

# Der Entwickler 04/1999

## Auf dem Weg in's neue Jahrtausend - Der Millenium-Bug: Was tun? von Henry Wolf und Andreas Tengicki

Unternehmen und vor allem zahlreiche Entwickler stehen zur Zeit vor der Aufgabe, ihre Systeme und die darauf laufende Software, nicht nur für die Einführung des EURO, vor allem hinsichtlich ihrer zuverlässigen Lauffähigkeit im neuen Jahrtausend fit zu machen. Dabei sind Hardwarehersteller ebenso gefordert wie die Softwareentwickler.

In der zurückliegenden Zeit wurde in allen Bereichen der Datenverarbeitung nur unzureichend an den Wechsel in ein neues Jahrtausend gedacht. Bei der Implementierung von Hard- und Software wurde, wie jeder weiß, mit einem 2-stelligen Datumsformat ( "99", "01" ) gearbeitet. Hinsichtlich der Hardware ist dieses Problem durch Ersatzbeschaffung relativ problemlos zu lösen. Ob die Hardware Jahr-2000-fähig ist, gibt entweder der Hersteller an oder läßt sich durch "Vordrehen" der Uhr in entsprechenden Testumgebungen festzustellen.

Bei Software existiert dieser Lösungsweg im allgemeinen nicht. Hersteller von Software, die ihre Kunden über Wartungsverträge betreuen, sind gefordert, Software mit entsprechender Funktionalität zur Verfügung zu stellen.

Bei der Prüfung von System - und Anwendungssoftware sind prinzipiell folgende Schwerpunkte zu setzen:

1. Ist die Systemplattform, hierbei insbesondere das Betriebssystem, Jahr - 2000 - fähig?
2. Wurde in der Implementierung der Software ein vierstelliges Datumsformat verwendet?
3. Wie verhalten sich alle zeitkritischen Anfragen an Datenbanksoftware bzw. an die Systemsoftware: Liefern Sie ein korrektes Ergebnis ( z.B. Interbase, Oracle, Sybase )?
4. Sind die Schnittstellen zu anderen Anwendungen oder Systemen Jahr-2000-fähig ( z. B. BDE, VisiBroker, ApplikationServer )?

Bei der Prüfung ist immer die Jahr-2000-Fähigkeit des Gesamtsystems im Auge zu behalten.

### 1. Systemsoftware

Nicht nur in Großrechnerumgebungen, auch im Bereich der "Microsoft-Welt" müssen die Betriebssysteme und darauf aufsetzende Komponenten fit gemacht werden. So erschien im vergangenen Jahr Windows98 und die allgemeine Erwartung war, mit dem Umstieg sei man fit für den Jahrtausendwechsel. Aber auch hier wurde leider nicht an alles gedacht. Microsoft stellt für alle seine Betriebssysteme Y2K-Patches zur Verfügung.

Betriebssystem	Y2K-Patch
Windows NT 4	Service-Pack 5 und 6 auf der <i>Entwickler-CD</i>
Windows9	Y2K-Patch unter <a href="http://windowsupdate.microsoft.com/">http://windowsupdate.microsoft.com/</a>
Windows98	Y2K-Patch unter <a href="http://windowsupdate.microsoft.com/">http://windowsupdate.microsoft.com/</a> bzw. CD
Windows3.1	Y2K-Patch unter <a href="http://windowsupdate.microsoft.com/">http://windowsupdate.microsoft.com/</a> bzw. CD

Eines sollte man seinen Kunden in allen Fällen empfehlen: Stellen Sie in der Windows-Systemsteuerung auf 4-stelliges Datum um. Damit lassen sich viele Probleme einfach lösen, da unter Windows das Datum über die API ermittelt und nur ein zweistelliges Jahr zurückgegeben wird, wenn nach dem Kurzformat gefragt wird. Stellt man dieses auf 4 Stellen, wird es ebenfalls entsprechend zurückgeliefert.

Auch andere Hersteller von Systemsoftware bieten im allgemeinen Patches oder Updates für den Jahrtausendwechsel an. So stellt z.B. die Firma Oracle allen Kunden, auch denen ohne üblichen Wartungsvertrag die aktuelle, Jahr-2000-fähige Version kostenfrei zur Verfügung.

Die Unix-Betriebssysteme, auch in der Intel-Welt, arbeiten schon seit geraumer Zeit mit 4-stelligen Datumsformaten, das z.B. unter SCO bzw. Linux primär die Anwendungssoftware zu prüfen und zu bearbeiten ist.

### 2. Entwicklungswerkzeuge

Ist die prinzipielle Y-2000-Fähigkeit des Betriebssystems hergestellt, gilt es die Entwicklungswerkzeuge näher zu betrachten. Auch hier gilt, daß viele Produkte entweder gar nicht oder nur unter Beachtung und

Anwendung besonderer Kniffe für die Entwicklung von Jahr-2000-fähiger Software geeignet sind. Insbesondere bei 16 Bit Windows- bzw. DOS - Entwicklungswerkzeugen halten sich die Hersteller sowohl mit Angaben zur Jahr-2000-Fähigkeit als auch mit Patches für die entsprechende Software sehr zurück. Einzige Chance ist in solchen Fällen die konsequente Nutzung des 4-stelligen Datumsformates, sowie die bereits erwähnte Umstellung der Kurzdarstellung des Datums auf 4-stellige Jahresangaben. Ansonsten sollte die Software in jedem Falle auf die 32 Bit-Plattform und dabei auf die neuesten Werkzeugversionen umgestellt werden.

Im Folgenden sollen Probleme bei der Nutzung von 32 Bit-Werkzeugen von borland.com/Inprise etwas näher betrachtet werden.

Entwicklungsumgebung	Jahr-2000-Fähigkeit
BDE 16 Bit	Upgrade auf 32 Bit empfohlen
BDE 32 Bit	Y2K-Ready unter Beachtung bestimmter Hinweise und Nutzung neuester Versionen
Delphi 1	Upgrade auf Delphi 4 empfohlen
Delphi 2	Upgrade auf Delphi 4 empfohlen
Delphi 3	Y2K-Ready unter Beachtung bestimmter Hinweise
Delphi 4	Y2K-Ready unter Beachtung bestimmter Hinweise
C++ Builder 1	Upgrade auf C++-Builder 4 empfohlen
C++ Builder 3	Y2K-Ready unter Beachtung bestimmter Hinweise
C++ Builder 4	Y2K-Ready unter Beachtung bestimmter Hinweise
Borland C++ 5	Y2K-Ready unter Beachtung bestimmter Hinweise
JBuilder 1	Y2K-Ready unter Beachtung bestimmter Hinweise
JBuilder 2	Y2K-Ready unter Beachtung bestimmter Hinweise
DBase für DOS	Upgrade auf Visual dBase für Windows 5.6 empfohlen
Visual dBase 16 Bit	Y2K-Ready unter Beachtung bestimmter Hinweise
Visual dBase 32 Bit	Y2K-Ready unter Beachtung bestimmter Hinweise
Paradox	Version 7 Y2K-Ready unter Beachtung bestimmter Hinweise, Bei allen anderen Versionen ein Upgrade auf Version 7 empfohlen

### 3. BDE

Datenbankzugriffe, ob SQL-Server oder Desktopdatenbank, erfolgen bei den meisten Inprise-Werkzeugen über die BDE. Die Jahr-2000-Fähigkeit dieser Schnittstellensoftware zur Datenbasis der Anwendungen ist somit die Grundlage der Jahr-2000-Fähigkeit all dieser Werkzeuge und der damit erstellten Applikationen. Hierbei ist ausschließlich die aktuelle Version 5.01 Jahr-2000-fähig.

Bsp.: `SELECT 1/1/32 FROM DUAL`

BDE-Version < 5.01 : Ergebnis 1/1/32

BDE-Version 5.01 : Ergebnis 1/1/2032.

Zweistellige Datumsformate werden von der BDE 5.01 als Werte zwischen 1950 und 2049 interpretiert. Die BDE gibt ab dieser Version prinzipiell ein 4-stelliges Jahresformat zurück, wenn der BDE-Parameter *Fourdigityear* auf *True* gesetzt wurde. Als weiterer BDE-Parameter wurde *Yearbiased* eingeführt, welcher bestimmt, ob zu einem 2-stelligen Datumsformat 1900 hinzuaddiert werden soll. Ist dieser *true* gesetzt, wird der 01.01.96 als 01.01.1996 interpretiert. Ist dieser Parameter dagegen *false*, wird es als 01.01.0096 interpretiert.

Des Weiteren wurden einige DB-spezifische Probleme mit zeitkritischen Anfragen ( z.B. DB/2, Oracle ) in dieser Version behoben.

### 4. Delphi

Um Delphi-Anwendungen in einfacher Form Y2K-Ready erstellen zu können, wurde in Delphi die Variable *TwoDigitYearCenturyWindow* eingeführt. Diese Variable erzeugt ein Jahrhundertfenster für 2-stellige Jahresangaben. Weist man dieser Variable den Wert 80 zu, werden alle Datumsanfragen im 2-stelligen Format 80 Jahre rückwärts und 20 Jahre vorwärts interpretiert. Diese Möglichkeit gestattet es, relativ einfach

Software Y2K-Ready zu gestalten. Etwas ähnliches wurde für die IDE zur Verfügung gestellt. Über den Parameter:

"HKEY\_CURRENTUSER\Software\Borland\Delphi\4.x\Globals\TwoDigitYearCenturyWindow" wird die Variable *TwoDigitYearCenturyWindow* für alle Programme der Entwicklungsumgebung von Delphi ( SQL-Explorer, IDE ) gesetzt.

Dennoch existieren - wie immer - noch eine Reihe von kleinen, lästigen Problemchen. So arbeitet die Komponente *TDateTimePicker* im Jahr 2000 nicht mehr korrekt.

Hierbei schlägt ein Betriebssystem-Bug von Windows durch. Der 29.02.2000, der Y2K-Schaltjahr-Bug, wird durch die Nutzung der neuesten Common-Control DLL von Microsoft behoben ( *ComCTL32.DLL* ). Viele andere Komponenten, sowohl Freeware als auch kommerzielle Komponenten, welche mit dem Datum arbeiten, haben mit dem selben Bug Probleme. Hinsichtlich dieser und anderer Komponenten gilt es, die Y-2000-Fähigkeit jeweils zu überprüfen.

Auf jeden Fall sollte man das aktuelle Service-Pack 3 für Delphi 4 einspielen ( auf der aktuellen *Entwickler-CD* ), da mit dieser Software unter anderem die BDE 5.01 zur Verfügung gestellt wird und kleinere Probleme innerhalb der BDE behoben werden.

## **5. C++-Builder / Borland C++**

Für Borland C++ 5.02 ( *bc5y2k.exe* ) und den C++-Builder 3 ( *bc3Y2K.exe* ) werden von Inprise Patches zur Herstellung der Jahr-2000- Fähigkeit der OWL zur Verfügung gestellt und hierbei gleich einige Programme ( *TDump* u. a. ) dieser Werkzeuge mit bearbeitet. Hinsichtlich der Nutzung von VCL-Komponenten gelten alle für Delphi gültigen Aussagen auch für den C++Builder.

Ein bisher ungelöstes Problem bei den C++-Werkzeugen ist, das ANSI-C - Funktionen im Jahr 2038 falsche Ergebnisse zurückgegeben werden. Hierbei wird vom Hersteller auf einen Bug im Betriebssystem Win32 verwiesen und der Anwender auf zukünftige Versionen getröstet. Auch für die C++-Werkzeuge gilt: arbeitet man mit der BDE, sollte man sich die aktuelle Version der BDE beschaffen und in seine Software integrieren.

## **6. Visual dBase**

Visual dBASE 5.6 und Visual dBASE 7 sind prinzipiell Jahr 2000-fähig. Zur Interpretation des 2-stelligen Datums wurde adäquat zu Delphi der Befehl SET EPOCH eingeführt. Implementiert man den Befehl SET EPOCH TO 1950 werden alle 2-stelligen Datumsangaben als Werte zwischen 1950 und 2049 interpretiert. Weiterhin wird durch den Befehl SET CENTURY ON die Nutzung des 4-stelligen Jahresformates für die Visualisierung und Speicherung erzwungen.

## **7. Paradox**

Paradox bzw. ObjectPal interpretieren 2-stellige Datumsangaben immer entsprechend dem aktuellen Jahrhundert. Gibt man am 31.12.1999 das Datum 31.12.99 ein, wird es immer als 31.12.1999 interpretiert. Gibt man am 01.01.2000 das Datum 31.12.99 ein, wird es als 31.12.2099 interpretiert. Inwieweit man dies Y2K-Ready nennt, bleibt dem Hersteller vorbehalten...

## **8. JBuilder 1/2**

Bei der Implementierung von JAVA-Anwendungen sollte man prinzipiell das 4-stellige Jahresformat nutzen. Zwar arbeiten die Datumsobjekte in *java.sql.Date* korrekt, jedoch entstehen an zahlreichen anderen Punkten ( Beans u.a. ) Probleme. Nutzt man das 4-stellige Datumsformat sind keine Probleme zu erwarten. Jedoch sollte man JBuilder 1 auf die Version 1.01 umstellen nur diese Produkte explizit vom Hersteller hinsichtlich der Jahr-2000-Fähigkeit überprüft wurden.

## **9. Middleware ( MIDAS/CORBA/Application-Server )**

Die Y-2000-Fähigkeit der von Inprise angebotene Middleware ist laut Herstellerangaben abhängig von der verwendeten Betriebssystemplattform und der BDE. Werden also alle notwendigen Patches verwendet, besteht auch hier Y-2000-Fähigkeit.

## **10. Geschichte des Millenium-Bugs**

Ist nun sichergestellt, daß Betriebssystem und Entwicklungsumgebung Jahr-2000-fähig sind, muß man noch seine eigene Software untersuchen. Dabei kann man sich eigentlich zu der Aussage hinreißen lassen, daß, sofern ein modernes Entwicklungswerkzeug eingesetzt wurde, keine Probleme zu erwarten sind. Um dies zu erläutern, hilft ein Blick in die Ursachen des Millenium-Bugs und damit in die Geschichte der Softwareentwicklung.

Als man begann, betriebswirtschaftliche Lösungen zu schreiben, war der Leistungsumfang, den heute jeder Billig-PC aufweist, sehr teuren, aufwendigen Computern vorbehalten bzw. wurde nicht einmal von diesen

erreicht. Das bedeutet insbesondere, daß Platten- und Hauptspeicherplatz sehr teuer war. Es wurde an jedem Byte gespart, folglich auch das Datumsformat klein gehalten.

Des weiteren existierten in der damals üblichsten Programmiersprache Cobol keine Mechanismen zur Darstellung von Zahlen in Mantissen-Schreibweise oder Datumswerten. D.h. eine zwölfstellige Zahl war

```
01 ZAHL12 PIC 999999999999999999
```

und wenn man diesen Wert im Ausdruck mit Tausender-Punkten haben wollte, formatierte man

```
01 ZAHL12P PIC 999.999.999.999
```

Ein Datum war dann auch schnell dargestellt und die Uhrzeit dazu natürlich auf gleichem Wege extra. Dabei kann die Darstellung mit und ohne Trennzeichen erfolgen.

```
01 MYDATUM PIC 9999999
```

```
01 MYDATUMT PIC 99.99.99
```

```
01 MYZEIT PIC 9999
```

```
01 MYZEITT PIC 99:99
```

Datumsberechnungen sind mit einem solchen, quasi alphanumerischen Datumsformat nicht ganz so einfach zu bewerkstelligen. Es bedarf dazu einiger Handarbeit und der Interpretation der Daten. Dazu ging man z. B. zur Bestimmung der Differenz in Tagen wie folgt vor.

Das Beispiel ist schematisch und rechnet mit einem theoretischen Jahr von 360 Tagen und Monaten zu 30 Tagen. Auch daran sollten die Probleme deutlich werden, ohne daß die Beispiele unnötig verkompliziert werden.

```
01 MYDATUM1 PIC 9999999
```

```
01 MADATUM1E REDEFINES MYDATUM
```

```
05 YY1 PIC 99
```

```
05 MM1 PIC 99
```

```
05 DD1 PIC 99
```

```
DIFF = (yy1*360+mm1*12+dd1)-(yy2*360+mm2*12+dd2)
```

Diese Berechnung schlägt natürlich dann fehl, wenn vom Jahr 2000 (00) das Jahr 1999 (99) abgezogen wird.

Je nach Aufbau der Software sind Befehle dieser Art in der ganzen Software verstreut und müssen dezentral korrigiert werden. Dies ist der eigentliche klassische Millenium-Bug. D. h. an vielen schlecht bestimmbar Stellen einer Software wird mit dem Datum fehlerhaft umgegangen. Die kritischen Stellen sind insbesondere deshalb schwer zu bestimmen, weil sich die Darstellung einer Datumsvariablen ( PIC 9999999 ) nicht von einer "normalen" 6-stelligen Zahl unterscheidet.

Wurde bei der Implementierung der COBOL-Software schon etwas abstrahiert, so wurde die Datumsberechnung in speziellen "Paragraphen" realisiert, die dann aufgerufen wurden.

```
MOVE MYDATUM1 TO SUBDATUM1
```

```
MOVE MYDATUM2 TO SUBDATUM2
```

```
PERFORM DATUMSDIFFERENZ
```

```
MOVE SUBDIFF TO DIFF
```

Diese Art der Realisierung ermöglicht die zentrale Korrektur der Berechnung, benötigt aber nach wie vor eine Änderung und Neu-Kompilierung aller beteiligten Programme, in denen entsprechende Paragraphen zu korrigieren sind. Das Problem der Lokalisierung der problematischen Zeilen wird dann etwas kleiner, wenn *alle* Datumsberechnungen durch Paragraphen zentralisiert wurden.

Optimal, doch meist nicht realisiert, wäre der Aufruf eines echten Unterprogramms unter COBOL zur Datumsberechnung. Denn dieses wird extra übersetzt und einzeln gelinkt. Da aber der Aufruf eines solchen "echten" Unterprogramms ähnlich dem Aufruf einer Windows DLL ist und den Computer deutlich mehr belastet hat, wurde darauf meist verzichtet: Ressourcen waren, wie schon erwähnt, damals in der Steinzeit grundsätzlich knapp.

## 11. Realisierung von klassischen Jahr 2000 Projekten

Zur Behebung des Millennium-Bugs können nun alle Datumsfelder auf 4-stelliges Datum umgestellt werden ( große Lösung ). Oder aber, alle Datumsberechnungen werden angepaßt, dabei bleiben Oberflächen-, Listen- und Dateidarstellung eines Datumsformates gleich ( interpretative Lösung ). Risikofreudige Projektmanager können auch versuchen, nur die wirklich gefährlichen Stellen zu korrigieren und hoffen, dabei alle zu erwischen.

Selbst durchgeführt haben und empfehlen können wir nur die interpretative Lösung. Wenn genügend Zeit vorhanden ist, bleibt natürlich auch noch die Neuentwicklung unter Beachtung aktueller Konzepte und Werkzeuge.

Die interpretative Lösung in einer COBOL oder COBOL-ähnlichen Umgebung bedeutet also, alle Datumsberechnungen zu suchen, zu finden und zu korrigieren. Alle Eingabemasken und Ausgabelisten bleiben unverändert, die Dateidarstellung ändert sich nicht. Dies ist also viel mehr ein logisches als ein intellektuelles Problem. Man kann nun alle Sourcen manuell durchsuchen und durch möglichst viel Personal schnell zum Ziel kommen ( Ameisentaktik ) oder aber geeignete Werkzeuge zum Analysieren und ggf. auch Korrigieren einsetzen. Dabei hängt die Eignung eines Werkzeuges sehr stark von den individuellen Anforderungen des jeweiligen Projektes ab.

## 12. Parallelen zu heutigen Entwicklungsumgebungen

In heutigen Entwicklungsumgebungen stellt sich die Frage meist nicht mehr, vorausgesetzt man verwendet die am Anfang des Artikels genannten Versionen. Ein Datum wird meist als Anzahl Tage ab dem 1.1. im Jahre x gespeichert. Im "Teil hinter dem Komma" verbirgt sich dann die Uhrzeit als Bruchteil eines Tages. Auch diese Art der Darstellung wird problematisch, wenn der Speicherplatz zur Darstellung der Tage seit dem 1.1. nicht mehr ausreicht ( z. B. ANSI-C s. o. ). Nur tritt dieses Problem nicht Ende dieses Jahres auf, sondern Ende des Jahres 2037. Wir haben also noch etwas Zeit und können dieses Problem getrost unseren Nachfahren überlassen.

Eine Differenzberechnung bei einem auf Gleitkommawerten basierendem Datumsformat, erfolgt einfach über  
diff := datum1-datum2;

und diese Berechnung bleibt auch über die Jahre 2000 und 2038 hinaus richtig.

Aber wo treten jetzt Probleme auf? Probleme treten dann auf, wenn das Jahr als Text formatiert und dann z. B. von einer Datenbank interpretiert wird.

```
sql := 'Select * from daten where datum <' + FormatDatetime('dd.mm.yy',dValue);
```

Jetzt hängt es weitgehend von der Datenbank ab, ob sie mit diesem Wert richtig umgeht. Bei Paradox ist dies beispielsweise nicht der Fall ( s.o. ).

Die zweistellige Formatierung des Datums kann teilweise auch impliziet geschehen. Dabei hängt es von den Windowseinstellungen ab, wie das Datum formatiert wird. Daher am Anfang des Artikels der Rat, ab nächstem Jahr grundsätzlich mit 4-stelligem Systemdatum zu arbeiten.

```
sql := 'Select * from daten where datum <' + DateToStr(dValue);
```

Aber können Sie gewährleisten, daß ihre Kunden alle das richtige Systemdatum einstellen? Sie können die Probleme vermeiden, indem sie alle Textdarstellungen eines Datums vierstellig machen. Dann kann es auch nicht fehlinterpretiert werden.

```
sql := 'Select * from daten where datum <' + FormatDateTime('dd.mm.yyyy',dValue);
```

## 13. Schlußbemerkung

Die Brisanz des Jahr-2000-Problems hängt also ganz von den speziellen Gegebenheiten Ihres Systems ab.

Als Anwender sollten Sie sich die folgenden Fragen stellen. Haben Sie Ihr Betriebssystem auf den richtigen Stand gebracht? Haben Sie das Systemdatum auf vier Stellen gestellt? Verwenden Sie COBOL oder Paradox geschriebene Software? Oder verwenden sie mit aktuellen Werkzeugen auf dem aktuellen Stand entwickelte Software? Wenn sie sich nicht sicher sind, fragen Sie den Hersteller oder stellen Sie, wenn es möglich ist, das Systemdatum probeweise um. Das geht natürlich nicht, wenn das Datum Ihres Rechners steuernde Eigenschaften hat.

Als Anbieter bzw. Hersteller von Software gehen Sie den sicheren Weg, wenn Sie Ihr Betriebssystem und Ihre Entwicklungsumgebung auf den aktuellen Stand bringen. Informieren Sie Ihre Kunden über die optimalen Einstellungen des Systems. Überprüfen Sie diese Einstellungen, wenn möglich, mit Ihrer Software. Verwenden Sie in Ihrer Software bei jeder alphanumerischen Darstellung eines Datums explizit die

vierstellige Darstellung.

Bei der rechtzeitigen Beachtung aller Hinweise in diesem Artikel sollten Sie Ende des Jahres in Ruhe feiern können und am Montag, dem 3. Januar 2000 entspannt nach Ihrem PC sehen können.