

---

# Feature-Überblick: Vista Toolkit

*Craig Boyd*

Microsoft® Windows Vista™ wird mit neuen Richtlinien für die Benutzeroberfläche ausgeliefert. Diese Richtlinien legen fest, wie eine Windows-Anwendung unter Windows Vista aussehen und wie sie sich verhalten soll.

---

## Gemeinsame Dialogklassen

---

### A) Einführung

---

Neben anderen Themen umfassen die Richtlinien auch die Verwendung der gemeinsam verwendeten Windows-Dialoge. Zu diesen Dialogen gehören Datei öffnen, Datei speichern, Verzeichnis öffnen, Suchen und Ersetzen, Drucken, Seiteneinrichtung, Schrift und Farbauswahl. Durch den Einsatz dieser Dialoge können Visual FoxPro-Entwickler von den vielen Erweiterungen profitieren, die diese neuen Dialoge anbieten, ihren Visual FoxPro-Anwendungen eine elegantere Erscheinung geben und den Windows-Anwendern eine konsistente Erfahrung bei der Arbeit mit ihren Anwendungen unter Windows Vista vermitteln.

Auch wenn die Dialoge Verzeichnis öffnen und Farbauswahl bereits durch die Funktionen `GetDir()` und `GetColor()` in Visual FoxPro 9.0 bereitgestellt werden, kann auf die anderen gemeinsamen Dialoge von Windows (siehe <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/UxGuide/UXGuide/Windows/CommonDialogs/CommonDialogs.asp>) bislang aus Visual FoxPro heraus nicht zugegriffen werden.

Das primäre Ziel dieses Projekts besteht darin, Visual FoxPro-Entwicklern verschiedene Wrapperklassen bereitzustellen, die eine einfache Möglichkeit beinhalten, die neuen gemeinsamen Dialoge in ihre Anwendungen zu integrieren und sie einzusetzen.

---

### B) Features

---

---

#### *Datei öffnen*

---

Klassenwrapper für den Dialog Datei öffnen, der verwendet werden kann, um die Dateien schnell zu finden und zu markieren, die verwendet werden sollen und um die Auswahl des Anwenders an Visual FoxPro zurückzugeben. Entspricht der Funktionalität der Funktion `GetFile()` von Visual FoxPro.

---

#### *Datei speichern*

---

Klassenwrapper für den Dialog Datei speichern, der verwendet werden kann, um Dateien zusammen mit den zu ihnen gehörenden Metadaten auf der Festplatte des Computers zu speichern. Entspricht der Funktionalität der Funktion `PutFile()` von Visual FoxPro.

---

#### *Suchen und Ersetzen*

---

Klassenwrapper für den Dialog Suchen und Ersetzen, der verwendet werden kann, um eine Suchen- und Ersetzen-Funktionalität innerhalb eines Textstrings auszuführen. Die Suche kann die Groß- und Kleinschreibung beachten oder ignorieren.

---

### *Drucken*

---

Klassenwrapper für den Dialog Drucken, der verwendet werden kann, um die Benutzereingaben für den Drucker, die Seiten, die Kopienanzahl sowie die Druckreihenfolge entgegenzunehmen. Zusätzliche Drucker können konfiguriert werden.

---

### *Seiteneinrichtung*

---

Klassenwrapper für den Dialog Seiteneinrichtung, der verwendet werden kann, um die Benutzereingaben für die Papiergröße, die Papierquelle, die Ausrichtung und die Ränder für einen Druckauftrag entgegenzunehmen.

---

### *Font*

---

Klassenwrapper für den Dialog Font, der verwendet werden kann, um die Benutzereingaben für die Auswahl aus den installierten Schriften, den Schriftstilen, die Größe in Punkt, der Farbe sowie der verfügbaren Effekte entgegenzunehmen. Entspricht der Funktionalität der Funktion `GetFont()` von Visual FoxPro.

---

## Windows Presentation Foundation

---

---

### A) Einführung

---

Einer der überraschendsten Aspekte der WinFX API – der verwalteten API der nächsten Generation – ist die Windows Presentation Foundation (WPF). WPF, vormals unter dem Namen „Avalon“ bekannt, bildet die Grundlage, die Entwickler nutzen können, um ihren Anwendern unter Windows Vista™ und Windows XP™ ganz neue Erfahrungen zu geben. Einzelheiten zu diesem Thema finden Sie unter <http://msdn.microsoft.com/winfx/reference/presentation/default.aspx>.

Obwohl WPF aus einem Satz vollständig gemanagter APIs besteht, steht die Funktionalität von WPF in Visual FoxPro noch nicht zur Verfügung. Visual FoxPro wird verwendet, um nicht gemanagte auf COM basierende Lösungen zu entwickeln, im Gegensatz zu den Assemblies von .NET. Mit Hilfe der Features von .NET für die Interoperabilität mit COM gibt es die Möglichkeit, innerhalb einer Visual FoxPro-Anwendung WPF zu verwenden oder auch innerhalb von WPF eine Visual FoxPro-Anwendung aufzurufen. Visual FoxPro und WPF können gleichberechtigt verwendet werden, um hybride Visual FoxPro-Anwendungen zu erstellen, die Funktionalitäten sowie visuelle Effekte aufweisen, die früher nicht möglich waren.

Dieses Projekt versucht, Klassen und Beispiele bereitzustellen, die Visual FoxPro-Entwicklern helfen, hybride Anwendungen zu erstellen, die sowohl bestehenden Visual FoxPro-Code verwenden als auch gleichzeitig Möglichkeiten bieten, auf die überraschenden neuen Eigenschaften der WPF zuzugreifen.

---

### B) Ziele

---

Dieses Projekt adressiert drei Ziele:

- **Von den Features profitieren, die WPF einer Visual FoxPro-Anwendung zur Verfügung stellt.** Eines der besten Features von Visual FoxPro ist seine Erweiterbarkeit. Visual FoxPro-Entwickler verwenden ständig andere Microsoft-Technologien, um die Anwender mit mächtigen datenzentrierten Anwendungen zu versorgen. WPF ist da keine Ausnahme. Die Features und Funktionalitäten, die WPF mittels Crossbow bereitstellt, können verwendet werden, um Visual FoxPro-Anwendungen zu erstellen, die XAML-Steuerelemente enthalten, die mit Lösungen arbeiten, die dieses Projekt präsentiert.
- **In einer WPF-Anwendung von den Features profitieren, die Visual FoxPro zur Verfügung stellt.** Die Features des .NET Frameworks für die COM-Interoperabilität stellen Visual FoxPro-Entwicklern eine Möglichkeit bereit, innerhalb einer WPF-Anwendung Visual Fox-

Pro-Code aufzurufen. Dieses Projekt ermöglicht es, die Mächtigkeit von Visual FoxPro und seiner Cursor-Engine als Teil einer effektiven WPF-Lösung einzusetzen.

- **Ermöglichen einer schrittweisen Migration einer Visual FoxPro-Anwendung nach WPF.** Visual FoxPro-Entwickler, die vorhaben, eine bestehende Visual FoxPro-Anwendung nach WPF zu migrieren, können der Anwendung Features von WPF hinzufügen und so einige der in diesem Projekt vorhandenen Techniken ersetzen. Dadurch werden die Risiken vermindert und es ist möglich, die Anwendung im Laufe der Zeit Stück für Stück umzustellen.

---

## C) Features

---

### *XAML-Steuerelemente in Visual FoxPro*

---

Demonstriert die Schritte, die erforderlich sind, um mit Visual Studio 2005 ein XAML-Steuerelement zu erstellen, das in einer Visual FoxPro-Anwendung verwendet wird. Beispiele und Erklärungen werden bereitgestellt.

### *Visual FoxPro COM-Objekte in WPF*

---

Zeigt, wie Visual FoxPro COM-Objekte erstellt und in einer WPF-Anwendung oder einem WPF-Steuerelement eingesetzt werden. Beispiele und Erklärungen werden bereitgestellt.

---

## Windows Desktop Search

---

### A) Einführung

---

Windows Desktop Search (WDS) umfasst das schnellstmögliche Finden von Informationen auf einem PC. WDS wird von Microsoft kostenfrei zur Verfügung gestellt und kann eingesetzt werden, um Visual FoxPro-Anwendungen erweiterte Suchmöglichkeiten zu ermöglichen. Einzelheiten finden Sie unter der URL <http://www.microsoft.com/windows/desktopsearch/default.aspx>.

Visual FoxPro beherrscht die Arbeit mit Informationen bereits und ermöglicht es den Anwendern, diese Informationen so effizient wie möglich zu suchen. So gut Visual FoxPro in dieser Hinsicht auch ist, ist das Erstellen einer Anwendung, die es dem Anwender ermöglicht, schnell und effektiv nach Informationen in E-Mails, Kalendereinträgen, Photos, Dokumenten und anderen Dateiformaten zu suchen, die auf dem Rechner gespeichert sind, im Bestfall eine sehr schwierige Aufgabe. Durch den Einsatz der Flexibilität und Geschwindigkeit von WDS können Visual FoxPro-Entwickler diese Art der Funktionalität in ihren Anwendungen relativ leicht anbieten. Die dafür erforderlichen Mechanismen sind nicht viel komplexer als das Schreiben einer Standard-SQL-Anweisung.

WDS kann durch das Erstellen von Add-Ins erweitert werden. Auch wenn diese Add-Ins für WDS nicht direkt in Visual FoxPro erstellt werden können, zeigt dieses Projekt einige Möglichkeiten, wie Add-Ins erstellt und genutzt werden können, um WDS zusätzliche Dateitypen hinzuzufügen. Außerdem wird demonstriert, wie Visual FoxPro-Befehle durch WDS ausgeführt werden. Dadurch sind Visual FoxPro-Entwickler in der Lage, WDS in eine Visual FoxPro-Anwendung zu integrieren.

Dieses Projekt versucht, Visual FoxPro-Entwickler mit einem Set gemanagter und nicht gemanagter Klassen und Beispiele zu versorgen, die es ihnen ermöglichen, die Power der WDS API auszunutzen, um Abfragen zu starten und Ergebnismengen mit Informationen zurückzugeben, die auf dem Rechner des Anwenders in unterschiedlichen Datenformaten vorhanden sind. Außerdem soll das Projekt einige der möglichen Einsatzbereiche für WDS Add-Ins in Visual FoxPro-Anwendungen aufzeigen, sowie das Erstellen und Verwenden dieser Add-Ins.

---

## B) Ziele

---

Dieses Projekt adressiert zwei Ziele:

- **Erstellen von Visual FoxPro-Anwendungen, die den Anwendern die Möglichkeit bereitstellen, auf ihrem Computer Informationen aus verschiedenen Quellen zu suchen.**

Mit Hilfe der Interoperabilität von .NET und der API von WDS können Visual FoxPro-Anwendungen Anfragen an WDS senden. Die Ergebnismenge kann anschließend als Recordset, XML oder als einfacher Cursor von Visual FoxPro zurückgegeben werden, um in einer Visual FoxPro-Anwendung weiterverarbeitet oder angezeigt zu werden. Die Bandbreite der gesuchten Information wie auch die Geschwindigkeit, in der die Ergebnisse zurückgeliefert werden, wäre mit Visual FoxPro allein schwer zu erreichen.

- **Erstellen und Verwenden der Add-Ins von WDS, um WDS zusätzliche Funktionalitäten hinzuzufügen, die für eine Anwendung spezifisch sind in der Art, in der die Anwendung ihre Daten speichert.**

WDS umfasst ein Erweiterungsmodell, das es Entwicklern ermöglicht, Add-Ins für WDS zu erstellen, mit denen die Fähigkeiten von WDS erweitern lassen, Daten zu suchen und zu empfangen. So gesehen dient WDS dazu, neue Dateitypen und Speicherarten zu verstehen, und um verschiedene Aktionen durchzuführen, wenn auf Elemente in den Suchergebnissen doppelgeklickt wird. Diese Erweiterbarkeit wurde durch das Erstellen der Handler IFilters und Protocol möglich.

---

## C) Features

---

*Klassen für die Abfrage von Informationen via WDS*

Ausführen von Abfragen über die WDS API mit einer gemanagten Klasse.

*Beispiele für Add-Ins in WDS sowie für deren Einsatz*

Beispiele für das Erstellen, Installieren und Verwenden von Add-Ins für WDS. Einfache IFilters und Protokollhandler werden bereitgestellt.

---

## RSS

---

### A) Einführung

---

Die Verwendung von RSS, ein Format für die Verbreitung von News und Information, wurde in letzter Zeit immer wichtiger. Daher hat Microsoft in Windows Vista verschiedene RSS-Plattformkomponenten integriert, die es Entwicklern ermöglichen, RSS Feeds schnell und einfach zu verarbeiten. Einzelheiten erfahren Sie unter der URL <http://msdn.microsoft.com/windowsvista/reference/rss/>.

Unified Feed Parsing API, Shared Feed List (IFeeds) und Automatic Download for Encllosures versorgen die Entwickler mit den Werkzeugen, die sie benötigen, um RSS-fähige Anwendungen zu erstellen. Zusammen mit diesen Werkzeugen können die XML-Features von Visual FoxPro die gesamte Power von RSS nutzen.

Dieses Projekt versorgt Visual FoxPro-Entwickler mit Klassen und Beispielen, die zeigen, wie diese neuen Plattformkomponenten zu RSS effektiv eingesetzt werden, um mit Visual FoxPro auf einfache Weise RSS-fähige Anwendungen zu erstellen. Außerdem zeigt dieses Projekt, wie Visual FoxPro-Entwickler Daten in Visual FoxPro als RSS formatieren können.

---

## **B) Features**

---

### *Klassenwrapper für RSS-Plattformkomponenten*

---

Entwickler werden mit einem Satz Klassen versorgt, die es Visual FoxPro-Anwendungen ermöglichen, RSS Feeds in Form von XML oder Cursors zu verarbeiten. Dazu gehören das Abonnement von Feeds, das Suchen der Inhalte dieser Feeds sowie der Zugriff darauf.

### *Klassen für das Bereitstellen von Visual FoxPro-Daten als RSS*

---

Ein Satz Klassen, der es Entwicklern ermöglicht, in Visual FoxPro Daten zu lesen (als Cursor, Ansicht, Tabelle, XML usw.) und als RSS zu formatieren.