

## Anhang 2

### IntelliSense in VFP 7.0

**Um eine Sache haben VFP-Entwickler die Anwender anderer Entwicklungsumgebungen von Microsoft lange Zeit beneidet: um IntelliSense. Mit VFP 7 ist diese Erweiterung der Produktivität auch hier verfügbar.**

Eine der offensichtlichsten Unterschiede zwischen VFP 6 und anderen Entwicklungsumgebungen von Microsoft ist IntelliSense. Geben Sie in Visual Basic eine Anweisung wie „Dim SomeVariable AS“ ein und Sie erhalten eine Liste der Typen, als welche eine Variable definiert werden kann (sowohl Datentypen als auch Objekte). Die Auswahl aus der Liste verhindert Rechtschreibfehler und Sie müssen nicht jedes Mal aus der Dokumentation die Syntax seltener Befehle herausuchen. Seit der Einführung von IntelliSense haben sich die VFP-Entwickler benachteiligt gefühlt.

Jetzt nicht mehr! VFP 7 führt IntelliSense auch für VFP-Entwickler ein und bietet nach dem langen Warten eine bessere Version von IntelliSense als andere Produkte. Wenn Sie erst einmal herausgefunden haben, in welchem Maße IntelliSense Ihre Produktivität steigert, werden Sie sich wundern, wie Sie jemals ohne diese Funktionalität ausgekommen sind.

---

#### Was IntelliSense bietet

---

IntelliSense ist nur ein Wort, bietet aber unterschiedliche Verhalten, einschließlich:

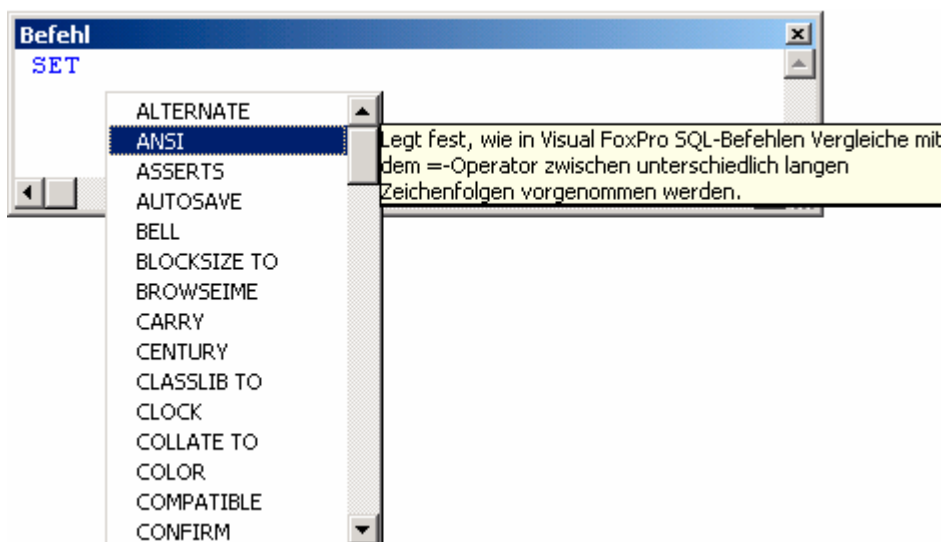
- Automatische Anweisungsvervollständigung
- Tipps zur Syntax von Befehlen und Funktionen
- Listen der dazugehörigen Objekte, Werte und häufig genutzte Dateien
- Automatische Anweisungsvervollständigung

Xbase-Sprachen wie Visual FoxPro haben immer zwei Arten der Eingabe von Befehlen und Funktionen unterstützt: die vollständige Eingabe des Schlüsselworts (z. B. BROWSE) sowie die Eingabe der mindestens vier ersten Buchstaben (z. B. brow). Erfahrene Entwickler betonen immer, dass es eine feine Sache ist, „Repl“ im Befehlsfenster einzugeben, aber im Code sollten Sie aus Gründen der Lesbarkeit das vollständige „Replace“ ausschreiben.

VFP 7 bietet das Beste aus beiden Welten: Sie können nun gerade so viele Zeichen eingeben, dass ein Schlüsselwort eindeutig identifiziert werden kann, anschließend ein Leerzeichen eingeben oder auf Enter drücken und VFP vervollständigt den Namen der Funktion für Sie. Geben Sie als Beispiel „modi“ ein und benutzen die Leertaste, ersetzt VFP Ihre Eingabe durch „MODIFY“. Da einige Schlüsselworte mit den gleichen Buchstaben beginnen, müssen Sie sicherstellen, dass Sie genügend Buchstaben eingeben, um das Schlüsselwort von anderen zu unterscheiden. Wollen Sie z. B. die Funktion `MessageBox()` einsetzen und geben lediglich „mess(“ ein, erweitert VFP dies zu „MESSAGE(“.

Einige Befehle bestehen aus mehr als einem Schlüsselwort, dazu gehören Alter Table, Report Form und Open Database. Da dem ersten Schlüsselwort immer das zweite folgen muss, fügt VFP dies automatisch hinzu. Dies ist ein wenig gewöhnungsbedürftig. Vielleicht geben Sie plötzlich den Befehl „OPEN DATABASE DATABASE MyData“ ein, da Sie nicht bemerkt haben, dass VFP „DATABASE“ automatisch eingefügt hat, nachdem Sie das Leerzeichen nach „OPEN“ eingegeben haben.

„SET“ ist das erste Schlüsselwort in einer langen Liste von Befehlen. Wenn Sie „set“ eingeben und die Leertaste drücken, zeigt IntelliSense eine Liste der Schlüsselworte an, die auf „set“ folgen können. Dazu gehören „deleted“ und „exact“ (vgl. Abbildung 1). Einige Befehle enthalten das Wort „to“, z. B. SET CLASSLIB TO“. Weiter hinten in diesem Kapitel werden Sie erfahren, dass Sie wählen können, ob IntelliSense das „to“ automatisch einfügen soll oder nicht.



**Abbildung 1.** IntelliSense zeigt eine Liste aller möglichen SET-Befehle an.

VFP vervollständigt keine Schlüsselwörter in der Mitte einer Anweisung. Geben Sie z. B. die WHILE-Klausel des Befehls REPLACE als „whil“ ein bleibt es so stehen. VFP erweitert es nicht zu „WHILE“.

Beachten Sie die Groß- und Kleinschreibung, die VFP nutzt. Geben Sie z. B. das gesamte Wort „modify“ ein, wird dies von VFP durch „MODIFY“ ersetzt. Dies ist zwar für Xbase-Entwickler interessant, die Schlüsselwörter immer in Großschreibung einsetzen, ärgert aber diejenigen, die Kleinschreibung oder die Kamelnotation (z.B. „TableUpdate“()) bevorzugen. Zum Glück können Sie dieses Verhalten aber auch kontrollieren. Mehr dazu im Abschnitt „Configuring IntelliSense“ in diesem Kapitel.

Es kann passieren, dass Sie nicht wollen, dass VFP ein Schlüsselwort vervollständigt. Mit Strg-Leertaste statt der Leertaste allein, unterdrücken Sie die Erweiterung des Befehls. Um die Erweiterung zurückzunehmen, drücken Sie zwei mal Strg-Z (beim ersten Drücken wird die Erweiterung gelöscht, beim zweiten Mal das Original-Schlüsselwort markiert zurück geschrieben).

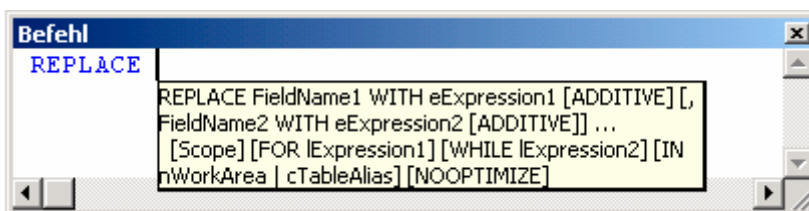
Mit der Taste „Ende“ platzieren Sie den Cursor am Ende des Schlüsselworts.

---

## Tipps zur Syntax von Befehlen und Funktionen

---

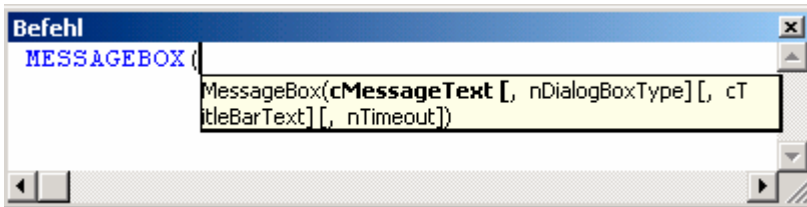
Nach der Eingabe von „repl“ und dem Drücken der Leertaste stellen Sie fest, dass außer der Anweisungsvervollständigung noch etwas geschehen ist: ein Fenster ähnlich einem Tooltip zeigt die vollständige Syntax des Befehls Replace an (vgl. Abbildung 2). Dieses Feature, von Microsoft QuickInfo genannt, erhöht die Produktivität in erheblichem Maße. Wie oft haben Sie schon die Hilfe zu VFP aufgerufen, weil Sie die exakte Syntax für Alter Table nicht im Kopf hatten oder nicht mehr wussten, ob der Suchstring AT() als erster oder zweiter Parameter übergeben wird?



**Abbildung 2.** Durch QuickInfo müssen Sie nicht die Hilfe aufrufen, da die komplette Syntax für Befehle und Funktionen direkt angezeigt wird.

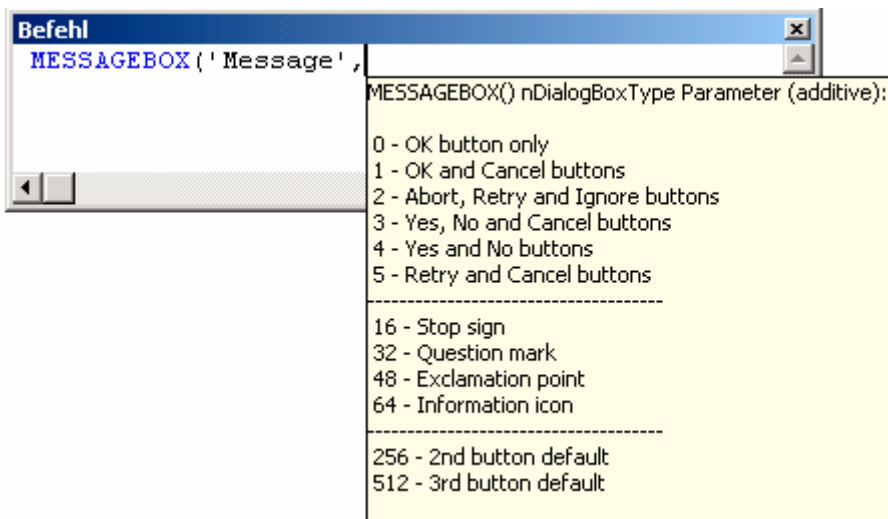
Das Fenster mit dem Tipp bleibt auf dem Monitor stehen, während Sie den Rest des Befehls oder die Parameter einer Funktion, einer Methode oder eines Ereignisses eingeben und verschwindet erst, wenn Sie die Funktion mit „)“ beenden, in eine neue Zeile wechseln (durch Drücken auf „Enter“) oder die Löschtaste betätigen (in diesem Fall kommt das Fenster wieder, wenn Sie mit

dem Cursor auf das nächste Element des Befehls wechseln). Sie können es manuell durch Drücken von Esc schließen und mit Strg-I oder dem Menüpunkt Bearbeiten – QuickInfo erneut aufrufen. Besonders hilfreich ist es bei Funktionen, Methoden und Ereignissen: der Parameter, den Sie gerade eingeben, wird in fett angezeigt (vgl. Abbildung 3).



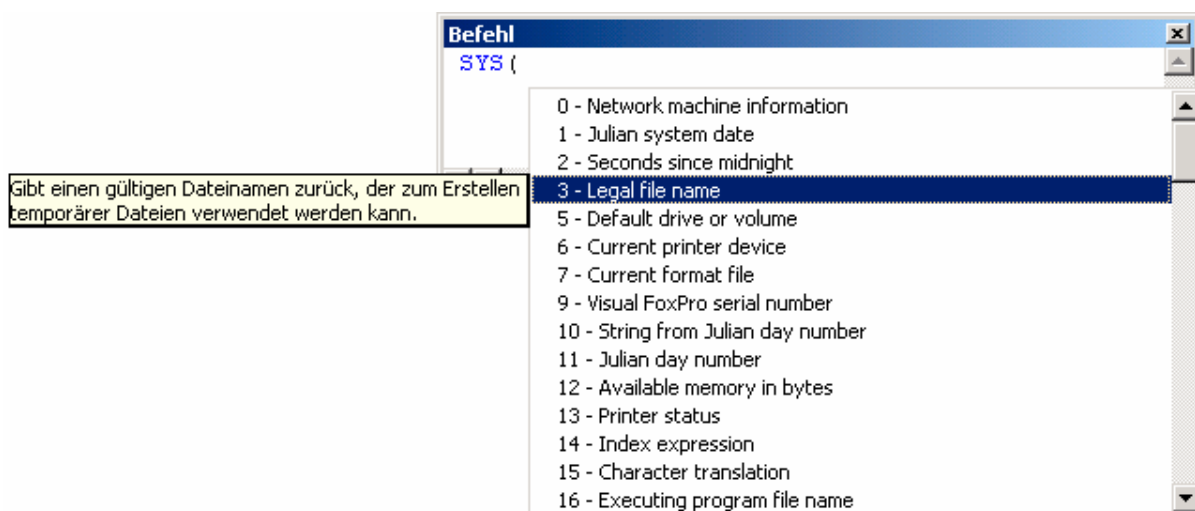
**Abbildung 3.** Der von Ihnen gerade eingegene Parameter erscheint im QuickInfo-Fenster in fetter Schrift.

Bei einigen Funktion zeigt das QuickInfo-Fenster Informationen über mögliche Werte des speziellen Parameters an. Ein Beispiel: der zweite Parameter von `MessageBox()` ist ein zusätzlicher Wert für Schaltflächen und Icons des Dialogs. Wie Sie in Abbildung 4 sehen zeigt IntelliSense die vollständige Liste gültiger Werte für diesen Parameter an. Andere Funktionen akzeptieren bei manchen lediglich einen Wert aus einer Liste vordefinierter Werte. Auch hier ein Beispiel: Der zweite Parameter von `DBGetProp()` zeigt den Typen des Datenobjekts (Verbindung, Datenbank, Feld, Tabelle oder Ansicht) an und die Liste der Werte für den dritten Parameter, die Eigenschaft ändert sich mit dem Typ des Objekts (z. B. ist `DefaultValue` nur für Felder verfügbar). Für den Typparameter zeigt IntelliSense eine Liste der Objekttypen an. Wählen Sie den richtigen Typen aus, und IntelliSense fügt ihn komplett mit Anführungszeichen in die Befehlszeile ein. Die Liste der Werte, die für den Eigenschaftsparameter angezeigt wird, enthält nur die für diesen Typen passenden. Wählen Sie wieder eine Eigenschaft aus der Liste, um diese der Befehlszeile hinzuzufügen.



**Abbildung 4.** Das Fenster mit den Tipps zeigt bei manchen Funktionen detaillierte Informationen über den aktuellen Parameter.

Die Funktion `SYS()` ist ein Spezialfall davon. Obwohl es sich dabei nur um ein Schlüsselwort handelt, besteht `SYS()` doch aus einer Vielzahl an Funktionen. Um welche Funktion es sich handelt, wird durch den ersten Parameter festgelegt. Wie Abbildung 5 zeigt, stellt IntelliSense für diesen Parameter eine Liste dar, nicht nur die numerischen Werte, sondern auch deren Bedeutung. Nachdem Sie den von Ihnen gewünschten ausgewählt haben, zeigt das Fenster mit den Tipps die Syntax für den restlichen Funktionsaufruf.



**Abbildung 5.** IntelliSense macht es einfach herauszufinden, welcher Wert `SYS()` für welche Funktion übergeben werden muss.

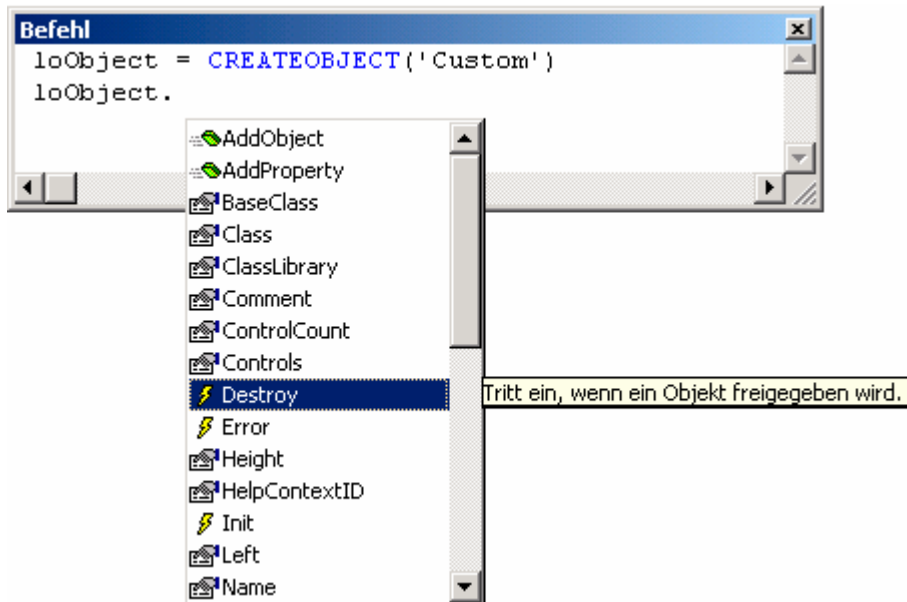
---

## Auflisten von Elementen

---

Im heutigen objektorientierten Code spart Ihnen die Funktion des Auflistens von Elementen mehr Schreibarbeit als alle anderen. Wenn Sie einen Objekt-namen mit nachfolgendem „,“ eingeben, zeigt VFP Ihnen eine Liste aller E-

lemente (Eigenschaften, Methoden, Ereignisse und eingebettete Elemente) des Objekts und ein Icon zeigt den Typ des Elements an. Abbildung 6 zeigt Ihnen ein Beispiel, in denen Ihnen die Elemente eines neu instanziierten Objekts Custom angezeigt werden.



**Abbildung 6.** Das Auflisten von Elementen von IntelliSense zeigt die in einem Objekt enthaltenen Elemente.

Innerhalb der Liste können Sie sich mit den Pfeiltasten, Seite-auf- und Seite-ab-Tasten sowie mittels inkrementeller Suche bewegen, indem Sie die ersten Buchstaben des gesuchten Elements eingeben (die eingegebenen Buchstaben erscheinen mit Verzögerung auch in der Befehlszeile.). Während Sie sich in der Liste bewegen, zeigt Ihnen ein Tooltip die Beschreibung jedes Elements und erspart Ihnen wieder das Suchen in der Hilfedatei.

IntelliSense fügt den Namen des ausgewählten Elements der Befehlszeile hinzu und beendet die Liste, wenn Sie die Tabulatortaste (der Cursor erscheint rechts neben dem Elementnamen), Enter (der Cursor rutscht in die nächste Zeile) oder ein Zeichen aus der Interpunktion wie „.“, „=“ oder „(, (dieses Zeichen erscheint hinter dem Elementnamen) eingeben. Das Betätigen der Tasten „Pos 1“, „Ende“, sowie die rechten und linken Pfeiltasten beendet die Liste, ohne der Befehlszeile das Element hinzuzufügen. Durch Drücken von Strg-J oder den Menüpunkt Bearbeiten – Elemente auflisten zeigen Sie die Liste manuell wieder an.

Das Auflisten von Elementen funktioniert auch in Objekthierarchien. Bedenken Sie einmal die Tipparbeit, wenn Sie nur programmatisch die Überschrift des Headers der dritten Spalte eines Grid auf Seite 2 eines Pageframes auf einem Formular ändern wollten:

```
Thisform.PageFrame1.Page2.Grid.Column3.Header.Caption = 'some value'
```

Ups, der Name des Grid ist `grdCustomers`. Daher wird der Code bei der Ausführung lediglich einen Fehler hervorrufen. Die Auflistung der Elemente erspart Ihnen sowohl Tippfehler als auch die ganze Eingabe des Code. Nachdem Sie den Punkt hinter `Thisform` eingegeben haben, wählen Sie aus der Liste `PageFrame1` und drücken „,“, um die Elementliste für den Pageframe anzeigen zu lassen, wählen `Page2` und drücken „,“, um die Elementliste für die Seite anzeigen zu lassen und so geht es bis zur `Caption` des Header weiter, die Sie auswählen und „=“ eingeben um die Liste zu schließen.

Die Liste der Elemente funktioniert nicht mit den nativen Objekten von VFP, sondern nur mit instantiierten Objekten wie dem ActiveX-Kontrollelement `TreeView`, das Sie einem Formular oder einem COM-Objekt hinzugefügt haben, das Sie mit `CreateObject()` erstellt haben. Anders als das IntelliSense von Visual Basic unterstützt VFPs Version Elementlisten für Objekte in Collections (wie der `Collection Cells` von Excel) und hört nicht bei der Collection selbst auf. Noch besser: es kann mit Objekten arbeiten, die Sie zur Zeit nicht instantiiert haben. Damit kommen wir zum nächsten Punkt.

---

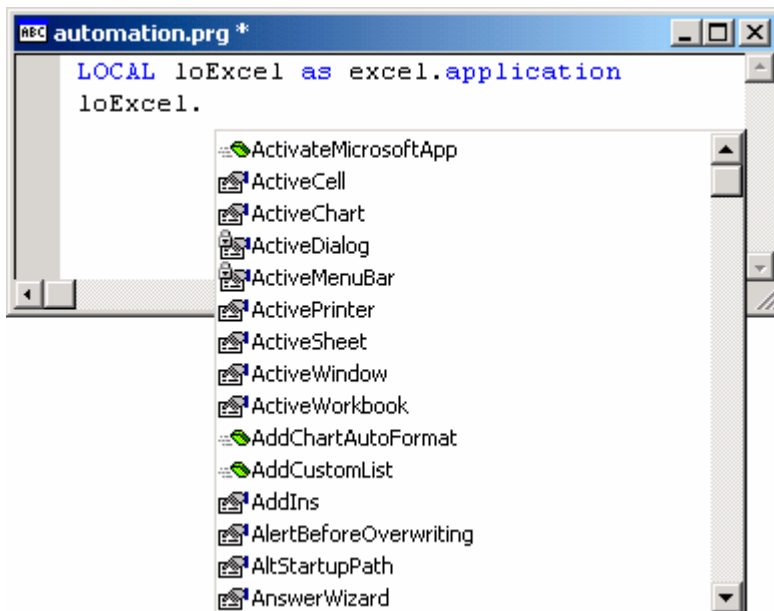
## Das frühe Binden

---

VFP ist für COM-Objekte ein Client mit spätem Binden. Das bedeutet, dass VFP während der Entwicklung (im Editor und während des Kompilierens) keine Ahnung hat, welche Art von Objekten Sie instanziiieren, wenn Sie `CreateObject()` aufrufen. Nur wenn der Code ausgeführt wird findet VFP heraus, welche Eigenschaften, Ereignisse und Methoden ein Objekt besitzt. Dafür wird die Typenbibliothek für das Objekt gelesen. Die Typenbibliotheken werden in Kapitel 12, „Building World-Class COM Servers in VFP 7“ behandelt. Clients, die das frühe Binden unterstützen, Visual Basic gehört dazu, lesen die Typenbibliothek statt dessen während der Entwicklungszeit. Das hat verschiedene Vorteile, so kann IntelliSense die Elemente eines COM-Objekts im Editor anzeigen. Das klingt erst einmal wie die Elementliste von VFP 7. Das Problem ist aber, dass VFP nicht weiß, dass Sie in Ihrem Code ein Excel-Objekt instanziiert haben. Wie sollen dann nun die Elemente von Excel aufgelistet werden?

Um dieses Problem zu lösen, hat Microsoft den Befehlen `Local` und `Public` die Klausel `As` gegeben (die Klausel gibt es auch in `Parameters`, `LParameters`,

Function und in Prozedurdeklarationen, aber dort findet sie sich aus einem anderen Grund. Vgl. Sie dazu den Abschnitt „Strong typing“ in Kapitel 12, „Building World-Class COM Servers in VFP 7“). Diese Klausel ermöglicht es Ihnen, eine Klasse des Objekts anzugeben, die eine Variable zur Laufzeit enthalten wird. Wenn Sie den Variablennamen mit dem anschließenden Punkt im Editor von VFP eingeben, sieht IntelliSense, dass die Variable vom „Typ“ Klasse ist, nimmt die Elemente dieser Klasse (bzw. liest die Typbibliothek, falls es sich um ein COM-Objekt handelt) und zeigt die Elementliste an. Abbildung 7 zeigt die Elemente der Variablen loExcel, die als Excel.Application-Objekt deklariert wurde. Beachten Sie, dass der Code keine Anweisung CreateObject() enthält, so dass es zur Laufzeit nicht ausgeführt wird. Die Klausel AS instanziiert das Objekt nicht – das müssen Sie in Ihrem Code machen – sie teilt IntelliSense lediglich mit, wie die Variable im Editor behandelt werden soll.



**Abbildung 7.** IntelliSense kann Elemente von COM-Objekten anzeigen, wenn Sie eine Variable "AS" das COM-Objekt deklarieren.

Wenn Sie das Schlüsselwort AS eingegeben und die Leertaste gedrückt haben, erscheint eine Liste mit Typen. Diese Liste enthält die Datentypen von VFP (wie Integer und Numerisch), die Basisklassen (einschließlich Form und Custom), registrierte Typbibliotheken (z. B. ADODB und Excel), Ihre eigenen Klassen sowie Webdienste. Wie viele Dinge in VFP ist auch diese Typenliste in einer Tabelle in einer Tabelle definiert, so dass Sie Typbibliotheken, Klassen, Webdienste und andere Typen selbst hinzufügen können. Detaillierte Informationen dazu erhalten Sie weiter hinten in diesem Kapitel im Abschnitt „Configuring IntelliSense“.



Dieses neue Feature hat eine Konsequenz: Sie können Variablen in Local- und Public-Anweisungen nicht mehr durch Leerzeichen trennen. Die folgende Schreibweise war bislang möglich:

```
Local MyVar1 MyVar2
```

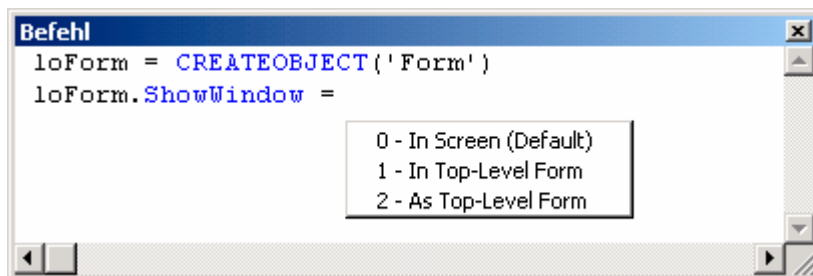
Da die meisten Entwickler diese Schreibweise nicht kannten (die Hilfedatei zeigt immer die korrekte Syntax, in der die Variablen durch Kommata getrennt werden), betrifft diese Änderung nicht viel Code.

---

## Wertlisten

---

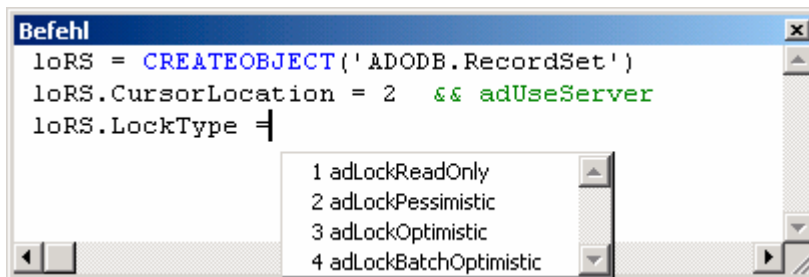
Einige Eigenschaften in Objekten akzeptieren nur eine kleine Anzahl an Werten. So ist z. B. Form.AutoCenter vom Typ Logisch, so dass sie nur die Werte True und False akzeptiert. Object.BackColor akzeptiert einen großen Bereich numerischer Werte, aber diese Werte repräsentieren jeweils eine Farbe, von denen nur wenige häufig gebraucht werden. Leider ist es nicht einfach, die Nummer für eine bestimmte Farbe herauszufinden. IntelliSense erleichtert die Auswahl des richtigen Wertes für diese Eigenschaft. Wenn Sie hinter manchen Eigenschaftsnamen „=“ eingeben, zeigt IntelliSense eine Liste von gültigen Werten für diese Eigenschaften, vergleichen Sie Abbildung 8 für ein Beispiel.



**Abbildung 8.** Die Wertlistenfunktion macht es einfach, den richtigen Wert für eine Eigenschaft zu ermitteln.

Bei logischen Eigenschaften wie AutoCenter und AlwaysOnTop wird eine Liste mit True und False angezeigt. Numerische Eigenschaften wie BorderStyle, ScrollBars und WindowType zählen in ihrer Wertliste mögliche Werte und deren Bedeutung auf. Bei Eigenschaften mit komplexeren Werten zeigt IntelliSense einen passenden Dialog an, beispielsweise eine Farbauswahl bei Eigenschaften, durch die Farbwerte repräsentiert werden (wie BackColor, FillColor und ForeColor) oder einen Dialog „Grafik öffnen“ für Eigenschaften, die einen Namen einer Bilddatei enthalten (z. B. Icon und Picture). Wie auch beim Auflisten von Elementen unterstützen die Wertlisten sowohl COM-Objekte wie auch native Objekte; IntelliSense zeigt eine Wertliste für solche Eigenschaften an, die in der Typbibliothek definiert sind. Ab-

bildung 9 zeigt, wie sinnvoll dieses Feature mit einem ADO RecordSet ist, der numerische Werte in vielen Eigenschaften nutzt. Beachten Sie die Codezeile, die über der steht, durch die die Liste angezeigt wird. IntelliSense hat nicht nur den Wert eingefügt, sondern auch einen Kommentar, der einen Namen für den Wert darstellt. Dadurch wird Ihr Code erheblich lesbarer.



**Abbildung 9.** Wertlisten funktionieren auch mit numerischen Eigenschaften von COM-Objekten.

Einige Eigenschaften, bei denen Sie denken werden, sie sollten über eine Wertliste oder einen Dialog verfügen, haben keine. Obwohl sie einen Dateinamen enthalten, zeigt IntelliSense für DragIcon, MouseIcon und OLEDragPicture keinen Grafik öffnen-Dialog an. Sie sind vielleicht überrascht, für die Eigenschaft FontName keinen Dialog zu erhalten. Weiter hinten in diesem Kapitel werde ich Ihnen zeigen wie Sie IntelliSense Ihren Wünschen entsprechend erweitern können.

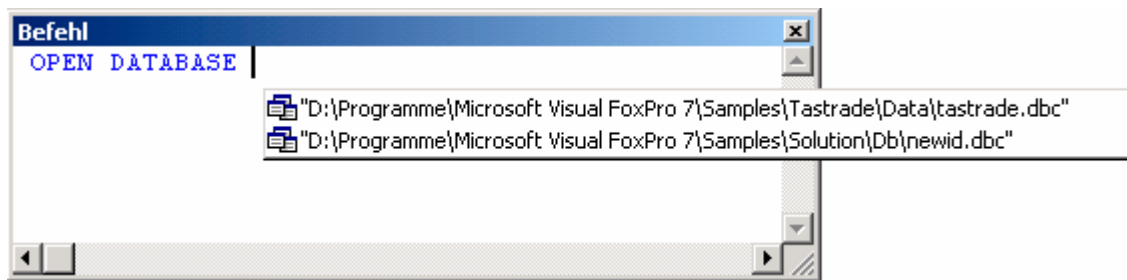
Für die Wertlisten gibt es keinen eigenen Menüeintrag oder Hotkey, der Menüeintrag Elemente auflisten und der Hotkey Strg-J behandeln auch diese Funktionalität.

---

## Die MRU-Liste

---

Einige VFP-Befehle öffnen Dateien oder führen diese aus. Beispiele dafür sind die Modify-Anweisungen (z. B. Modify Program), Open Database und Report Form. IntelliSense präsentiert eine Liste der zuletzt genutzten (most recently used = MRU) Dateien und erleichtert es auf diese Weise, eine Datei zu öffnen oder auszuführen, mit der Sie bereits einmal gearbeitet haben. Abbildung 10 zeigt diese Liste in Aktion. Sie erhalten auch mit dem Befehl CD eine Liste von Verzeichnissen. Beachten Sie, dass dieses Feature nur im Befehlsfenster unterstützt wird. Vergleichen Sie den Eintrag „Automatische MRU-Dateien (Most Recently Used, zuletzt verwendet) in der Hilfedatei von VFP, um eine vollständige Aufzählung der Befehle mit MRU-Liste zu erhalten.



**Abbildung 10.** Mit der MRU-Liste von IntelliSense ist es einfach, die Dateien auszuwählen, die Sie zuvor bereits verwandt haben.

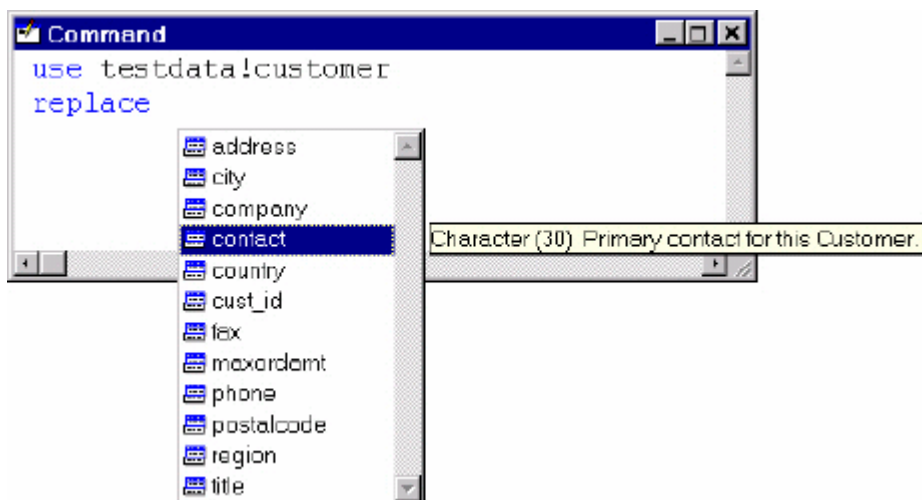
Die Seite Umgebung des Optionen-Dialogs von VFP enthält ein Feld, in dem Sie einstellen können, wie viele Dateien in der MRU-Liste angezeigt werden sollen.

---

## Tabelle, Feld und Variablenliste

---

IntelliSense erweitert die MRU-Liste für den Befehl Use: zusätzlich zu den Tabellen, die Sie vorher geöffnet haben, listet es die Tabellen und Ansichten in der aktuellen Datenbank auf. Die Liste hat ein Tooltip-Fenster, in dem der Kommentar zur gewählten Tabelle oder Ansicht angezeigt wird. Außerdem erweitert IntelliSense die Wertliste für die Befehle Replace, Modify Memo und Modify General: Wenn im aktuellen Arbeitsbereich ein Cursor geöffnet ist, wird eine Liste der Felder im Cursor angezeigt. Es werden der Datentyp, die Größe, Überschrift und der Kommentar für das gewählte Feld angezeigt; sehen Sie sich Abbildung 11 an, um ein Beispiel zu sehen. Wenn Sie im Befehlsfenster „m.“ eingeben, zeigt IntelliSense eine Liste der deklarierten Variablen an. Ein Tooltip-Fenster zeigt für jede Variable den aktuellen Wert an. Wie auch bei der MRU-Liste arbeiten diese Features nicht im Editor.



**Abbildung 11.** IntelliSense zeigt eine Liste der Felder des aktuellen Cursor für einige Befehle an.

