

## Tips und Techniken für die AS/400



[Künstler Burgy Zapp](#)

Endlich ist sie da, unsere lang ersehnte jährliche Ausgabe mit Tips und Techniken! In diesem Abschnitt haben wir fünfzig der besten Tips, Tricks und Techniken für die AS/400 zusammengestellt. Einige davon stammen von Anwendern wie Ihnen, andere von den technischen Redakteuren von NEWS/400. Das Ergebnis ist eine prall gefüllte Kiste mit unverzichtbaren Tips, Tricks und Techniken. Überzeugen Sie sich selbst, wie Sie sich mit diesen erstaunlichen AS/400-Tips Ihre tägliche Arbeit erleichtern können.

Dieser Abschnitt enthält Tips zur RPG- und CL-Programmierung, Informationen zu ILE und SQL, Nützliches zu Datums- und Zeitoperationen, Ratschläge für die Systemverwaltung und vieles mehr. Wenn Ihnen das noch nicht reichen sollte, finden Sie weitere interessante und nützliche Tips, z.B. 25 Tips zur Netzwerkanbindung und Client/Server-Programmierung in NEWS/400 Februar 98, Seite 42, Fortsetzung in der April-Ausgabe von NEWS/400.. Wem auch das noch nicht genug ist, der findet weitere Tips unter: <http://www.news.400.com> Nach der Lektüre dieser Tips werden nicht nur Sie selbst beeindruckt sein, sondern auch Ihre Freunde und Kollegen werden staunen. Entscheidend ist jedoch, daß die Tricks Ihnen helfen, das Beste aus Ihrer AS/400 herauszuholen. Und das Erfolgsrezept heißt: NEWS/400:

[Auswertung logischer Ausdrücke](#)

[Vereinfachte Datendefinitionen](#)

[Vermeiden von Array-Indexfehlern](#)

[Zusammensetzen von Programmen mit Teildatei-Kopien](#)

[RPG IV und der Gleitkomma-Datentyp](#)

[Schnellere Array- und Stringbearbeitung](#)

[Verwenden von leeren logischen Anweisungen](#)

[Arrays vs. Datenstruktur mit Mehrfachvorkommen](#)

[Konvertieren von RPG III-Code zu RPG IV mit Code/400](#)

RPG

## Auswertung logischer Ausdrücke

RPG IV erlaubt Ihnen, logische Ausdrücke für alle Operationen zu kodieren, die den erweiterten Faktor 2 unterstützen. Viele von Ihnen werden bereits mit der Kodierung der logischen Ausdrücke für If-, DoW-, DoU- und When-Operationen vertraut sein. Sie können jedoch auch logische Ausdrücke für Eval- und Return-Operationen kodieren. Durch logische Ausdrücke läßt sich der Code erheblich vereinfachen. Die folgenden Beispiele sollen das verdeutlichen. Eine Prozedur ruft ein API auf, das die standardmäßige API-Fehlerdatenstruktur zurückgibt. Wenn der Aufruf an das API keinen Fehler aufweist, gibt die Prozedur \*On zurück. Falls das API mit einem Fehlerzustand endet, gibt die Prozedur \*Off zurück. Anstatt der Langversion:

**können Sie auch die folgende Kurzform eingeben:**

Beide Codepassagen führen zu einem absolut identischen Ergebnis. Die Kurzfassung ist jedoch wesentlich kompakter. Sehen wir uns die Kurzfassung einmal genauer an. Zunächst wertet das Programm den in der Klammer stehenden logischen Ausdruck aus. Wenn ErDtaLen gleich 0 ist, ist der Ausdruck wahr, und das Ergebnis ist \*On. Wenn ErDtaLen nicht gleich 0 ist, ist der Ausdruck falsch, und das Ergebnis ist \*Off. Anschließend gibt das Programm als Ergebnis des Ausdrucks \*On oder \*Off an die aufrufende Prozedur zurück. Nachdem ich die Grundidee aufgezeigt habe, können wir uns nun einem etwas komplexeren Beispiel zuwenden. Stellen Sie sich eine Anzeige vor, in der Sie ein Element in einer Datei erstellen, ändern oder löschen können. Die DDS für die Anzeige definiert einen Titel für jeden Vorgang, und die Ausgabe wird durch Bedingungsindikatoren gesteuert:

**Die folgenden C-Spezifikationen bestimmen die Bedingungsindikatoren:**

Für die Option Create (Erstellen) ist \*In01 auf Ein (On) gesetzt. \*In02 und \*In03 sind auf Aus (Off) gesetzt, und der Titel „Create Item“ wird eingebledet. Wenn Sie sich etwas in die hier behandelte Logik vertiefen, werden Sie feststellen, daß sich die Programmlogik ebenso auf die Optionen Change (Ändern) und Delete (Löschen) anwenden läßt.

## **Vereinfachte Datendefinitionen**

Sie können die Definition der Variablen in Ihren Programmen vereinfachen, indem Sie sie wie andere Variablen definieren. Definitionen mit „Like“ (Wie) verbessern die Lesbarkeit und Wartungsfähigkeit Ihrer Programme. Verwenden Sie das Schlüsselwort Like in den D-Spezifikationen, wenn Sie eine Variable wie eine andere definieren möchten. Sie können die „Like“-Definitionen einsetzen, um Kopien von extern beschriebenen Dateien zu halten und Variablen von Standardtypen zu erstellen. Erstellen Sie eine Variable, die eine Kopie eines Feldes aus einer extern beschriebenen Datei hält (die in einem Programm benutzt wird), indem Sie die Variable wie das Feld definieren. Das folgende Beispiel zeigt das Erstellen einer Variable, um eine Kopie von Feld ItmNbr in der Datei ItmMst zu speichern, die in einem Programm verwendet wird. Geben Sie den folgenden Code ein:

```
D SavItmNbr Like ( ItmNbr )
```

Sie können Variablen von Standardtypen erstellen, indem Sie die in Abbildung 1 aufgeführte Teildatei-Kopie StdTypH verwenden. StdTypH definiert Standarddatentypen, die in beliebigen Programmen verwendet werden können. StdIdx definiert einen Index-Datentyp für ein beliebiges Array, und StdNam definiert einen Datentyp für OS/400-Namen, z.B. Namen von Objekten und Teildateien. Um eine Variable von einem Standardtyp zu erstellen, müssen Sie zunächst eine Copy-Anweisung für die Teildatei StdTypH in den D-Spezifikationen des Programms kodieren. Definieren

Sie anschließend die Variable wie einen der Standardtypen in StdTypH. Wenn Sie beispielsweise ein Objektnamen-Array mit einem Array-Index erstellen möchten, geben Sie folgendes ein:

```
/Copy SrcFile,StdTypH D ObjNamAry Like( StdNam ) D Dim( 999 ) D ObjNamIdx Like( StdIdx )
```

## Vermeiden von Array-Indexfehlern

Mit etwas Sorgfalt können Sie Array-Indexfehler in Ihren RPG IV-Programmen vermeiden. Dabei sind die D-Spezifikationen von entscheidender Bedeutung. Achten Sie darauf, bei jeder Definition eines Arrays auch einen Index zu definieren, der groß genug für den Verweis auf jedes Element in dem Array ist. Außerdem müssen Sie eine benannte Konstante definieren, mit der die Gesamtgröße des Arrays definiert wird. Bei der Definition eines Objektnamen-Arrays kann der Code wie in diesem Beispiel aussehen:

```
/Copy SrcFile,StdTypH D ObjNamAry Like(StdNam) D Dim(999 ) D ObjNamIdx Like(StdIdx ) D  
ObjNamSiz C %Elem(ObjNamAry)
```

Das Array ObjNamAry enthält 999 Elemente. ObjNamIdx bezeichnet den Index für das Array und ist definiert wie StdIdx, das groß genug ist, um einen Index für ein Array von beliebiger Größe zu speichern. (Das Array und der Index wurden mit Standarddatentypen definiert, wie sie im vorherigen Abschnitt „Vereinfachte Datendefinitionen“ beschrieben wurden.) ObjNamSiz definiert die Array-Größe mit Hilfe der eingebauten %Elem-Funktion. Wenn Sie die Größe von ObjNamAry ändern wollen, müssen Sie nur das Dim-Schlüsselwort ändern - ObjNamIdx verweist weiter auf jedes Element in dem Array, und ObjNamSiz definiert die neue Array-Größe. Die C-Spezifikationen des Programms sollten beim Überprüfen des Wertes für ObjNamIdx immer ObjNamSiz verwenden, um Array-Indexfehler zu vermeiden. Gehen Sie dabei folgendermaßen vor:

```
C Eval ObjNamIdx = ObjNamIdx + 1 C If ObjNamIdx > Ø And C ObjNamIdx = ObjNamSiz C  
Eval ObjNamAry( ObjNamIdx ) C =ObjNam C EndIf
```

## Zusammensetzen von Programmen mit Teildatei-Kopien

Teildatei-Kopien sind bei der Modularisierung von Code auf der Source-Ebene praktisch. Mit Hilfe der Teildatei-Kopien können Sie standardmäßige Datendefinitionen zur Verfügung stellen. Außerdem lassen sich mit Teildatei-Kopien die Prototypen und die erforderlichen Definitionen für die Prozeduren in Ihren Service-Programmen definieren. Diese Tips enthalten Beispiele für beide Arten von Teildatei-Kopien. Das Beispiel unter „Vereinfachte Datendefinitionen“ (s.o.) zeigt, wie mit der Teildatei-Kopie StdTypH (Abbildung 1) Standarddatentypen definiert werden, die in beliebigen Programmen verwendet werden können. V3R7 RPG IV unterstützt verschachtelte Teildatei-Kopien, d.h. Teildatei-Kopien, die wiederum Kopier-Anweisungen für andere Teildatei-Kopien enthalten.

Sie müssen sich als Abonnent anmelden um den hier fehlenden Teil des Inhalts zu sehen. Bitte [Login](#) für Zugriff.

Noch nicht Abonnent? [Sonderaktion nutzen](#).

- [7 Euro/Monat NEWSabo digital - sofort zugreifen & online bezahlen.](#)
- [13,5 Euro/Monat NEWSabo plus inkl. 5x Logins & Print-Ausgaben - sofort zugreifen & per Firmen-Rechnung bezahlen.](#)