Konvertierung in 4D v17

Willkommen zum Handbuch "Konvertierung in 4D v17". Es beschreibt verschiedene Punkte, die Sie vor, während und nach dem Konvertieren einer 4D v16 Anwendung in 4D v17 beachten müssen.

Das Konvertieren von 4D v16 Anwendungen sollte ein einfacher Vorgang sein. Damit alles reibungslos läuft, geben wir Ihnen einige Empfehlungen im Abschnitt "**Vorgehensweise beim Konvertieren**". Aber auch nach der Konvertierung sind noch ein paar Dinge wichtig, wie **"Neue Kompatibilitätsoptionen"** und "**Geändertes Verhalten (global)**", die sowohl für die Anwendung als auch für die 4D Befehle gelten und die Sie kennen sollten, um die neuen Features in 4D v17 bestmöglich einzusetzen.

Am Ende gibt es eine Liste, die **Überholte Funktionalitäten in 4D v17** zeigt. Entwickler sollten einen Blick darauf werfen, um einschätzen zu können, wieviel Zeit zum Einrichten neuer Features benötigt wird.

Hinweis: Einige der hier aufgeführten Änderungen wurden im R-Release Zyklus von 4D v16 eingeführt.

Beim Konvertieren früherer oder viel älterer Anwendungen müssen oft Zwischenversionen verwendet werden. Und es gibt weitere Punkte zu beachten, die in der Dokumentation zum Konvertieren der früheren Versionen beschrieben werden:

- **4D v16**: "Konvertierung in 4D v16" und "Überholte Funktionalitäten in 4D v16 und höher".
- **4D v15**: "Konvertierung in 4D v15" und "Überholte und entfernte Funktionalitäten in 4D v15 und höher".
- **4D v14**: "Konvertierung in 4D v14" und "Überholte und entfernte Funktionalitäten 4D v14 und höher".
- **4D v13**: "Konvertierung in 4D v13" und "Deprecated features 4D v13 and higher".
- **4D v12**: "Deprecated features 4D v12 and higher" (für diese Version gab es kein Konvertierungsdokument)
- **4D v11**: "Conversion to 4D v11 SQL".

Vorgehensweise beim Konvertieren

Was ist vor der Konvertierung zu tun?

- Zum Durchführen der Konvertierung müssen Sie eine **interpretierte Version** der Anwendung (xxxx.4DB für die Struktur, xxxx.4DD für die Daten, .RSR und .4DR für Versionen älter als 4D v11) und das Designer Kennwort haben.
- Machen Sie vor dem Konvertieren **eine Kopie Ihrer Anwendung**.
- Führen Sie eine Syntaxprüfung durch. Diese Überprüfung ist hilfreich, selbst wenn Sie Ihre Anwendung nicht kompilieren wollen, denn sie kann auf mögliche Fehler hinweisen.
- Rufen Sie das **Maintenance und Security-Center** auf, um Ihre Struktur und Daten zu prüfen und ggf. zu reparieren.
- Prüfen Sie mit dem Befehl **GET PICTURE FORMATS**, ob Sie PICT Bilder haben und konvertieren sie über den Befehl **CONVERT PICTURE**.
  Seit 4D v14 werden QuickTime Codecs standardmäßig nicht mehr unterstützt. Zur Wahrung der Kompatibilität können Sie diese in 32-bit Versionen über den Selector QuickTime Support für den Befehl **SET DATABASE PARAMETER** noch reaktivieren. Danach ist ein Neustart erforderlich. Beachten Sie jedoch, dass die Unterstützung von QuickTime in zukünftigen 4D Version gänzlich wegfällt. Weitere Informationen zum Konvertieren veralteter Bildformate finden Sie unter **Bildtyp konvertieren**.
- (Optional) Soll Backup/Journal verwendet werden, müssen Sie Primärschlüssel einrichten (gilt bereits seit Version 14). Weitere Informationen dazu finden Sie im Abschnitt **Primärschlüssel** des Handbuchs *4D Designmodus*. Wir raten dringend zu Primärschlüsseln. Sie lassen sich aber auch nach der Konvertierung einrichten.
- Ab Version 13.5 müssen Felder mit der Eigenschaft **Einmalig** indiziert sein. Sie können nicht mehr Datensätze mit einem einmaligen Feld erstellen bzw. ändern, das nicht indiziert ist: Beim Versuch, diesen Datensatz zu sichern, wird ein Fehler generiert (**-9998** Dieser einmalige Datensatz ist bereits vorhanden, **1088** Index ist ungültig oder fehlt). Wie Sie fehlende Indizes oder eine Datei auf der Festplatte mit allen nicht-indizierten Feldern erstellen können, wird im **Anhang: Hilfsmittel beim Konvertieren** beschrieben.

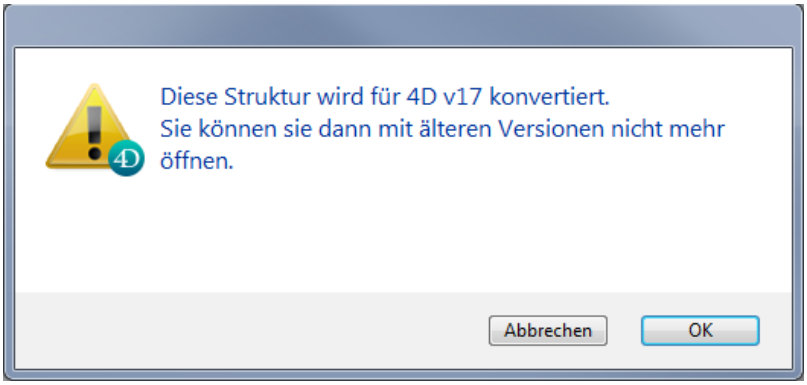
Anwendung konvertieren

Anwendungen, die in Version 16 von 4D oder 4D Server (sowie in v11, v12, v13, v14 und v15) erstellt wurden, sind mit 4D Version 17 (Struktur- und Dateidatei) kompatibel. Sie können jede interpretierte Strukturdatei konvertieren. Dazu starten

Sie einfach 4D v17 und öffnen Ihre Strukturdatei (xxx.4DB Datei) im interpretierten Modus.

- **Strukturdatei konvertieren (.4DB)**

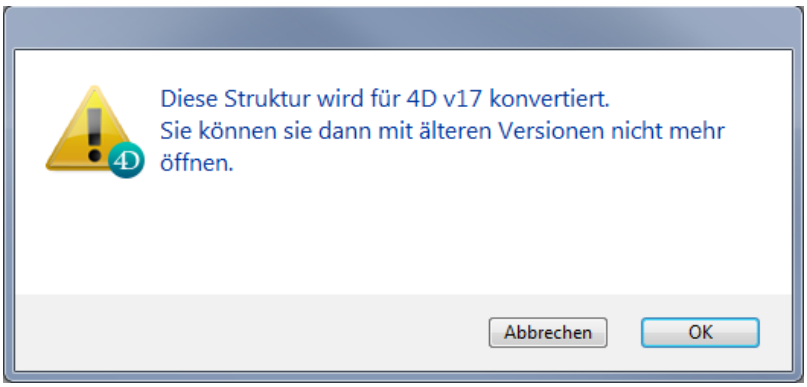
Auf dem Bildschirm erscheint die Meldung, dass die Strukturdatei konvertiert wird:



Klicken Sie auf die Schaltfläche OK, wird Ihre Strukturdatei in 4D v17 konvertiert und lässt sich nicht mehr mit einer älteren Version öffnen.

- **Datendatei konvertieren (.4DD)**

Datendateien für Anwendungen in 4D v14 und älter müssen konvertiert werden. In diesem Fall erscheint ein zweites Dialogfenster:

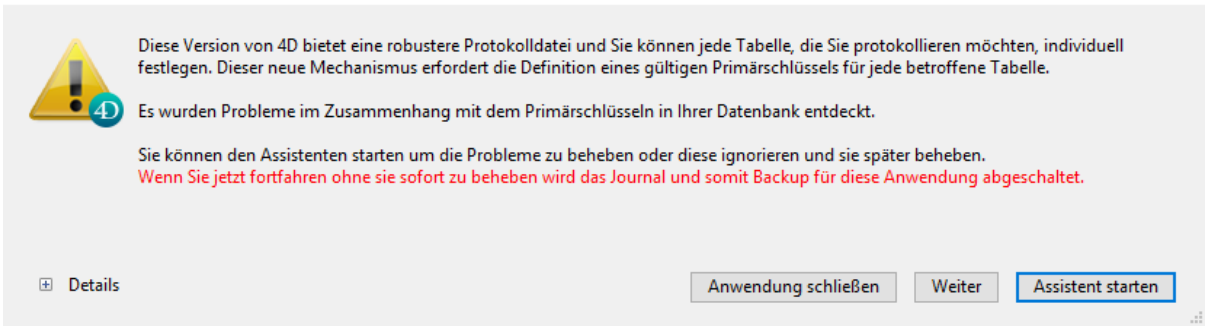


Diese Datendatei wird auch in Version 17 konvertiert, lässt sich aber weiterhin mit 4D v15 oder 4D v16 öffnen.

Datendateien für Anwendungen in 4D v15 oder 4D v16 müssen nicht konvertiert werden.

- **Primärschlüssel für Logbuch einrichten**

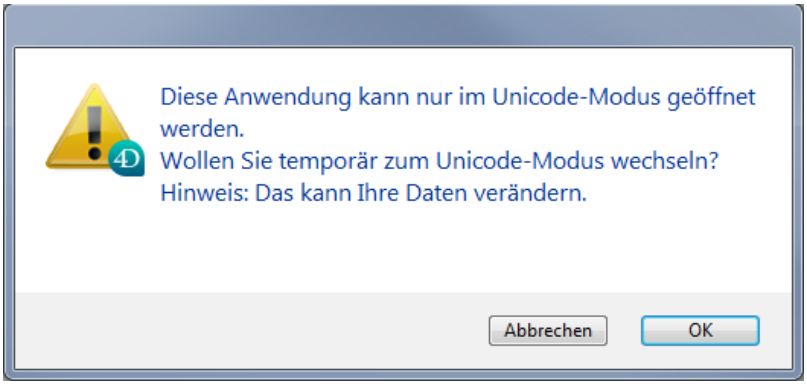
Sind keine Primärschlüssel eingerichtet, erscheint folgendes Dialogfenster:



Es ist ratsam, Primärschlüssel einzurichten (Schaltfläche "Assistent starten"). Sie können diesen Schritt auch später ausführen (Schaltfläche "Weiter"). Weitere Informationen dazu finden Sie im Kapitel **Primärschlüssel-Assistent**.

- **Temporär in Unicode wechseln**

Öffnen Sie Ihre Anwendung in 4D v17 64-bit und ist Ihre Ausgangsversion nicht in Unicode, erscheint folgendes Dialogfenster:



Da Unicode die Geschwindigkeit deutlich steigert, empfehlen wir schon seit einigen Versionen, auf Unicode umzustellen. Klicken Sie auf die Schaltfläche **OK**, können Sie temporär in Unicode wechseln. Weitere Informationen dazu finden Sie im Kapitel **Seite Kompatibilität**.

Nach der Konvertierung

Verwenden Sie erneut das **Maintenance und Security Center** (MSC), um die Struktur und Daten zu prüfen und zu reparieren.

In gewohnter Weise erscheinen folgende Angaben zur **Struktur**:

- Verwaiste Methoden (__Orphan__xxxxx) erscheinen als **Warnungen** im Protokoll des MSC und lassen sich über den Explorer löschen (nach Prüfen, dass ihr Code nicht mehr nützlich ist)

- Auffinden doppelter Objektnamen: Formulare können keine doppelten Objektnamen enthalten. Sie erscheinen im Protokoll des MSC als **Warnungen**. Führen Sie eine Operation zum Reparieren der Anwendung aus, um diese Namen zu ändern (in diesem Fall sollten Sie die Programmierung von Objektnamen achten).

- Auffinden veralteter Bilder im PICT Format. Weitere Informationen dazu finden Sie im Abschnitt **Anwendung prüfen** des Kapitels MSC sowie im Abschnitt **Bilder in PICT Format**. Diese Warnungen können statische Bilder betreffen, aber auch Bilder, die in der Bildbibliothek oder in Formularobjekten gespeichert sind.
  Weitere Informationen zum Konvertieren veralteter Bildformate finden Sie unter **Bildtyp konvertieren**.

- **Unzulässige Zeichen in Namen (".", "[", und "]")**
  Seit 4D v16 R4 ist Verwenden von Punkten (.) bzw. eckigen Klammern ([ ]) in Namen für folgende Elemente problematisch:
  - Variablen
  - Tabellen
  - Feld
  - Projektmethoden

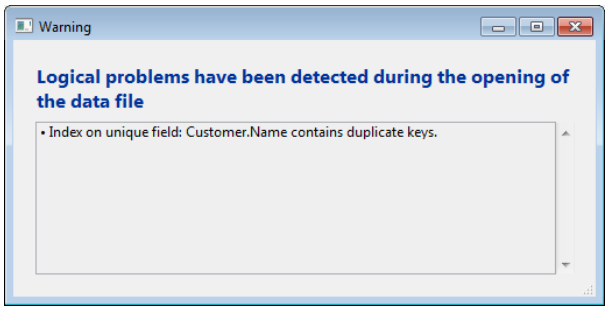
  Um Entwickler beim Anpassen ihres Code an diese Regel zu unterstützen, überprüft die Aktion **Anwendung prüfen** automatisch, ob in Namen von Variablen, Tabellen, Feldern und Methoden unpassende Zeichen enthalten.

Werden solche Zeichen gefunden, protokolliert das MSC diese Vorkommen und das Logbuch enthält die entsprechenden Warnungen:

In diesem Fall raten wir dringend, diese Elemente in Ihrer Anwendung umbenennen.

Angaben für die **Daten**:

- Auffinden von Dupletten in einmaligen Feldern:
  Es gibt zusätzliche Informationen: Beim Verwenden des MSC oder einem Befehl wie **VERIFY DATA FILE** zeigen die generierten Protokolle jetzt die Namen der betroffenen Tabellen und Felder und jeden doppelten Wert an.
  Hinweis: Bei der Dateneingabe zeigt das Dialogfenster mit der Fehlermeldung jetzt den Namen der dazugehörigen Tabelle bzw. des Feldes und den duplizierten Wert an. Auch der Befehl **GET LAST ERROR STACK** zeigt detaillierte Information zu doppelten Werten.
  Öffnet 4D eine Datendatei, wenn ein Index erstellt oder neu aufgebaut werden muss, werden doppelte Werte jetzt automatisch in den zugewiesenen Feldern mit der Eigenschaft "einmalig" ausfindig gemacht. In diesem Fall erscheint vor dem Öffnen der Anwendung eine spezifische Meldung, so dass der Benutzer Dupletten finden und entfernen kann:



Indizes erneut aufbauen

Beim Konvertieren von v16 auf v17 müssen die Indizes der Anwendung nicht neu aufgebaut werden (mit Ausnahme der japanischen 4D Version).

Beim Upgrade von einer älteren Version - insbesondere wenn ein Update der Unicode Library (ICU - International Components for Unicode) notwendig ist - werden alle Text- und Schlüsselwort-Indizes in 4D neu aufgebaut. Diese Operation wird automatisch beim ersten Öffnen der Anwendung durchgeführt und kann eine Weile dauern.

Hinweis: Wir haben ab 4D v16 den globalen Algorithmus zur Reindizierung der Daten in der Anwendung in hohem Maße optimiert. All seine Prozesse wurden gesichet und diese Operation kann jetzt doppelt so schnell sein. Eine globale Reindizierung ist z.B. nach Reparatur der Struktur erforderlich oder wenn die Datei .4dindx gelöscht wurde**.**

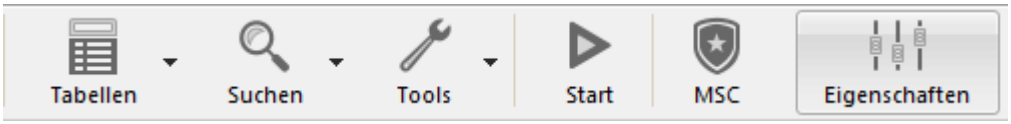
Hinweis zur Kompatibilität

Sie können eine Anwendung in Version 16 Rx mit einer Version v16.x und umgekehrt öffnen, dasselbe gilt für v17 Rx und v17.x. Jedoch funktioniert dann

Code, der die neuen Features verwendet, nicht und sollte deaktiviert werden. Ist eine Anwendung einmal in v17 konvertiert, lässt sie sich nicht mehr in 4D v16 öffnen.

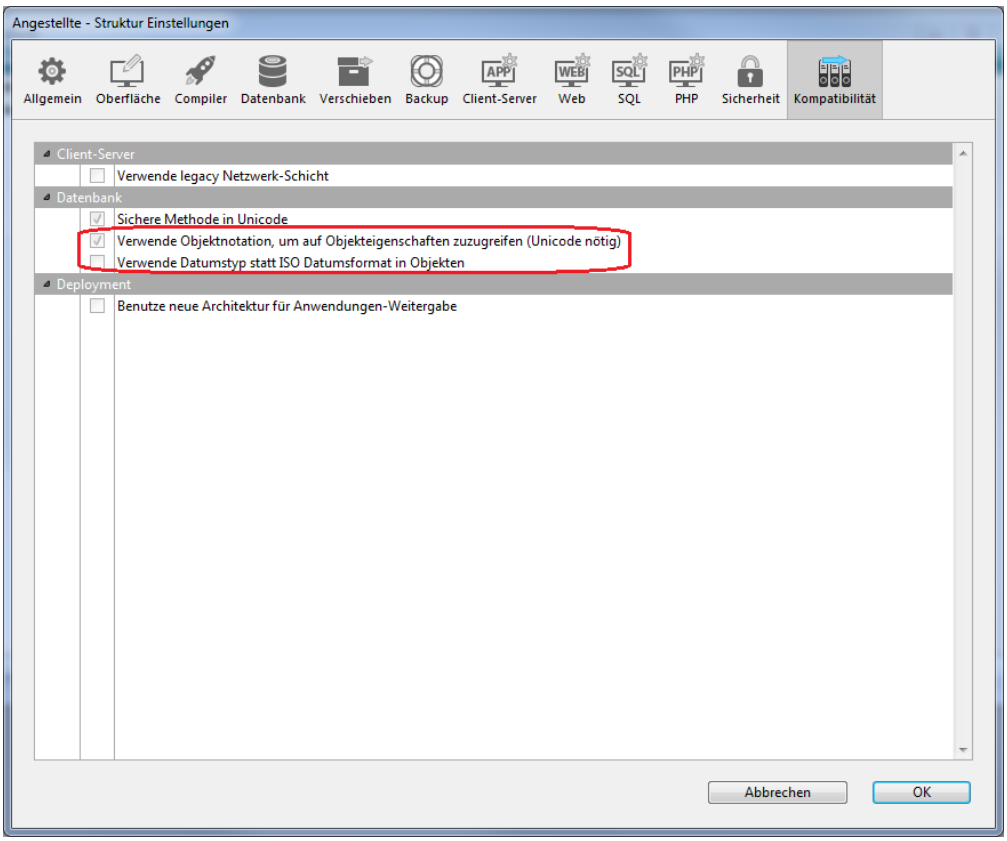
Dialogfenster Kompatibilität

Um dieses Dialogfenster zu öffnen, klicken Sie in der Werkzeugleiste am oberen Rand auf den Icon "Eigenschaften":



Dann auf die Registerkarte "Kompatibilität".

In v17 sind zwei neue Optionen dazugekommen:



Verwende Objektnotation, um auf Objekteigenschaften zuzugreifen

Um die Vorteile von Objektnotation zu nutzen (siehe Abschnitt **Objektnotation verwenden**), können Sie die Verwendung der Zeichen ".", "[" And "]" als Symbole für Objeknotation definieren, um auf tokenisierte Eigenschaften in **Namen von Variablen**, **Tabellen**, **Feldern** und **Projektmethoden** zuzugreifen.

Diese Option ist zwingend für Anwendungen, die mit einer Version vor 4D v17 erstellt wurden, da hier in den Elementnamen die Zeichen ".", "[" And "]" erlaubt waren. Das ist bei Verwenden von Objektnotation nicht mehr zulässig.

Warnung:

- Für diese Option muss die Anwendung in Unicode sein
- Um Entwickler beim Anpassen Ihrer Anwendung zu unterstützen, überprüft die Aktion **Anwendung prüfen** des MSC automatisch (siehe Abschnitt **Anwendung prüfen**), ob Ihr vorhandener Code konform ist. Sie sucht automatisch nach unpassenden Zeichen in Namen von Variablen, Tabellen, Feldern und Methoden und meldet dies als Fehler:

In diesem Fall müssen Sie diese Elemente in Ihrer Anwendung umbenennen.

Verwende Datumstyp statt ISO Datumsformat in Objekten

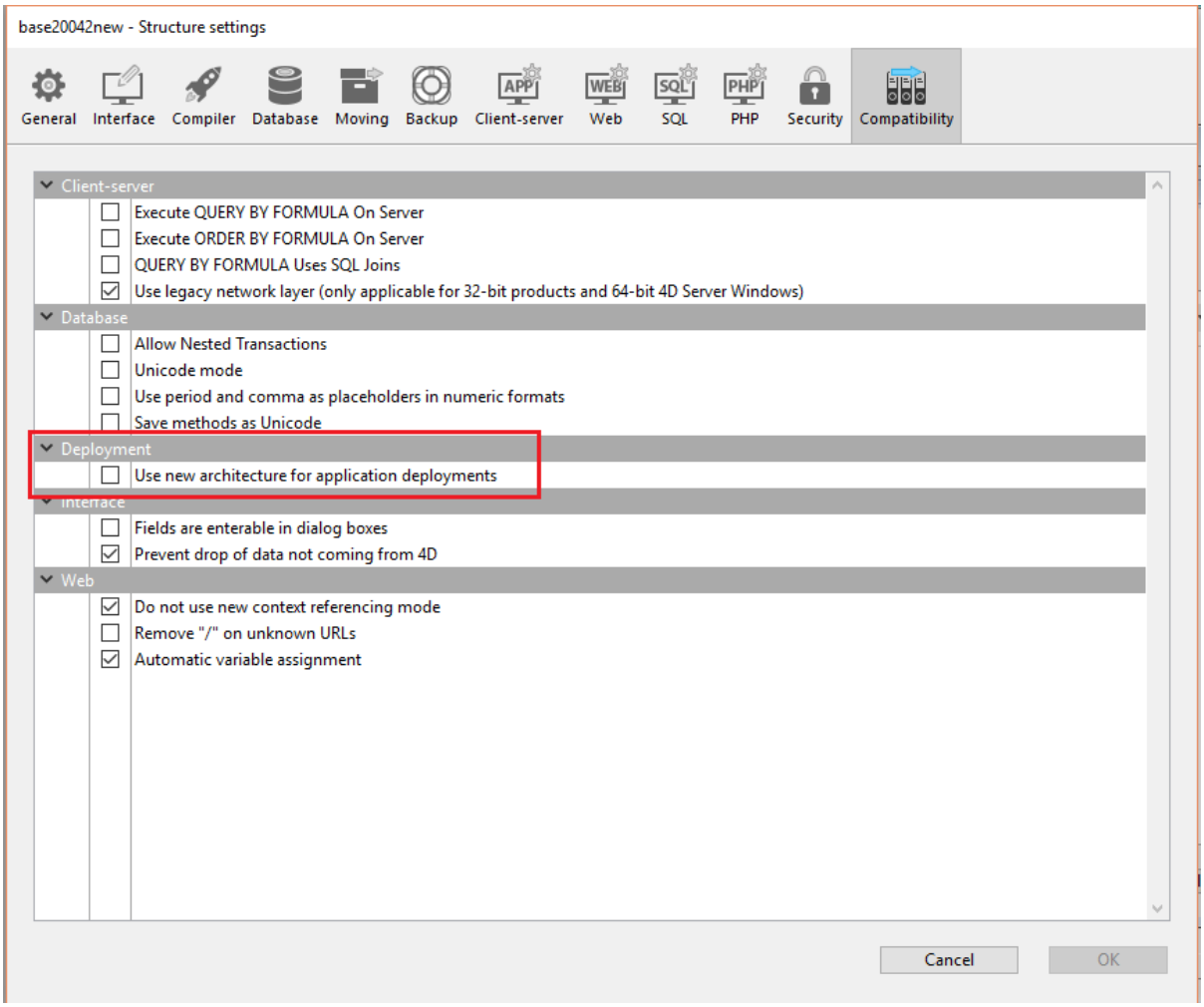
In einem Objekt lassen sich Datumsangaben jetzt als Datum und nicht als String definieren. Das macht die Verwendung von Datum in Objekten leichter und intuitiver. Die Befehle **OB SET** und **OB Get** können ohne die Konstante Is date arbeiten und Objektnotation lässt sich wie für jedes andere Objekt verwenden. Damit Sie Datum in Objekten einer konvertierten Anwendung benutzen können, markieren Sie die Option "Verwende Datumstyp statt ISO Datumsformat in Objekten". Siehe unter **Seite Kompatibilität**.

Entfernte Option

Seit 4D v16 ist die Option **Benutze 4DVAR Kommentare statt Klammern** auf der Seite **Kompatibilität** entfernt.

Alte Optionen für Kompatibilität

Je älter Ihre 4D Version ist, desto mehr Optionen erscheinen:



Weitere Informationen dazu finden Sie im Abschnitt **Seite Kompatibilität**.

Seite Kompatibilität

Die Seite Kompatibilität bietet Parameter zur Wahrung der Kompatibilität mit älteren 4D Versionen. Beachten Sie, dass eine Reihe der Optionen abhängig von der 4D Version angezeigt werden, mit der die Originaldatenbank erstellt wurde: 2004.x, v11, v12 usw. sowie die in dieser Datenbank geänderten Einstellungen.

Hinweis: Diese Seite erscheint nicht in Anwendungen, die mit der aktuellen 4D Version erstellt wurden (nicht-konvertierte Anwendungen).

- **Felder in Dialogfenstern eingebbar**: In früheren Versionen von 4D war es nicht möglich, Werte über Felder in Dialogfenstern einzugeben. Sie werden z.B. über den 4D Befehl **DIALOG** angezeigt. Diese Einschränkung ist seit 4D 2004 aufgehoben. Sie können das vorige Verhalten beibehalten, insbesondere wenn Ihre Datenbank in Dialogfenstern Felder zum Anzeigen von Daten verwendet. Diese Option ist für konvertierte Datenbanken, die mit einer Version vor 2004 erstellt wurden, standardmäßig aktiv, für Datenbanken, die in Version 2004 oder neuer erstellt wurden, inaktiv.
- **Optionsfelder nach Namen gruppieren**: In früheren Versionen von 4D war koordiniertes Verhalten einer Gruppe von Optionsfeldern möglich, wenn die zugewiesenen Variablen denselben Anfangsbuchstaben hatten, z.B. m_button1, m_button2, m_button3, etc.. (Koordiniert = Es lässt sich immer nur ein Optionsfeld in der Gruppe auswählen). Das wurde ab 4D 2004 wie folgt geändert: Ein Satz Optionsfelder arbeitet koordiniert, wenn er im Formulareditor gruppiert ist. Weitere Informationen dazu finden Sie im Abschnitt **Optionsfelder und Optionsbilder**.
  Dieser Modus gilt für Optionsfelder, 3D Optionsfelder und Optionsfelder vom Typ Bild.
  Aus Kompatibilitätsgründen wird in konvertierten Datenbanken standardmäßig der alte Modus beibehalten. Wollen Sie den neuen Modus verwenden, deaktivieren Sie diese Option. Datenbanken, die in Version 2004 oder neuer erstellt wurden, verwenden die neue Arbeitsweise.
- **Formular für jeden Datensatz während PRINT SELECTION** neu laden: In früheren Versionen von 4D wurde das Formular, welches über den 4D Befehl **PRINT SELECTION** beim Drucken verwendet wird, für jeden Datensatz erneut geladen. So ließen sich alle Objekteinstellungen, die der Entwickler unter Umständen per Programmierung verändert hatte, über das Formularereignis On printing detail erneut initialisieren.
  Diese Funktionsweise wurde ab 4D 2004 zur Optimierung der Performance aufgehoben. Der 4D Entwickler muss jetzt selbst die gewünschten Einstellungen in der Formularmethode reinitialisieren – das ist identisch mit der Arbeitsweise von Listenformularen mit dem Formularereignis On display detail. Über diese Option können Sie die bisherige Arbeitsweise beibehalten. Datenbanken, die in Version 2004 oder neuer erstellt wurden, verwenden die neue Arbeitsweise.
- **Verwende nicht neue Kontext-Referenzmethode**: Ist diese Option nicht markiert (Standardeinstellung), setzt der 4D Web Server die Kontextnummer in der URL der gesendeten HTML Dokumente.
  In älteren Versionen (Option markiert) sendete der 4D Web Server die

Kontextnummer für jeden Eintrag einer Seite an den Browser, was die Bearbeitung verlangsamte. Die Option muss aus Kompatibilitätsgründen unter Umständen aktiv sein. Beachten Sie, dass eine Änderung dieser Einstellung erst nach Neustart der Datenbank in Kraft tritt.

- **"/" von unbekannten URLs entfernen**: In früheren 4D Versionen wurden unbekannte URL, d.h. URL, die weder zu einer vorhandenen Seite, noch zu einer 4D spezifischen URL gehören, in den Datenbankmethoden *On Web Authentication* und *On Web Connection* ($1) zurückgegeben und begannen nicht mit dem Zeichen "/". Diese Vorgehensweise wurde in 4D 2004 entfernt. Sollten Sie Algorithmen eingebaut haben, die auf diesem Sonderfall basieren und wollen Sie das bisherige Verhalten beibehalten, deaktivieren Sie diese Option.
- **Drag & Drop von außerhalb von 4D verweigern**: 4D erlaubt ab Version 11 im Anwendungsmodus Drag&Drop für Auswahlen, Objekte und externe Dateien, z.B. Bilddateien.
  Diese Möglichkeit muss der Code der Datenbank jedoch unterstützen. In konvertierten Datenbanken aus früheren Versionen können Störungen auftreten, wenn der vorhandene Code nicht entsprechend angepasst wird. Um das zu verhindern, markieren Sie diese Option. Dann lassen sich externe Objekte nicht per Drag&Drop in 4D Formulare legen. Beachten Sie, dass sich externe Objekte mit dem Attribut **Automatisches Drop** weiterhin in Objekte ziehen lassen, sofern die Anwendung die Daten interpretieren kann (Text oder Bild). Weitere Informationen dazu finden Sie im Abschnitt **Drag and Drop**.
- **QUERY BY FORMULA auf Server ausführen und Execute ORDER BY FORMULA auf Server ausführen**: Ab 4D Version 11 werden die Such- und Sortierbefehle „nach Formel" auf dem Server ausgeführt und nur das Ergebnis an den Client-Rechner zurückgegeben. Das gilt für die 4D Befehle **QUERY BY FORMULA**, **QUERY SELECTION BY FORMULA** und **ORDER BY FORMULA**.
  Werden Variablen direkt in der Formel aufgerufen, wird die Suche mit dem Wert der Variablen auf dem Client-Rechner durchgeführt. So wird zum Beispiel die Anweisung

  ```
  <strong>QUERY BY FORMULA</strong>([aTable];[aTable]aField=theVariable)
  ```

  Dieses Prinzip gilt dagegen nicht für Formeln mit Methoden, die ihrerseits Variablen aufrufen: In diesem Fall wird der Wert der Variablen auf dem Server berechnet.
  In konvertierten Datenbank kann diese neue Funktionsweise vorhandene Algorithmen beeinträchtigen. Folglich werden in diesem Kontext die Befehle weiterhin auf dem Client-Rechner ausgeführt. Um die Vorteile des neuen Algorithmus in konvertierten Datenbanken zu nutzen, können Sie einfach diese Optionen markieren.
  Hinweis: Sie können diese Option auch über den 4D Befehl **SET DATABASE PARAMETER** setzen.
- **QUERY BY FORMULA Uses SQL Joins**: Ab 4D Version 11 führen die 4D Befehle **QUERY BY FORMULA** und **QUERY SELECTION BY FORMULA** auf SQL joins basierende Verknüpfungen aus. Demzufolge muss für eine Formel, die z.B. [Tabelle_A]Feld_X=[Tabelle B]Feld_Y enthält, keine automatische Verknüpfung zwischen Tabelle A und Tabelle B existieren.
  Zur Wahrung der Kompatibilität ist diese Option in konvertierten Datenbanken standardmäßig nicht markiert. Wir empfehlen jedoch, sie zu aktivieren (nach Überprüfung des Code der Datenbank), um die optimierte Arbeitsweise der Suchbefehle nach Formel nutzen zu können.
  Hinweise:

    - Ist der Modus „SQL joins" aktiviert, verwenden die Befehle **QUERY BY FORMULA** und **QUERY SELECTION BY FORMULA** trotzdem automatische Verknüpfungen, die im Strukturedítor gesetzt wurden, wenn folgendes gilt:
      - Die Formel lässt sich nicht in die einzelnen Elemente in Form von {Feld ;Vergleichsoperator ;Wert} aufbrechen
      - Es werden zwei Felder derselben Tabelle miteinander verglichen.
    - Sie können diese Option auch über den 4D Befehl **SET DATABASE PARAMETER** setzen.
- **Verschachtelte Transaktionen erlauben**: Diese Option unterstützt mehrstufige Transaktionen. 4D akzeptiert ab Version 11 verschachtelte Transaktionen auf einer unbegrenzten Anzahl von Ebenen. Da diese Option jedoch die Funktionsweise von Datenbanken älterer Versionen beeinträchtigen kann, ist sie in konvertierten Datenbanken standardmäßig inaktiv, d.h. hier sind Transaktionen auf eine Ebene begrenzt. Um in einer konvertierten Datenbank Transaktionen auf mehreren Ebenen durchzuführen, müssen Sie diese Option markieren.
  Diese Option ist standardmäßig nicht markiert. Sie gilt spezifisch für eine Datenbank.
  Hinweis: Diese Option hat keine Auswirkung auf Transaktionen, die mit der SQL-Engine von 4D ausgeführt werden. SQL-Transaktionen sind immer verschachtelt.
- **Unicode Modus**: Aktiviert bzw. deaktiviert den Unicode Modus für die aktuelle Datenbank. Im Unicode Modus verwalten Datenbank-Engine und Programmiersprache Zeichenketten in Unicode nativ. Im Kompatibilitätsmodus wird von der Programmiersprache der ASCII Zeichensatz verwendet.
  Mit dieser Option lässt sich die Kompatibilität konvertierter Datenbanken aufrechterhalten. Sie ist für Datenbanken, die mit 4D v11 oder höher erstellt wurden, standardmäßig markiert, für konvertierte Datenbanken nicht markiert.
  Hinweise:
    - Ändern Sie diese Option, wird sie erst nach Neustart der Anwendung berücksichtigt.
    - Sie gilt immer für eine Datenbank. Im interpretierten Modus können Sie demnach eine Unicode Datenbank zusammen mit Komponenten im Ascii Kompatibilitätsmodus haben, oder umgekehrt.
    - Sie können den Modus Unicode auch mit dem 4D Befehl **SET DATABASE PARAMETER** festlegen.

  Weitere Informationen zur Unterstützung von Unicode finden Sie im Handbuch *4D Programmiersprache* im Abschnitt **EXPORT TEXT**.
- **Verwende Punkt und Komma als Platzhalter in Zahlenformaten**: Ab 4D Version 11 basieren Zahlenformate automatisch auf den landesspezifischen Parametern des Systems (siehe "Zahlenformate" unter **Anzeigeformate**). 4D ersetzt die Zeichen "." und "," in Zahlenformaten automatisch mit den Trennzeichen für Tausend und Dezimal, wie es im Betriebssystem definiert wurde. Punkt und Komma dienen als Platzhalter in den Formaten 0 oder #. In bisherigen 4D Versionen haben Anzeigeformate für Zahlen nicht die landesspezifischen Parameter des Betriebssystems berücksichtigt. So ist das Format "###,##0.00" ein für das amerikanische Betriebssystem gültiges Format. Das führt in einem deutschen oder schweizer System zu einem inkorrekten Ergebnis.
  In konvertierten Datenbanken ist diese Funktionsweise zur Wahrung der Kompatibilität nicht aktiviert. Um sie zu nutzen, müssen Sie diese Option markieren.
- **Automatische Zuweisung von Variablen:** In bisherigen 4D Versionen hat der Web Server automatisch den Wert von Variablen kopiert, die über ein Web

Formular oder eine URL in 4D Prozessvariablen gesendet wurden. Im interpretierten Modus wurde der Wert jeder empfangenen Variable, direkt in eine 4D Prozessvariable mit demselben Namen kopiert; im kompilierten Modus mussten die Variablen zuvor in einer Projektmethode COMPILER_WEB deklariert werden.
Dieser Mechanismus ist ab 4D v13.4 veraltet und in neuen Anwendungen nicht mehr verfügbar. Zur Wahrung der Kompatibilität wird er in komvertierten Anwendungen noch beibehalten. Sie können ihn aber auch deaktivieren. Wir empfehlen jetzt, die dafür vorgesehenen Befehle **WEB GET VARIABLES** oder **WEB GET BODY PART** zu verwenden.

- **Verwende legacy Netzwerk-Schicht**: 4D Applikationen enthalten ab Version 14 R5 eine neue Netzwerk-Schicht mit Namen *ServerNet*, um die Kommunikation zwischen 4D Server und Rechnern mit remote 4D (Clients) zu steuern. Die bisherige Netzwerk-Schicht ist überholt, wird jedoch zur Wahrung der Kompatibilität mit vorhandenen Datenbank beibehalten. Mit dieser Option können Sie die bisherige Netzwerkschicht je nach Ihren Bedürfnissen in Ihren konvertierten 4D Server Anwendungen aktivieren oder deaktivieren, z.B. beim Migrieren Ihrer Client Applikationen (siehe Abschnitt **Netzwerk und Client-Server Optionen**). Diese Option ist in konvertierten Datenbanken standardmäßig aktiviert und für neu angelegte Datenbanken automatisch deaktiviert.
  Beachten Sie, dass die Änderung erst nach dem Neustart der Anwendung berücksichtigt wird. Auch Client-Anwendungen, die sich mit der neuen Netzwerk-Schicht anmelden können, müssen neu gestartet werden. Zum Verwenden von *ServerNet* ist mindestens die Client-Version von 4D v14 R4 erforderlich, siehe Abschnitt **Netzwerk und Client-Server Optionen**).
  Hinweis: Diese Option lässt sich mit dem Befehl **SET DATABASE PARAMETER** auch per Programmierung steuern.

- **Sichere Methode in Unicode:** Damit können Sie in 4D Methoden Code-Strings in Unicode sichern. In 4D Versionen vor v15 wurden Code Strings (Formeln, Variablen- und Methodennamen, Kommentare, etc.) von 4D Methoden über die aktuelle lokale Codierung gespeichert. Das konnte zu Problemen führen, insbesondere wenn Entwickler aus verschiedenen Ländern den 4D Code gemeinsam nutzten: Wird z.B. 4D Code mit französischen Akzenten oder deutschen Umlauten an einen englischen Entwickler gesendet, können diese Sonderzeichen verloren gehen. Größere Probleme treten bei Code auf, der mit einer japanischen Version geschrieben wird. Solche Probleme entfallen bei Methoden, die in Unicode gesichert werden, und ermöglicht den korrekten Austausch von 4D Code mit spezifischen landessprachlichen Zeichen. Wir empfehlen, die Option Unicode Modus für Methoden in Ihren Anwendungen sobald wie möglich zu aktivieren, insbesondere wenn Sie auf internationaler Ebene arbeiten.
  Hinweise:
  - Diese Funktionalität gilt für die Sprache selbst und ihre Interpretation. Einige Fenster des 4D Editors, wie die Eigenschaftenliste, verwenden noch die aktuelle lokale Codierung und zeigen so u.U. manche Strings nicht korrekt an. Das behindert jedoch nicht die Ausführung von Code.
  - Sie können die Option jederzeit aktivieren/deaktivieren. Die Änderung gilt nur für anschließend gesicherte Methoden.

- **Verwende Objektnotation, um auf Objekteigenschaften zuzugreifen (Unicode nötig):** Damit können Sie die Zeichen ".", "[" und "]" in Ihrem Code als Symbole der Objektnotation zum Definieren von Token Teilen verwenden --

und nicht in Tabellen-, Feld-, Methoden- oder Variablennamen. Wollen Sie Objektnotation verwenden, müssen Sie diese Option für Anwendungen aktivieren, die vor 4D v17 erstellt wurden. Denn in allen früheren 4D Releases waren die Symbole ".[]" in Namen erlaubt. Aktivieren Sie diese Option für Ihre konvertierte Anwendung, machen Sie ihren Code mit Objektnotation kompatibel. Wir empfehlen, die Kompatibilität Ihres Code über das MSC zu prüfen (siehe **Seite Prüfen**). Weitere Informationen dazu finden Sie im Abschnitt **Objektnotation verwenden**.
Hinweis: Für Objektnotation muss die Anwendung im Unicode-Modus arbeiten, d.h. die Optionen **Unicode Modus** und **Sichere Methode in Unicode** müssen markiert sein.

- **Verwende Datumstyp statt ISO Datumsformat in Objekten:** Damit können Sie Datumsangaben in Objektattributen als Datumstyp anstatt als Text im ISO-Format speichern. In bisherigen 4D Versionen ließ sich ein Datum in Objektattributen nur als String abspeichern. Ab 4D v16 R6 können Sie diese neue Option markieren, um das Datum in Objektattributen als Datumstyp zu speichern. Dies gilt nur für ein neu eingegebenes Datum, zuvor im ISO-Format gespeicherte Datumsangaben bleiben erhalten. Dieses Feature können Sie auch per Programmierung über den Befehl **SET DATABASE PARAMETER** und die Selektoren Date type, String type with time zone und String type without time zone steuern.

- **Benutze neue Architektur für Anwendungen-Weitergabe:** Diese Option ist für Anwendungen ab 4D v15 R4 standardmäßig markiert und in konvertierten Anwendungen standardmäßig deaktiviert. Wollen Sie die neuen Mechanismen nutzen, müssen sie diese explizit markieren. Sie muss auf dem Rechner gesetzt werden, der die Endanwendung generiert. Weitere Informationen dazu finden Sie in den Abschnitten **Zuletzt geöffnete Datendatei** und **Verbindungsprozess für eingebundene Clients**.

Geändertes Verhalten

OBJECT Get action

Die Befehle **OBJECT Get action** und **OBJECT SET ACTION** verwenden jetzt Konstanten mit Werten vom Typ String anstatt Lange Ganzzahl (bei Verwendung in Formularobjekten).

Bisheriger Code:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};Object Cancel action)
    End if
End for
```

Code nach Konvertierung in v17:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(_o_OBJECT Get action(*;$arrObjects{$i})=_o_Object No standard action)
        OBJECT SET ACTION(*;$arrObjects{$i};_o_Object Cancel action)
    End if
End for
```

Hinweis: Nur die Funktion _o_Object Get action wurde als überholt deklariert, da der Ergebnistyp schon vor dem Kompilieren gesetzt werden muss. Der Grund dafür ist, damit Sie in Betracht ziehen, Ihren Code zu ändern (siehe unten), da sonst keine Kompilierung möglich ist.

Neuer Code: Befehl und überholte Konstanten ersetzen mit:

```
ARRAY TEXT($arrObjects;0)
FORM GET OBJECTS($arrObjects)
For($i;1;Size of array($arrObjects))
    If(OBJECT Get action(*;$arrObjects{$i})=ak none)
        OBJECT SET ACTION(*;$arrObjects{$i};ak cancel)
    End if
End for
```

Stunden ausgedrückt in Sekunden

Ab 4D v17 werden über Objekteigenschaften verwaltete Stunden (Typ **C_TIME**) in **Anzahl von Sekunden** konvertiert. In bisherigen Versionen wurden sie in Millisekunden konvertiert.

Diese Änderung gilt für alle Stunden, die über Objektnotation aus/in Objekten oder Collections konvertiert werden: Befehle wie **OB SET** und **OB Get**, **QUERY BY ATTRIBUTE** oder JSON Befehle wie **JSON Stringify** und **JSON Parse**. Sie wirkt sich auch auf stündliche/digitale Konvertierungen in folgenden 4D Features aus:

- Web Areas (via JavaScript),
- 4D Mobile,
- SQL (**CAST** Funktion)

Um die Migration Ihrer 4D Anwendungen zu erleichtern (insbesondere, wenn Stunden in Objektattrituten von Feldern gespeichert wurden), ist die neue Option zur Kompatibilitat verfügbar, um das bisherige Verhalten für die Sitzung wiederherzustellen:

```
//Bisheriges Verhalten wiederherstellen
//In die Datenbankmethode On opening / On server startup setzen
SET DATABASE PARAMETER(Times inside objects;Times in milliseconds)
```

XML Änderungen

- **Externe Entities auflösen**
  Ab 4D v16R3 ist die Auflösung externer Entities in XML Dokumenten zur Erhöhung der Sicherheit standardmäßig **nicht** aktiviert (die Deklaration einer externen Entity generiert einen Analysefehler). Mit dem neuen Selector XML external entity resolution, Wert XML enabled im Befehl **XML SET OPTIONS** können Sie zum vorherigen Verhalten zurückgehen.

- **Unterscheidung zwischen Groß- und Kleinschreibung**
  Ab 4D v16R4 unterscheiden die Befehle **DOM Get XML element** und **DOM Count XML elements** standardmäßig zwischen Groß- und Kleinschreibung. Mit dem neuen Selector können Sie zum vorigen Verhalten zurückgehen: XML DOM case sensitivity hat folgende Werte:
  - XML case sensitive (Standard): Befehle beachten Groß- und Kleinschreibung
  - XML case insensitive: Befehle beachten nicht Groß- und Kleinschreibung

4D Write Pro und Bildattribute

Ist das Bild nicht deklariert, geben die Attribute wk image, wk list style image und wk background image in den Befehlen **WP GET ATTRIBUTES** oder **OB Get** das Bild selbst und nicht seine URL zurück.

Mit den neuen Attributen wk image url, wk list style image url und wk background image url erhalten Sie die URL des Bildes.

Namensänderung

4D Write Pro

- WP Get paragraphs wurde umbenannt in **WP Create paragraph range**
- WP Get pictures wurde umbenannt in **WP Create picture range**

Konstanten für Standardaktionen

Die numerischen Konstanten unter dem Thema **Standardaktion** sind überholt (Sie beginnen jetzt mit "_o_"). Ab 4D v16 R3 empfehlen wir, die Textkonstanten zu verwenden, sie beginnen mit "ak ...".

Überholte Funktionalitäten

Überholte Befehle

Um alle überholten Befehle in Ihren Anwendungen zu finden, können Sie im Menü **Bearbeiten** über **Suche in Struktur** oder in der Werkzeugleiste über den Bereich **Suche in Struktur** nach **_o_** suchen. Wir raten dringend, überholte Befehle durch die aktuellen Befehle oder Funktionen zu ersetzen, da sie in zukünftigen Versionen des Programms nicht mehr unterstützt werden. Überholte Befehle funktionieren nur noch zur Wahrung der Kompatibilität für eine gewisse Zeit.

Die Liste mit allen überholten Befehlen erhalten Sie unter **Überholte bzw. entfernte Befehle**

Überholte Befehle in 4D v16

- **4D Befehle**

Bisheriger Name	Neuer Name	Kommentar	Überholt seit	Aktueller Status
Object Get action	**OBJECT Get action** (oder **_o_OBJECT Get action**bei Konvertierung) mit geändertem Parameter (hat sich von einem Parameter mit Wert vom Typ Lange Ganzzahl geändert in Parameter mit Wert vom Typ String)	bei Konvertierung gibt die Funktion _o_Object read action eine Änderung im Verhalten an (siehe Kaptiel "Geändertes Verhalten").	v16R3	Überholt
PICTURE TO GIF	**_o_PICTURE TO GIF**	**PICTURE TO BLOB**	v16R5	Überholt
Type document	**_o_Document type**	die Funktion **Path to object** verwenden	v16R6	Überholt
Document creator	**_o_Document creator**	die Funktion **Path to object** die Funktion	v16R6	Überholt
SET DOCUMENT TYPE	**_o_Document type**	die Funktion **Object to path** verwenden	v16R6	Überholt
SET DOCUMENT CREATOR	**_o_SET DOCUMENT CREATOR**	die Funktion **Object to path** verwenden	v16R6	Überholt
MAP FILE TYPES	**_o_MAP FILE TYPES**	UTIs und info.plist verwenden	v16R6	Überholt
Gestalt	**_o_Gestalt**	wurde ersetzt durch **Get system info** / **Is macOS** / **Is Windows**	v17	Überholt
PLATFORM PROPERTIES	**_o_PLATFORM PROPERTIES**	wurde ersetzt durch **Get system info** / **Is macOS** / **Is Windows**	v17	Überholt

- **4D Konstanten**

Bisheriger Name	Neuer Name	Kommentar	Überholt seit	Aktueller Status
lk display hor scrollbar	_o_lk display hor scrollbar	**OBJECT GET SCROLLBAR**	v16R3	Überholt
lk display ver scrollbar	_o_lk display ver scrollbar	**OBJECT GET SCROLLBAR**	v16R3	Überholt
lk hor scrollbar position	_o_lk hor scrollbar position	**OBJECT GET SCROLL POSITION**	v16R3	Überholt
lk ver scrollbar position	_o_lk ver scrollbar position	**OBJECT GET SCROLL POSITION**	v16R3	Überholt
lk footer height	_o_lk footer height	**LISTBOX Get footers height**	v16R3	Überholt
lk header height	_o_lk header height	**LISTBOX Get headers height**	v16R3	Überholt
_o_Object xxx	ak xxx	Konstanten, die Standardaktionen unterstützen. Siehe dazu die Befehle **OBJECT Get action** und **OBJECT SET ACTION**.	v16R3	Überholt

- **4D Internet Commands**

Bisheriger Name	Neuer Name	Kommentar	Überholt seit	Aktueller Status
FTP_Progress	gibt bei Aufrufen einen Fehler zurück	Die Funktionen FTP_Append, FTP_Receive, FTP_Send unterstützen nicht mehr den Parameter *progress*.	v16R3	Deaktiviert

Deaktivierte Funktionen

- Das Plug-In **4D Pack** ist seit 4D v16 R2 entfernt.
- **FTP_Progress**: Diese Funktion ist in 4D Internet Commands seit 4D v16R2 entfernt. Bei Aufruf gibt sie jetzt den Fehler -2201 zurück.
  Hinweis: Folgende Funktionen unterstützen den Parameter *progress* nicht mehr:
  FTP_Append
  FTP_Receive
  FTP_Send

Von 32-bit Versionen auf 64-bit Versionen wechseln

4D v17 funktioniert in 32-bit oder 64-bit, macOS oder Windows.

Für das Upgraden einer 4D Anwendung von einer 32-bit Version auf eine 64-bit Version sind ein paar Vorarbeiten erforderlich.

Die gesamte 4D Produktreihe in 64-bit ist nicht mehr von Altura's *Mac2Win* Library abhängig. **Altura's Mac2Win wurde komplett aus den 64-bit Versionen von 4D Developer Edition und 4D Volume Desktop entfernt, so dass diese Produkte moderne Windows APIs voll nutzen können:**

- 4D Developer Edition und 4D Volume Desktop Windows 64 bits in v17 (seit v16 R2)
- 4D Server Windows 64 bits in 4D v17 (seit v16 R4)

Hinweis: 4D Windows Versionen in 32-bit verwenden noch Mac2Win.

Die 4D Developer Edition in 64-bit enthält neue Editoren für Etiketten, Schnellbericht und Import-Export... Sie sind modern, intuitiv und leicht zu verwenden! Sie können auf Ihrem Rechner mehrere Instanzen der 4D Developer Edition in 64-bit starten und müssen Ihre Anwendung nicht doppelt installieren.

Läuft Ihre Anwendung auf 4D Server 64-bit Windows oder OS X, ist das meiste schon erledigt. Für 64-bit Versionen von Desktop Anwendungen sind evtl. zusätzliche Schritte erforderlich. Im folgenden werden die einzelnen Punkte erläutert, damit Sie alle notwendigen Operationen vor und nach dem Upgrade prüfen können.
Für die Migration Ihrer Produkte auf 64-bit wurden verschiedene Features upgedated, deaktiviert, oder als überholt deklariert. Alle Details dazu finden Sie im Abschnitt .

Hinweis: Wie bei allen Upgrade Operationen ist es eine bewährte Praxis, MSC zu verwenden und vor jedem größeren Schritt eine Überprüfung zu starten, um sicherzustellen, dass Daten und Struktur gültig sind.

Plug-Ins prüfen

In der ersten Etappe updaten Sie Ihre Plug-Ins (sofern vorhanden) auf die jeweiligen 64-bit Versionen:

- **4D Plug-Ins:**
  Alle Plug-Ins arbeiten bereits in 64-bit Versionen, mit Ausnahme von 4D Write und 4D View.
  - Verwendet Ihre Anwendung 4D Write, sollten Sie die Migration Ihres Code auf 4D Write Pro vornehmen. Ein gute Praxis ist, Ihren vorhandenen 32-bit Code beizubehalten und daneben ein neues auf 64-bit basierendes Modul mit 4D Write Pro zu starten. Wollen Sie

während der Übergangsphase sowohl Dokumente in 4D Write als auch in 4D Write Pro nutzen, sollten Sie 4D v17 in 32-bit einsetzen.
    - Verwendet Ihre Anwendung 4D View, müssen Sie die Funktionalitäten von 4D View Pro oder andere Alternativen einsetzen.
- **Third-party Plug-Ins:**
  Wenden Sie sich an die jeweiligen Hersteller, um 64-bit Versionen zu erhalten

Vorbereitungen in der 32-bit Version

1. Upgraden Sie Ihre Anwendung auf das neueste 32-bit Release, z.B. 4D v17 32-bit
2. Stellen Sie sicher, dass Unicode aktiviert ist.
3. Konvertieren Sie alle Bilder im Format PICT/cicn/QuickTime.
   Über den Befehl **GET PICTURE FORMATS** können Sie veraltete Bildformate in Ihren Daten ausfindig machen. Sie müssen auch alle nicht-unterstützten Bildformate in der Struktur Ihrer Datenbank ersetzen. Eine Überprüfung mit dem MSC findet veraltete Bilder in Ressourcen Dateien für Bild- oder 3D Schaltflächen sowie statische Bilder.
4. Ersetzen Sie auf XSLT-basierende Features (Befehle **_o_XSLT APPLY TRANSFORMATION**, **_o_XSLT SET PARAMETER** oder **_o_XSLT GET ERROR**) z.B. mit dem Befehl **PROCESS 4D TAGS**.
5. Ersetzen Sie Aufrufe von **_o_Font number** durch Aufrufe von Schriftnamen.
6. Entfernen Sie allen Code, der Ressourcen-Dateien erstellt oder ändert.

Jetzt ist alles bereit, um Ihre Anwendung mit einer 64-bit Version von 4D zu öffnen.

Anwendung in 64-Bit öffnen und prüfen

1. Öffnen Sie Ihre Anwendung mit einer 4D Developer Edition in 64-bit.
2. Setzen Sie das integrierte WebKit für Ihre Web Bereiche ein, überprüfen Sie diese, da automatisch zur Engine des Systems gewechselt wird (Der Zugriff auf 4D Methoden über $4d ist weiterhin gültig).
3. Verwendet Ihr Code die Konstante Mac spool file format option des Befehls **SET PRINT OPTION**, müssen Sie diese ersetzen mit Aufrufen von **SET CURRENT PRINTER** mit der Konstanten Generic PDF driver.
4. Überprüfen Sie Aufrufe und Verwendungen des Etiketteneditors (siehe **Etiketteneditor (64-bit)**).
5. Überprüfen Sie Aufrufe und Verwendungen des Schnellberichteditors (siehe **Schnellberichte (64-bit)**)

Ihre Anwendung ist jetzt voll kompatibel mit 64-bit und Sie können die Vorzüge aller neuen Features in der 64-bit Version von 4D nutzen.

Ihre Vorteile mit 64-bit Features

Die Highlights sind:

- Die **64-bit Architektur** hebt die Begrenzung des Datenbank Cache auf. Verbessern Sie die Performance Ihrer Anwendung einfach durch Verwenden eines größeren Cache.
  Setzen Sie die **leistungsstarken neuen Features** in 64-bit ein, wie z.B.

preemptive Prozesse, animierte Formularobjekte oder neue Features zum Drucken
Erstellen Sie Ihre Anwendungen mit der 4D Runtime Volume License 64-bit.
Verwenden Sie die final 64-bit Versionen von 4D Server auf Windows und macOS - siehe Abschnitte **4D Server 64-bit für OS X verwenden** und **Einen 64-bit 4D Server (Windows) einsetzen**

- **Neuer Schnellberichteditor**, kompatibel mit Berichten, die mit einer vorigen Version erstellt wurden - siehe unter **Etiketteneditor (64-bit)**.

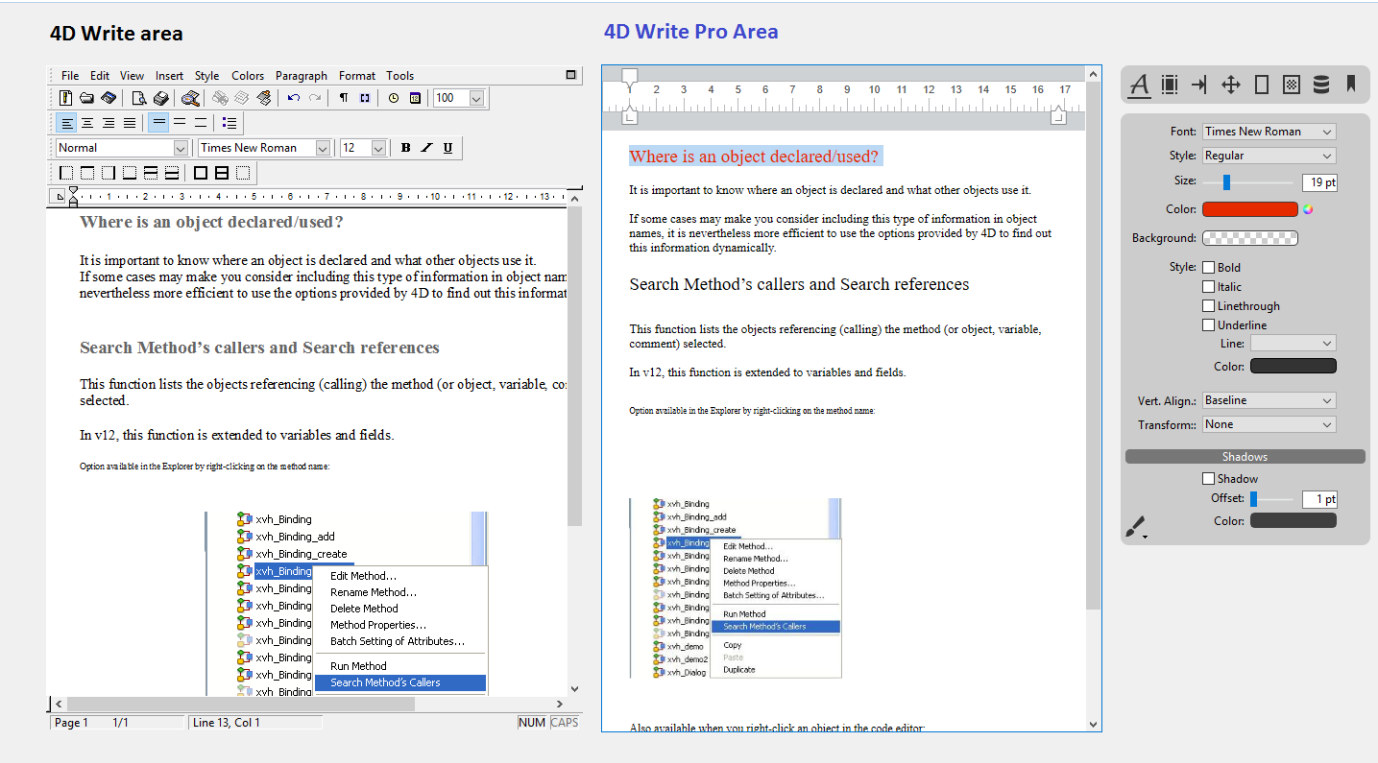
- **Neuer Etiketteneditor**, kompatibel mit Etikettendateien, die mit einer vorigen Version erstellt wurden - siehe unter **Schnellberichte (64-bit)**.

- Mit dem Befehl **GRAPH** können Sie **Diagramme** mit dem Parameter vom Typ *Objekt* in einem Aufruf erstellen.

4D Write Dokumente in 4D Write Pro konvertieren

Ein 4D Write Dokument konvertieren

4D Write Pro kann 4D Write Dokumente öffnen und konvertieren und die meisten der spezifischen Eigenschaften aufrechterhalten:



Im Bild sehen Sie links einen 4D Write Bereich und rechts einen 4D Write Pro Bereich (erstellt mit dem neuen Objekt aus der Bibliothek - siehe unten). Der Inhalt des 4D Write Bereichs wurde einfach über die Funktion **WP New** übertragen:

```
// Inhalt des 4D Write Bereich im 4D Write Pro Bereich wiederfinden
[WRITEAREAS]AreaNTWP:=WP New([WRITEAREAS]AreaNT_)
```

Da jedoch 4D Write nur mit 32-bit Versionen verwendbar ist, müssen Sie Ihre 4D Write Dokumente vor dem Wechsel auf die 64-bit Version konvertieren.

Im Gegensatz zu 4D Write ist 4D Write Pro kein Plug-In, sondern direkt in 4D integriert. Beachten Sie, dass 4D Write Pro dieselbe Lizenz wie 4D Write verwendet. Diese Lizenz muss in Ihrem Programm installiert sein, damit Sie 4D Write Pro nutzen können.

4D Write Pro Objekte ermöglichen, 4D Write Dokumente auf zwei Wegen zu importieren:

- Für 4D Write Dateien, die auf der Festplatte gespeichert sind, verwenden Sie die Funktion **WP Import document**. 4D Write Dateien (.4w6, .4w7, und .4wt - Vorlage) müssen in 4D Write Pro Dateien (.4wp) konvertiert werden.

```
// Zuerst mit 4D Write Befehlen .4w6 Dateien in .4w7 Dateien konvertieren
$offscreen:=WR New offscreen area
WR OPEN DOCUMENT($offscreen;"myFile.4w6";"4WR6")
WR SAVE DOCUMENT($offscreen;"myFile.4w7";"4WR7")
WR DELETE OFFSCREEN AREA($offscreen)
```

```
// Dann mit 4D Write Pro Befehlen .4w7 in .4wp konvertieren
C_OBJECT($docWritePro)
$docWritePro:=WP Import document("myFile.4w7")
WP EXPORT DOCUMENT($docWritePro;"myFile.4wp")
```

- Für 4D Write Dateien, die in einer Datei gespeichert sind, verwenden Sie die Funktion **WP New**. In BLOB oder Bildfeldern gespeicherte 4D Write Dateien müssen in Objektfelder gesetzt werden.

```
// Von einem Bildfeld in ein Objektfeld über ein BLOB
// [DocWRITE]WritePictArea_ is a picture field
// $Blob is a BLOB
// [DocWRITE]WriteProArea is an Object field
$offscreen:=WR New offscreen area
WR PICTURE TO AREA($offscreen;[DocWRITE]WritePictArea_)
$Blob:=WR Area to blob($offscreen;1)
[DocWRITE]WriteProArea :=WP New($Blob)
```

```
// Von einem BLOB Feld in ein Objektfeld
// [DocWRITE]WriteBLOBArea_ is a BLOB field
// [DocWRITE]WriteProArea is an object field
[DocWRITE]WriteProArea :=WP New([DocWRITE]WriteBLOBArea_)
WR DELETE OFFSCREEN AREA($offscreen)
```

Hinweise zur Kompatibilität:

- Nur 4D Write Dokumente der letzten Generation (4D Write v7) werden unterstützt.
- Unter **Welche Eigenschaften von 4D Write werden übernommen?** erfahren Sie, welche Features und Objekte sich importieren lassen.
- Kopieren aus einem 4D Write Dokument und Einsetzen in einen 4D Write Pro Bereich wird derzeit nicht unterstützt. Ein 4D Write Dokument lässt sich nur über Befehle der 4D Write Pro Programmiersprache importieren.
- Unter Windows basieren 4D Write Pro Features auf Direct2D. Stellen Sie bei Rechnern mit Windows 7 oder Windows Server 2008 sicher, dass die

Komponente *Platform Update for Windows* installiert ist, damit Sie die erforderliche Direct2D Version nutzen können.

4D Ausdrücke filtern

Das Filtern für 4D Write Pro Dokumente war in vorigen Versionen nicht aktiviert. Enthalten Ihre 4D Write Pro Dokumente Verweise auf 4D Methoden, werden diese nach dem Konvertieren in 4D v16 oder höher nicht mehr korrekt bewertet. An ihrer Stelle erscheint eine Meldung "## Error # 48".
In diesem Fall müssen Sie diese Methoden über den Befehl **SET ALLOWED METHODS** zur Liste der zugelassenen Methoden hinzufügen.

Neue und geänderte Befehle

In 4D Write Pro wurden vorhandene Befehle sowie Funktionalitäten weiter ausgebaut und neue hinzugefügt:

- Horizontales Lineal: Zum Definieren von Rändern, Einrückungen und Tabulatoren
- Eigene Toolbar: Erweitert den Einsatz von Standardaktionen
- Update des Befehls **Dynamic pop up menu**: Zum Gestalten Ihres eigenen Kontextmenüs in 4D Write Pro anhand von Standardaktionen
- Ausbau der Tabellenverwaltung: **WP Insert table**, **WP Table append row**, **WP Table get rows**, **WP Table get columns**, **WP Table get cells**
- Hyperlinks: Neues Attribut wk link url für die Befehle **WP SET ATTRIBUTES** und **WP GET ATTRIBUTES**
- Bild einfügen: Befehl **WP Add picture** und seitenfüllendes Hintergrundbild über das Attribut wk paper box des Befehls **WP SET ATTRIBUTES**
- Verwalten von Kopf-, Fuß- und Hauptteil: **WP Get header**, **WP Get footer**, **WP Get body**
- Befehle zum Setzen (**WP SET FRAME**) und Lokalisieren (**WP Get frame**) des Cursor
- Führende Zeichen für Tabulatoren (Befehl **WP SET ATTRIBUTES**).

4D View Dokumente in 4D View Pro konvertieren

Konvertierung

4D v16 R6 hat die erste Etappe zum Konvertieren Ihrer 4D View Dokumente in 4D View Pro als **Preversion** eingeführt. Mit der neuen Funktion VP Convert from 4D View werden die meisten Eigenschaften und die in 4D View Dokumenten gespeicherten Informationen automatisch konvertiert: die Struktur des Dokuments, Werte, Formate, Stilarten, Ränder und Formeln.

Weitere Informationen dazu finden Sie auf folgenden Seiten des 4D View Pro Handbuchs:

- **4D View Dokumente konvertieren** und **Vorgehensweise**
- **Verfügbare Funktionalitäten**

4D View Pro Area

Ein neues Formularobjekt 4D View Pro (**4D View Pro Objekt**) und neue Befehle sind verfügbar:

- Sie erstellen ein neues Dokument mit **VP NEW DOCUMENT**
- Sichern es auf der Festplatte mit **VP EXPORT DOCUMENT** oder in der Datenbank mit **VP Export to object**
- Und öffnen es erneut mit **VP IMPORT DOCUMENT** oder **VP IMPORT FROM OBJECT**

Anhang: 4D v16 Rx Release Notes

Im Laufe des R-Release Zyklus wurden verschiedene von 4D verwendete Libraries und Kompontenten aktualisiert und bekannte Fehler behoben. Diese Angaben wurden für die jeweilige R-Version in den "Release Notes" veröffentlicht, und erscheinen hier als Zusammenfassung.

4D v16 R2

4D Developer und 4D Volume Desktop für Windows 64-bit sind ab 4D v16 R2 final Versionen.

Das Plug-In 4D Pack ist nicht mehr in Installern enthalten. Beachten Sie, dass 4D Pack für 4D v16.x mit 4D v16 R.x kompatibel ist.

4D v16 R3

Die alte Altura Library wurde aus 4D Developer Edition und 4D Volume Desktop für Windows 64-bit entfernt.

XML: Zur Erhöhung der Sicherheit ist die Auflösung externer Entities in XML Dokumenten standardmäßig **nicht** aktiviert. Mit dem neuen Selector XML external entity resolution für **XML SET OPTIONS** können Sie zum vorigen Verhalten zurückgehen (siehe **XML Änderungen**).

4D v16 R4

Xerces: Update auf Version 3.1.4 (intern genutzte "Open Source" Library zum Verwalten von XML).

Die alte Altura Library wurde aus 4D Server für Windows 64-bit entfernt. Ab jetzt ist die gesamte Produktreihe in 64 bits "Altura-frei".

XML: Ab 4D v16 R4 unterscheiden die Befehle **DOM Get XML element** und **DOM Count XML elements** standardmäßig zwischen Groß- und Kleinschreibung. Mit dem neuen Selector XML DOM case sensitivity für den Befehl **XML SET OPTIONS** können Sie zum vorigen Verhalten zurückgehen. (siehe **XML Änderungen**).

4D v16 R5

CEF: Update auf Version 301. Chromium Embedded Framework (CEF).

libldap: Update auf Version 2.45.

libsasl: Update auf Version 2.1.17. (Simple Authentication Security Layer)

libzip: Update auf Version 1.2.

zlib: Update auf Version 1.2.11.(data compression software library).

HDPI Screen für Windows 10 und alte Plug-Ins:

- Die alten Plug-Ins 4D Write und 4D View sind nicht für High DPI Screens für Windows 10 (1709) zertifiziert. Wir empfehlen die Verwendung der 64-bit Produktreihe in diesem Kontext (4D Write Pro und 4D View Pro).

Web Area in 4D 32-bit und macOS 10.13:

- Der Web Bereich kann beim Wechseln vom Hintergrund in den Vordergrund den Fokus verlieren. Wir empfehlen die Verwendung der 64-bit Produktreihe in diesem Kontext.

PDFs in einem Web Bereich mit 4D 32-bit für macOS 10.13 rendern:

- Der Web Bereich zum Anzeigen eines PDF mit dem Plug-In PDF Viewer für macOS 10.13 kann fehlschlagen. Wir empfehlen die Verwendung der 64-bit Produktreihe in diesem Kontext.

4D v16 R6

Hunspell: Update auf Version 1.6.2.

Ist keine Bildvariable deklariert, geben die Attribute wk image, wk list style image und wk background image der Befehle **WP GET ATTRIBUTES** und **OB Get** das Bild selbst und nicht seine URL zurück. Um die URL des Bildes zurückzugeben, verwenden Sie die neuen Attribute wk image url, wk list style image url und wk background image url.

WP Get paragraphs wurde umbenannt in **WP Create paragraph range**.
WP Get pictures wurde umbenannt in **WP Create picture range**.

4D v17

Anhang: Hilfreiche Methoden für die Konvertierung

Zum Generieren einer Datei auf der Festplatte mit der Liste nicht-indizierter einmaliger Felder

TechTip um eine Datei auf der Festplatte zu erstellen, die nicht-indizierte einmalige Felder anzeigt:

```
C_LONGINT($maxTableNumber_l;$currentTable_l)
C_LONGINT($maxFieldCount_l;$currentField_l)
C_LONGINT($dontCare_l)  // Für GET FIELD PROPERTIES nicht-verwendete Werte
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$lf_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)

$delim_t:=Char(Tab)
$lf_t:=Char(Carriage return)+Char(Line feed)

$logHeader_t:="Unique fields without index:"+$lf_t

$logfile_t:=Get 4D folder(Logs folder)+"UniqueNotIndexed.txt"

$logfile_h:=Create document($logfile_t)

If(OK=1)

   SEND PACKET($logfile_h;$logHeader_t)

   $maxTableNumber_l:=Get last table number

   For($currentTable_l;1;$maxTableNumber_l)
      If(Is table number valid($currentTable_l))
         $maxFieldCount_l:=Get last field number(Table($currentTable_l))
         For($currentField_l;1;$maxFieldCount_l)
            If(Is field number valid($currentTable_l;$currentField_l))

// Beachten Sie die folgenden Zeilenumbrüche über zwei Zeilen im Text,
// Das ist eine Anweisung in der Methode:
                  GET FIELD PROPERTIES($currentTable_l;$currentField_l;$dontCare_l;\
```

```
                $dontCare_l;$isIndexed_f;$isUnique_f;$dontCare_f)

                If(($isUnique_f) & (Not($isIndexed_f)))

                    $tableName_t:=Table name(Table($currentTable_l))
                    $fieldName_t:=Field name(Field($currentTable_l;$currentField_l))

                    $logRecord_t:="["+$tableName_t+"]"+$fieldName_t+$lf_t

                    SEND PACKET($logfile_h;$logRecord_t)

                End if
            End if
        End for
    End if
End for

CLOSE DOCUMENT($logfile_h)
SHOW ON DISK($logfile_t)
End if
```

Zum Erstellen fehlender Indizes

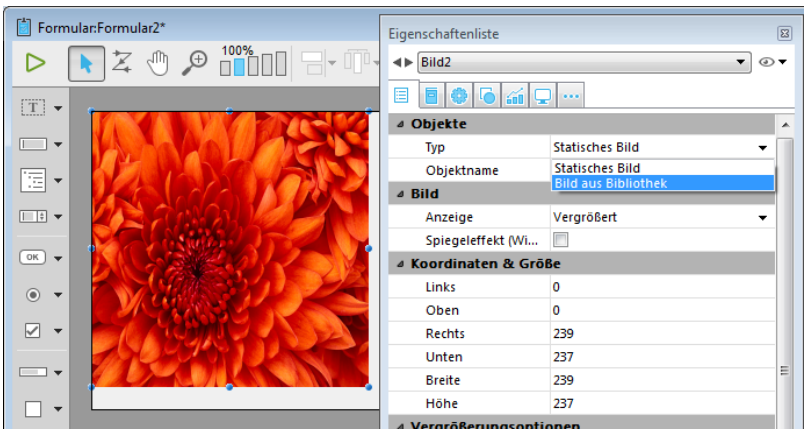
TechTip zum Erstellen fehlender Indizes auf Felder, die als einmalig deklariert, aber nicht indiziert sind:

```
C_LONGINT($maxTableNumber_l;$currentTable_l)
C_LONGINT($maxFieldCount_l;$currentField_l)
C_LONGINT($dontCare_l)  // Für GET FIELD PROPERTIES nicht verwendete Werte
C_BOOLEAN($dontCare_f;$isIndexed_f;$isUnique_f)
C_TEXT($logHeader_t;$logRecord_t;$logfile_t)
C_TEXT($delim_t;$lf_t)
C_TIME($logfile_h)
C_TEXT($tableName_t;$fieldName_t;$note_t)


$maxTableNumber_l:=Get last table number

For($currentTable_l;1;$maxTableNumber_l)
    If(Is table number valid($currentTable_l))
        $maxFieldCount_l:=Get last field number(Table($currentTable_l))
        For($currentField_l;1;$maxFieldCount_l)
            If(Is field number valid($currentTable_l;$currentField_l))

  // Beachten Sie die folgenden Zeilenumbrüche über zwei Zeilen im Text,
  // Es ist eine Anweisung in der Methode:
```

```
                GET FIELD PROPERTIES($currentTable_l;$currentField_l;$dontCare_l;\
                $dontCare_l;$isIndexed_f;$isUnique_f;$dontCare_f)

                If(($isUnique_f) & (Not($isIndexed_f)))

                    $tablePtr:=Table($currentTable_l)
                    $fieldPtr:=Field($currentTable_l;$currentField_l)
                    $tableName_t:=Table name($tablePtr)
                    $fieldName_t:=Field name($fieldPtr)
                    $indexName_t:="["+$tableName_t+"]"+$fieldName_t+" indexed for
uniqueness (kb#77023)"
                    ARRAY POINTER($fieldsArray_p;1)
                    $fieldsArray_p{1}:=$fieldPtr
                    CREATE INDEX($tablePtr->;$fieldsArray_p;Standard BTree
Index;$indexName_t;*)

                End if
            End if
        End for
    End if
End for
```

Bildtyp konvertieren

In einer 32-bit Version ausführen *(*z.B. vor der Umstellung auf 64 bits).
1 - Übertragen Sie Ihre statischen Bilder in Formularen in die Bildbibliothek:



2 - Konvertieren Sie die übertragenen Bilder in .png oder .jpeg:

```
C_LONGINT($i;$SOA;$RIS;$PictRef)
C_TEXT($PictName)
C_PICTURE($Pict)
 //---------- Arrays initialisieren ----------------
ARRAY LONGINT($aL_PictRef;0)
ARRAY TEXT($aT_PictName;0)
ARRAY TEXT($at_Codecs;0)
PICTURE LIBRARY LIST($aL_PictRef;$aT_PictName)
```

```
$SOA:=Size of array($aL_PictRef)
  //------------ PICT in png konvertieren ------------------
If($SOA>0)
    For($i;1;$SOA)  // for each image
        $PictRef:=$aL_PictRef{$i}
        $PictName:=$aT_PictName{$i}
        GET PICTURE FROM LIBRARY($aL_PictRef{$i};$Pict)
        GET PICTURE FORMATS($Pict;$at_Codecs)
        For($j;1;Size of array($at_Codecs))
            If($at_Codecs{$j}=".pict")  // ist das Format überholt
                CONVERT PICTURE($Pict;".png")  // in png konvertieren
  // und in der library speichern
                SET PICTURE TO LIBRARY($Pict;$PictRef;$PictName)
            End if
        End for
    End for
Else
    ALERT("Die Bildbiliothek ist leer.")
End if
  //------------ Ende der Methode ------------------
```

-

-