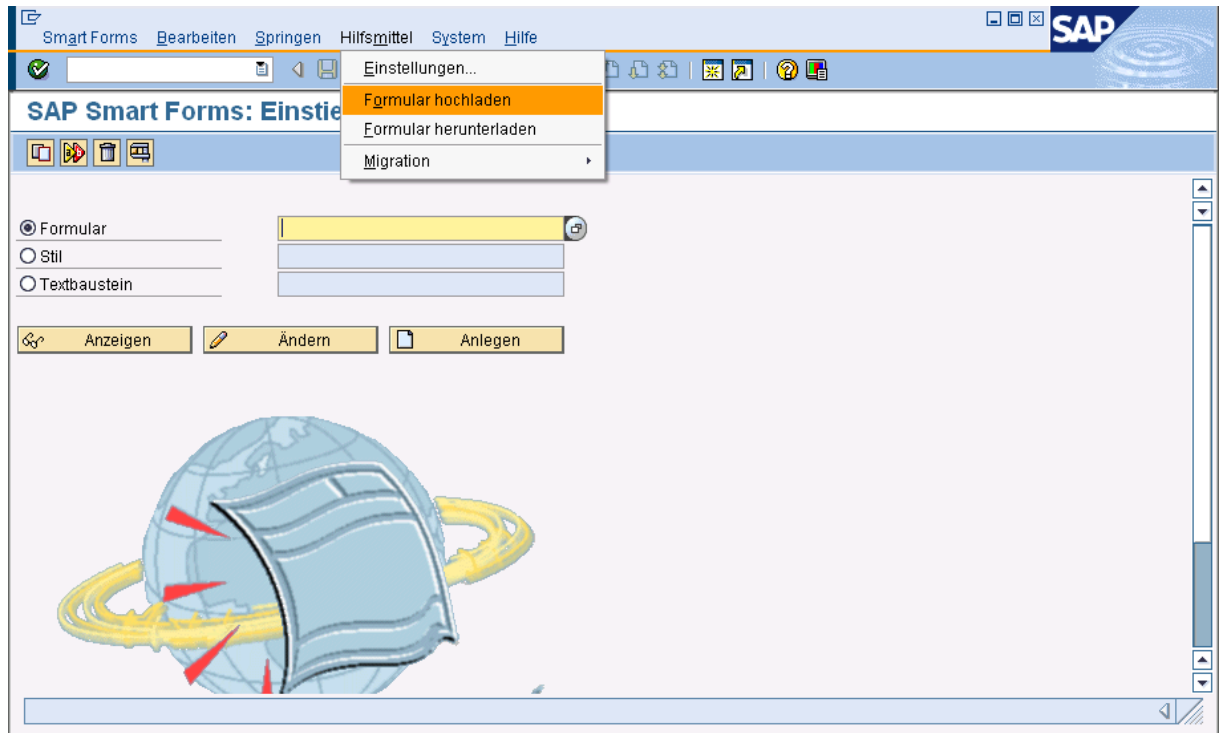


9. Das Formular wird importiert und zum Schluss erscheint eine Meldung, ähnlich, wie diese:

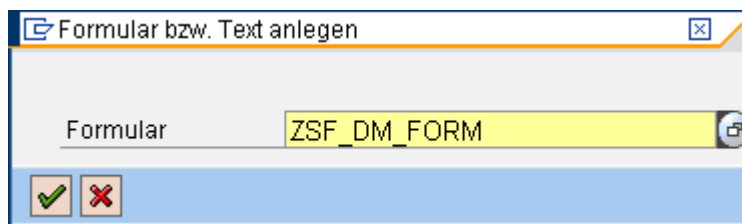


16 Anhang 8: Ein Smart Forms Formular hochladen.

1. Starten Sie die Transaktion **Smartforms**.
2. Wählen Sie aus dem Menü **Hilfsmittel / Formular hochladen**

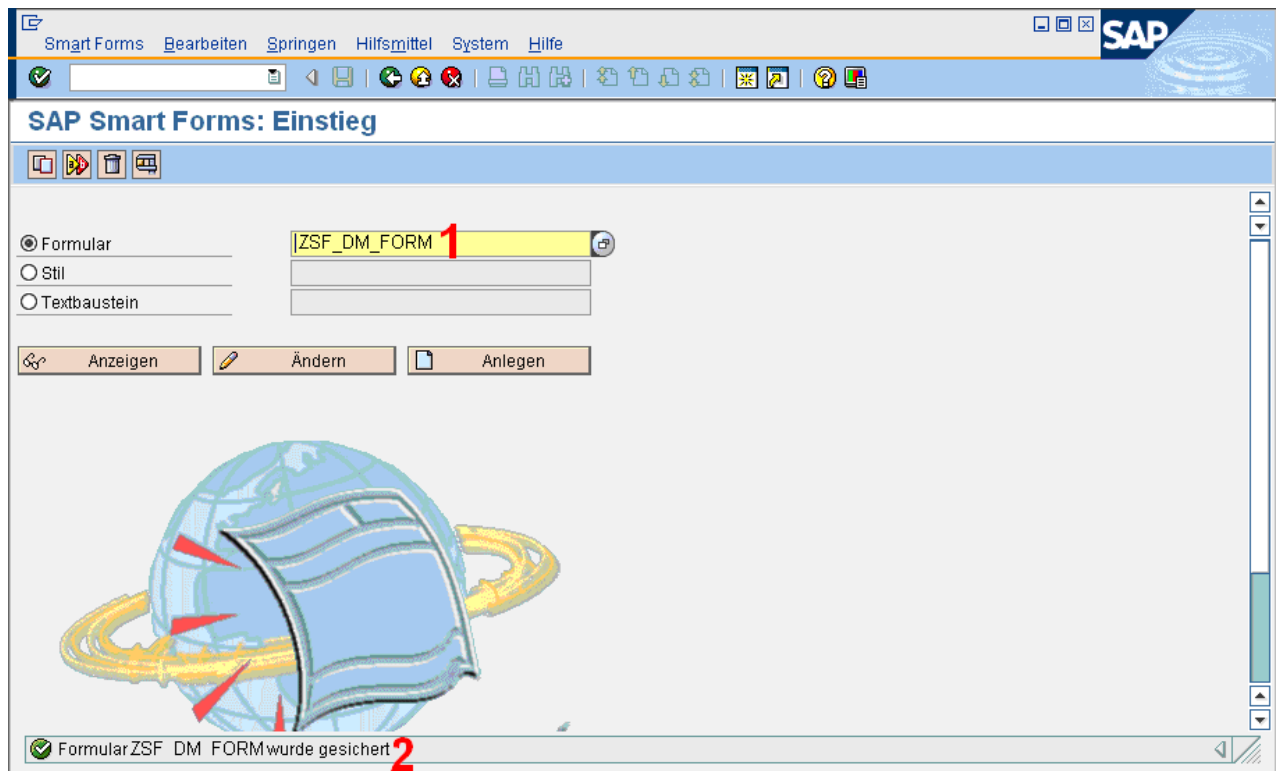


3. Geben Sie im nächsten Dialog den Formularnamen **ZSF_QR_FORM** ein und klicken Sie auf das grüne Häkchen.



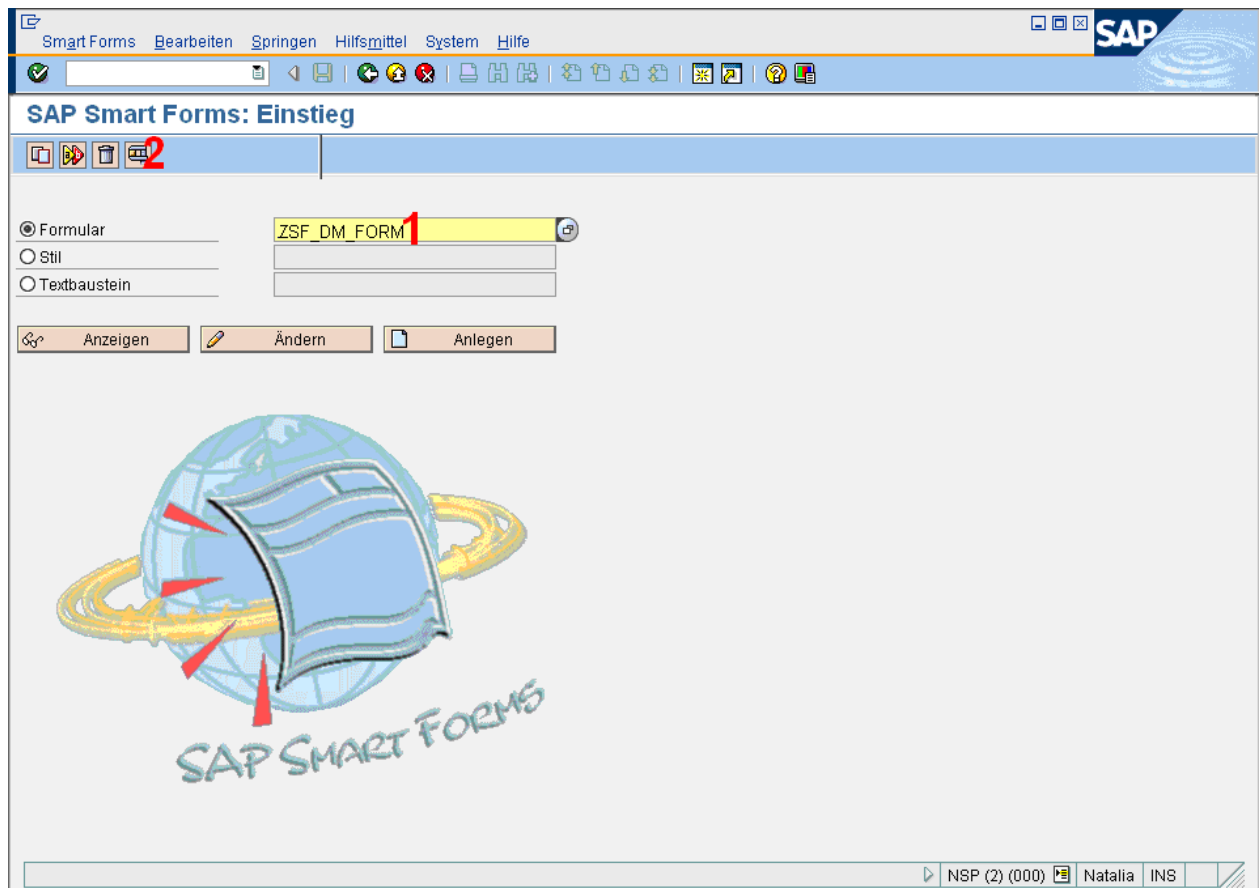
4. Es erscheint der Dateiauswahl Dialog. Wählen Sie die Datei **ZSF_QR_FORM.XML** aus dem Verzeichnis **TestForms** und klicken Sie auf **Öffnen**.

5. Das Formular wird hochgeladen. Danach erscheint der Name des hochgeladenen Formulars im Feld **Formular** (1) und eine Meldung in der Statusleiste (2)

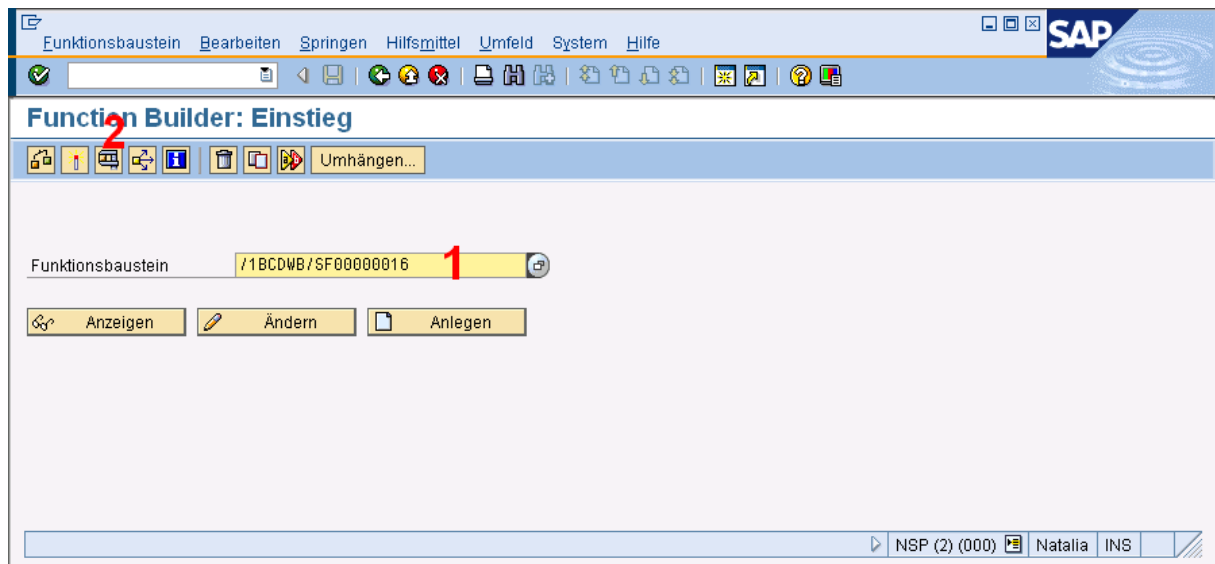


17 Anhang 9: Smart Forms Formular testen.

1. Starten Sie die Transaktion **Smartforms**.
2. Geben Sie im Feld **Formular** den Namen **ZSF_QR_FORM** ein (1) und klicken Sie anschließend auf das Symbol **Testen** (2).



3. Das System erzeugt einen Funktionsbaustein, der im nächsten Dialog erscheint (1). Klicken Sie erneut auf das Symbol **Testen** (2).



4. Klicken Sie im nächsten Dialog auf das Symbol **Ausführen** (1).

The screenshot shows the SAP 'Funktionsbaustein testen: Eingabebild' (Function Block Test: Input Screen) dialog. The title bar includes 'FBAusteine', 'Bearbeiten', 'Springen', 'Hilfsmittel', 'System', and 'Hilfe'. The main area contains a 'Debugging' button and a table for input parameters. A red arrow points to the 'Ausführen' (Execute) button in the top toolbar.

Import-Parameter	Wert
ARCHIVE_INDEX	00.00.0000
ARCHIVE_INDEX_TAB	0 Einträge
ARCHIVE_PARAMETERS	
CONTROL_PARAMETERS	
MAIL_APPL_OBJ	
MAIL_RECIPIENT	
MAIL_SENDER	
OUTPUT_OPTIONS	<Initial>

5. Danach startet das Druckprogramm. Geben Sie den Namen des Druckers ein, auf dem der Barcode gedruckt werden soll (1).
6. Setzen Sie ein Kästchen im Feld **Sofort Ausgeben** (2)
7. Klicken Sie auf Drucken (3), falls Sie das Formular mit den Barcodes auf dem Drucker ausgeben möchten, oder auf Druckansicht (4), falls Sie das Formular mit den Barcodes auf dem Bildschirm betrachten möchten.

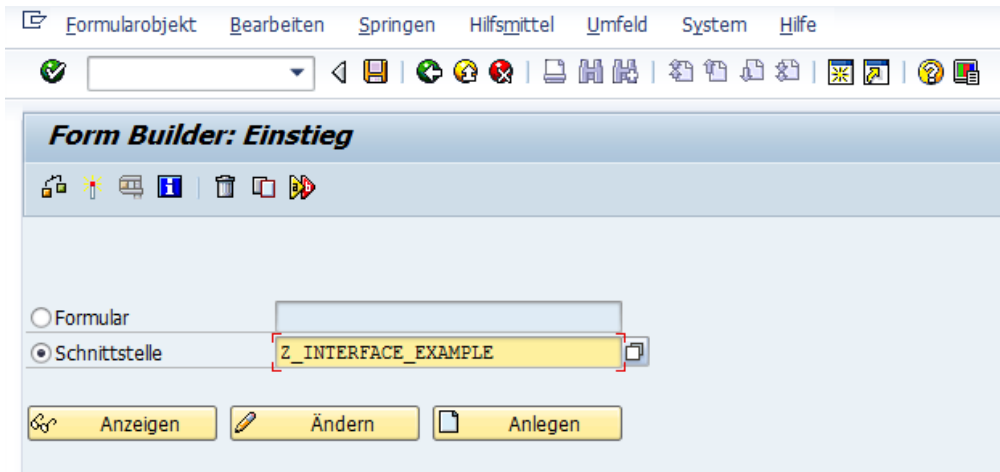
The screenshot shows the SAP 'Drucken:' (Print) dialog. It contains several sections for configuring the print job. Red numbers 1 through 4 mark specific fields and buttons:

- 1. **Ausgabegerät** (Printer Name): LP01
- 2. **Sofort ausgeben** (Print Immediately checkbox): Checked
- 3. **Drucken** (Print button)
- 4. **Druckansicht** (Print Preview button)

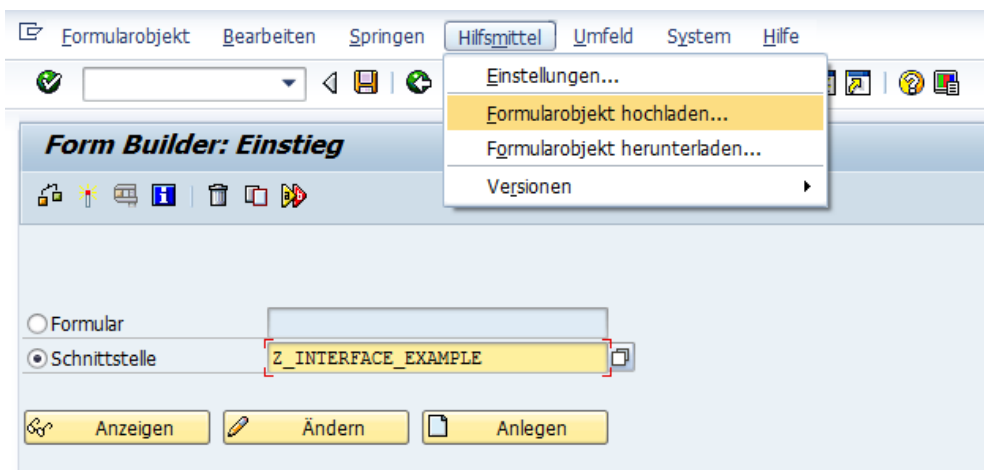
Other visible fields include: **Seitenausw.** (Page Selection), **Spool-Auftrag** (Spool Job) with Name SMART and BCUSER, **Spool-Steuerung** (Spool Control) with Spool-Verweildauer 8 Tag(e) and Ablagemodus Nur Drucken, **Exemplare** (Copies) with Anzahl 1, and **Deckblatteinstellungen** (Cover Sheet Settings) with SAP-Deckblatt nicht ausgeben.

18 Anhang 10: Eine Interactive Forms Schnittstelle hochladen.

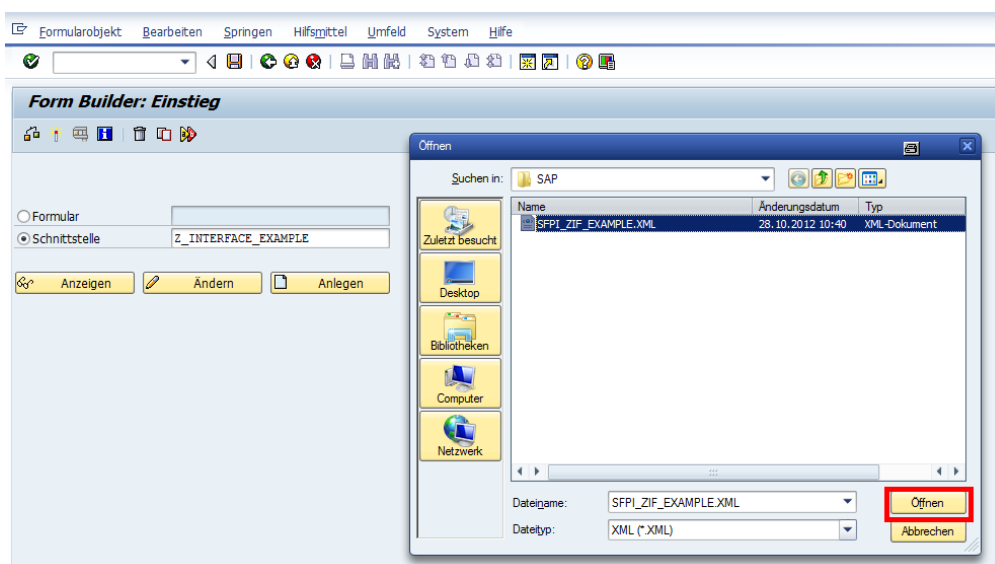
1. Starten Sie den FormBuilder durch Aufruf der Transaktion **SFP**.
2. Markieren Sie den Radiobutton **Schnittstelle** und geben Sie der Schnittstelle einen Namen.



3. Wählen Sie aus dem Menü **Hilfsmittel** → **Formularobjekt hochladen**.

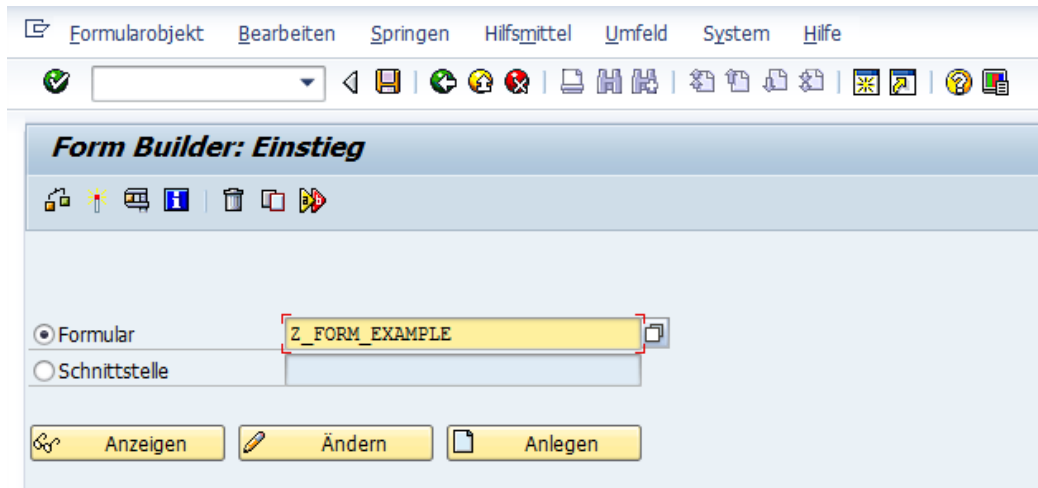


4. Wählen Sie aus dem Dateiselektor die XML-Datei aus und betätigen Sie den Button **Öffnen**.

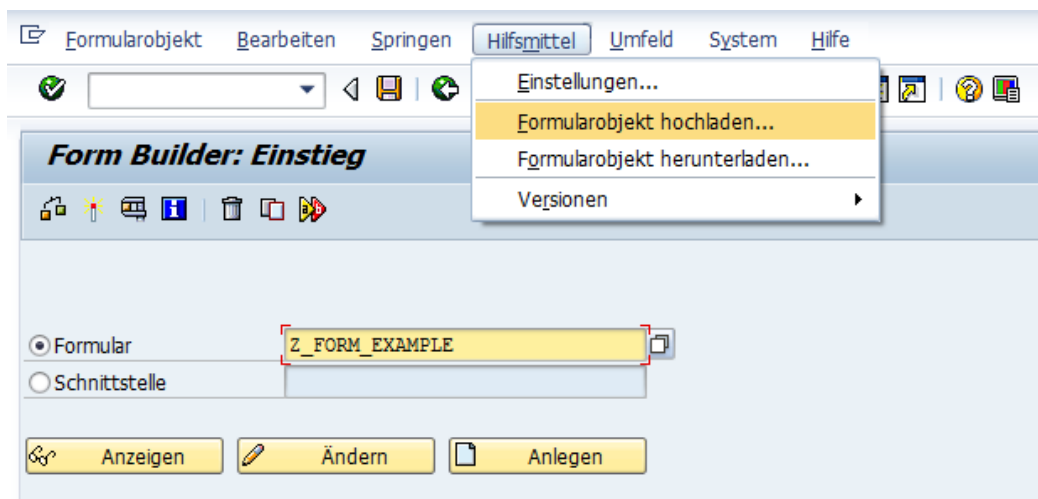


19 Anhang 11: Ein Interactive Forms Formular hochladen.

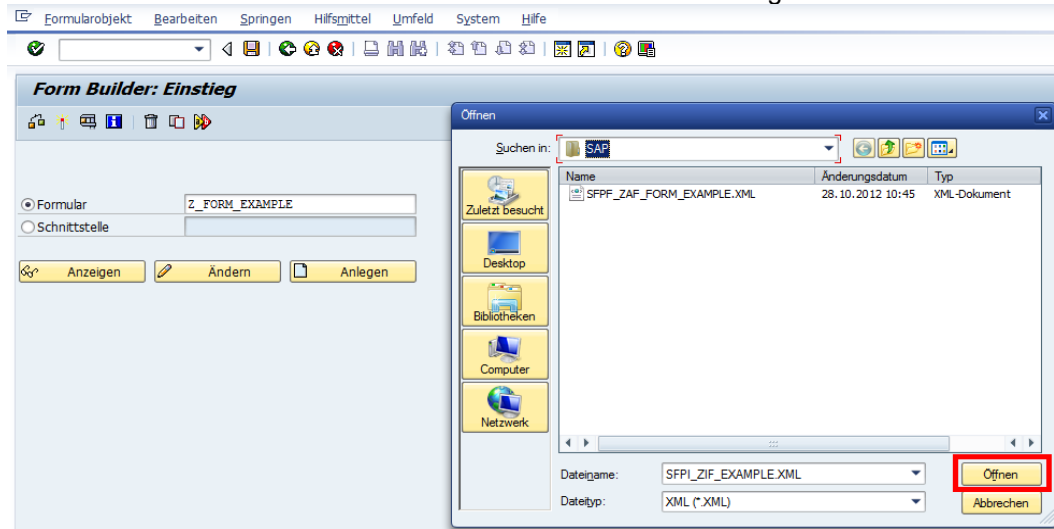
1. Starten Sie den FormBuilder durch Aufruf der Transaktion **SFP**.
2. Markieren Sie den Radiobutton **Formular** und geben Sie dem Formular einen Namen.



3. Wählen Sie aus dem Menü **Hilfsmittel** → **Formularobjekt hochladen**.

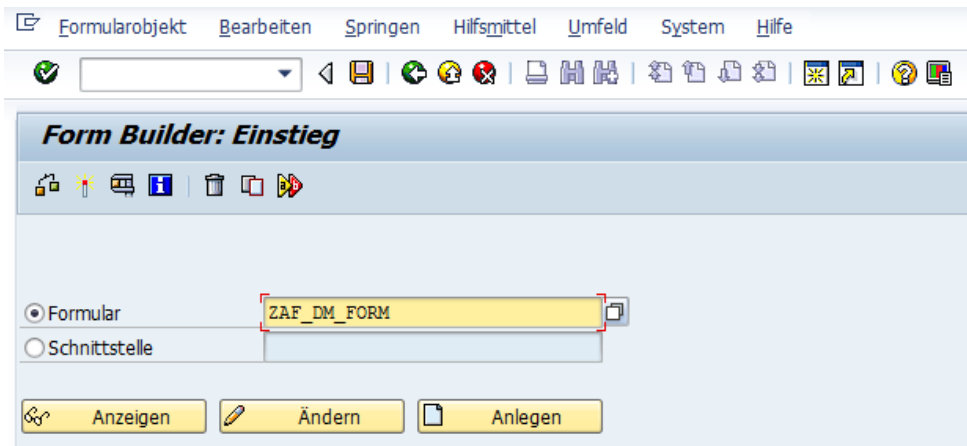


4. Wählen Sie aus dem Dateiselektor die XML-Datei aus und betätigen Sie den Button **Öffnen**.

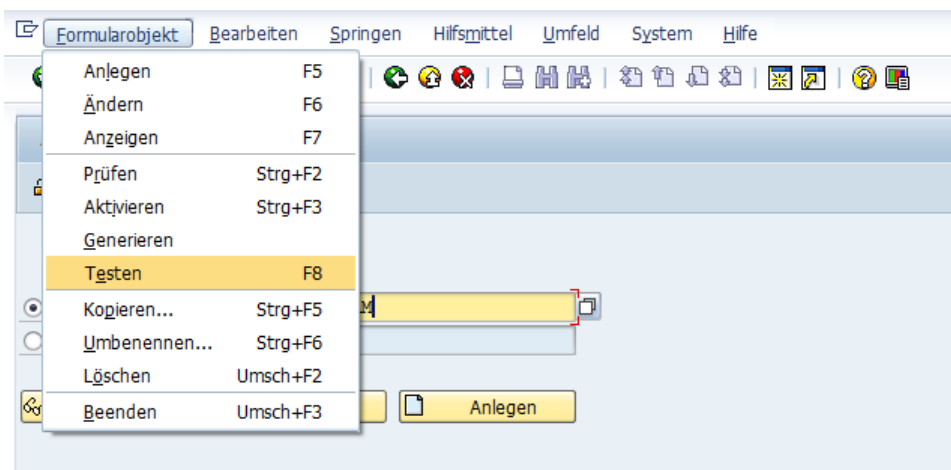


20 Anhang 12: Ein Interactive Forms-Formular testen

1. Starten Sie den FormBuilder durch Aufruf der Transaktion **SFP**.
2. Markieren Sie den Radiobutton **Formular** und geben Sie den Namen des Formulars ein (hier im Beispiel **ZAF_QR_FORM**).

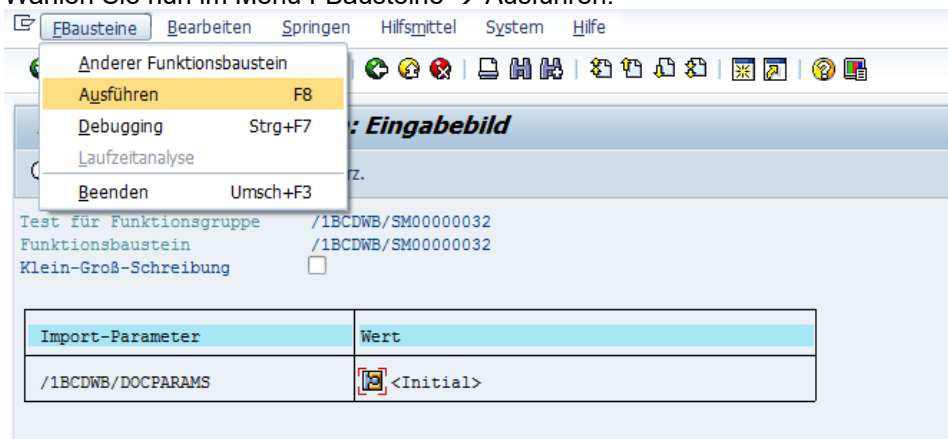


3. Wählen Sie im Menü Formularobjekt → Testen.



Der generierte Funktionsbaustein wird nun angezeigt.

4. Wählen Sie nun im Menü FBausteine → Ausführen.



Import-Parameter	Wert
/1BCDWB/DOCPARAMS	<Initial>

5. Geben Sie einen Drucker ein, dem kein SAPWIN oder SWIN-Gerätetyp zugeordnet ist und betätigen Sie den Button **Druckbildansicht**.

System Hilfe

SAP

Drucken:

Ausgabegerät: LOCA

Spoolauftrag

Name: PBFORM LOCA XXXX

Text für Deckblatt:

Berechtigung:

Spool-Steuerung

☒ Sofort ausgeben

☒ Löschen nach Ausgabe

☐ Neuer Spool-Auftrag

☐ Spool-Auftrag abschließen

Spool-Verweildauer: 8 Tag(e)

Ablagemodus: 1 Nur Drucken

Exemplare

Anzahl Exemplare: 1

Deckblatteinstellungen

SAP-Deckblatt: nicht ausgeben

Empfänger:

Abteilung:

Drucken Druckansicht

6. Das Formular sollte nun in der Druckbildansicht zu sehen sein.

Print Preview of LP01 Page 00001 of 00001

RBarc/Datamatrix

2D Barcode GS1 Datamatrix ECC 200 generated on a SAP System
A solution from Suchy MIPS www.suchymips.de

This demo presents 2 Datamatrix-Labels with identical content but encoded in different ways.

Both Datamatrix codes encode following data: '(>[RS]06[GS]2L'

[RS] means the function code for Record Separator (ASCII Value 30 dec)
[GS] means the function code for Group Separator (ASCII Value 29 dec)

The left Datamatrix-Code was encoded using different variables with different Data Type Flags:

dmvar1 = '(>'. dflag = 'T'.
dmvar2 = 'RS'. dflag = 'F'.
dmvar3 = '06'. dflag = 'T'.
dmvar4 = 'GS'. dflag = 'F'.
dmvar5 = '2L'. dflag = 'T'.

The right Datamatrix-Code was encoded using one variable which encodes all data in ASCII-HEX.

dmvar1 = '5B283E1E30361D324C'.
DFLAG = 'X'.

Available Data Type Flags for data encoding are:

'T' - any text
'X' - data encoded as an ASCII-Hex string (e.g. 5B283E1E...)
'F' - function codes like 'GS' 'RS' 'EOT'
'B' - Byte with values 0 - 255
'L' - ECI (Extended Channel Interpretation)

For more details refer to the manual.

21 Anhang 13: Swiss QRCode

RBarc/QRCode enthält eine Option zum Drucken des Swss-QRCodes.

Der Schweizer QR/Code ist nicht vollständig konform mit ISO 18004, der den eigentlichen QRCode spezifiziert. Der Barcode ist mit einem Schweizer Kreuz in der Mitte gekennzeichnet und hat - unabhängig von der Menge codierter Daten - eine fixe Größe von 46 x 46 mm. Dies hat zur Folge, dass die Modulgröße nicht wie beim ISO-QRCode vorgegeben werden kann, sondern dynamisch berechnet werden muss.

Um den Swiss QRCode mit RBarc / QRCode zu drucken, muss nur der Parameter X_DIMM auf den Wert 1046 gesetzt werden.

In diesem Fall wird der X_DIMM-Wert automatisch berechnet so, dass der gesamte Barcode 46 x 46 mm groß wird.

Der Parameter ECC_Level muss laut Spezifikation auf den Wert „M“ gesetzt werden.



Beispiel eines Swiss QRCodes generiert mit RBarc/QRCode

Bitte beachten:

- Der "Swiss Payment Standard" lässt als Feldtrennung entweder <LF> (Linefeed) oder <CR><LF> (Carriage Return + Linefeed) zu.
- Manche QR Codes beinhalten eine sog. QR Referenz. Die QR-Referenz entspricht im Aufbau der ESR-Referenz (26 numerische Zeichen gefolgt von einer Prüfziffer nach Modulo 10 rekursiv, und kann vom Rechnungssteller als strukturierte Referenz verwendet werden. Für die Berechnung der Prüfziffer für die QR Referenz liefern wir das ABAP Programm Z_QR_REFERENCE mit. Die QR Referenz kann aus dem Formular ermittelt werden, z.B.

```
REPORT Z_QR_REFERENCE_GET.

data: refnumber(26).
data: checksum.

refnumber = '21000000000313947143000901'.
perform qr_reference in program z_qr_reference using refnumber
                                                    changing checksum.

write:/'This is my refnumber: ', checksum.
```

Zusammenfassung

- ✓ Der Swiss QRCode wurde in RBarc/QRCode als Option implementiert (ab Vers. 1.7)
- ✓ Folgende Parameter müssen für den Swiss QRCode verwendet werden:
 - x_dim = 1046.
 - ecc_level = 'M'.
- ✓ Als Feldtrennzeichen muss **LF** oder **CR+LF** verwenden werden.
- ✓ Nach dem letzten Element entfällt die Zeilenschaltung.