
Deutsche VFP Entwicklerkonferenz 2006

von Jan Vit und Jan Král

Die deutsche Visual FoxPro-Entwicklerkonferenz wird durch die deutschsprachige Fox-Pro User Group organisiert. Sie ist seit 1994 die wichtigste Veranstaltung zu Visual FoxPro im deutschsprachigen Raum. 2006 fand die 13. Konferenz statt – wie immer Anfang November und erneut mit einer Online-Berichterstattung auf Universal Thread. Statt die Anzahl der Vorträge weiter zu erhöhen, haben wir uns entschieden, den Preis zu senken. Die internationalen Sprecher hielten ihre Vorträge in unterschiedlichen Sessions. Weitere Vorträge wurden in speziellen Tracks mit insgesamt vier Dutzend englischsprachigen Sessions gehalten. Während der gesamten vier Tage der Konferenz konnten die Teilnehmer jederzeit zwischen mehreren englischsprachigen Sessions wählen.

Tag Null

Mission Possible – Prag 6:00 Uhr morgens



Ich, Jan und natürlich ich besteigen in Prag das Auto und die Reise hatte begonnen. Nach zehn Minuten trat die erste Schwierigkeit auf. Wir hatten uns verfahren. Es hatte den Anschein, dass es ein kleines Problem sein könnte, nach Frankfurt zu kommen, wenn wir nicht einmal den richtigen Weg aus unserer Heimatstadt herausfinden können. Nach etwa einer halben Stunde hatten wir es aber dank meiner Führung geschafft, auf die richtige Autobahn aufzufahren. Ich muss sagen, dass nahezu jeder Fehler, den wir begangen haben, auf das Versagen des Fahrers zurückzuführen war (er ist

derjenige, der fährt und die Entscheidungen trifft). Hoffentlich liest er diesen Abschnitt nicht; das könnte mein Ende sein.

Als wir die deutsche Grenze erreichten, wurde uns klar, dass es an der Zeit war, eine Straßenkarte zu kaufen. Wir hielten also an und Jan sagte mir, ich solle aussteigen, um eine gute Karte zu kaufen. Nach zehn Minuten fand ich eine Karte und zahlte auch das Benzin. Das wäre kein Problem gewesen, wenn meine Kreditkarte korrekt funktioniert hätte. Computer! Zum Glück hatte ich noch eine zweite Karte. Wir nahmen noch einen kleinen Snack und passierten die Grenze ohne weitere größere Probleme. Sicherheitshalber diskutierten wir kurz mögliche Pannen, beispielsweise einen geplatzen Reifen bei Tempo 140 und die daraus entstehenden Konsequenzen.

Im Laufe der Zeit kamen wir Frankfurt immer näher. Ich spielte mit dem Gedanken, noch etwas zu schlafen, aber wissen Sie noch, was ich über den Fahrer geschrieben habe? Dann ist Ihnen ja klar, dass ich letztendlich entschied, wach zu bleiben. Als Jan mir mitteilte, dass wir etwas hinter dem Zeitplan zurückhängen und daher kräftiger aufs Gaspedal trat, zögerte ich nicht lange, schnallte mich an und begann, meine Kenntnisse in Erster Hilfe aufzufrischen. Sie können sich nicht vorstellen, wie schnell ich meine Erinnerungen aufgefrischt habe.

Wenn ich ganz ehrlich sein soll, muss ich zugeben, dass nicht alles die Schuld des Fahrers war. Ich weiß nicht weshalb, aber irgendwie war ich der Meinung, dass es einfacher wäre, nach Frankfurt über Heilbronn zu fahren. Auch wenn Jan meine Meinung nicht teilte und über Würzburg fahren wollte, setzte ich meine Macht als Navigator ein und wies ihn an, die von mir gewählte

Strecke zu fahren. Später wurde es offensichtlich, dass Jan Recht hatte. Ich versuchte trotzdem, meine frühere Entscheidung auf jede mögliche Weise zu verteidigen. Als letztes Argument fiel mir ein, dass es von Vorteil wäre, eine schöne Landschaft im Raum Heilbronn zu sehen, aber als Jan ausführte, dass selbstverständlich bei Nacht nichts zu sehen wäre, so gegen 1:00 Uhr und besonders wenn wir müde sind, gingen mir die Argumente aus. Ob Tag oder Nacht hätten wir nur von der Autobahn aus einen Blick auf Heilbronn werfen können.

Wir freuten uns darauf, alle auf der Konferenz zu sehen und hofften, in Frankfurt eine gute Zeit zu haben.

Donnerstag, 9. November

von Jan Vit und Jan Král

Herzlich willkommen zur 13. dFPUG Microsoft Visual FoxPro-Konferenz!

Wenn Sie diese Zeilen lesen, ist Ihnen klar, dass wir das Hotel schließlich doch noch sicher erreicht haben und bereit sind, von diesem großartigen Ereignis zu berichten.

Neben der Visual FoxPro DevCon findet parallel noch eine SQL Server und .NET DevCon statt. Sie ist einen Tag kürzer und Sie finden einen Bericht über diese DevCon unter <http://www.ut-coverage.com/German/20062>.

Willkommens-Session – Rainer Becker

Nach einem reichhaltigen Frühstück versammelten sich alle Teilnehmer im größten Konferenzraum und Rainer begann mit seiner Willkommens-Session. Neben News aus der Community, beispielsweise die aktuelle CTP für Sedna und das Service Pack für VFP 9.0 erwähnte er auch einige interessante Details zur Konferenz.



Zunächst verglich er die Teilnehmerzahlen. In den Jahren 2004 und 2005 zählte die Konferenz etwa 200 Teilnehmer. In diesem Jahr verringerte sich diese Zahl um etwa 15 % auf ungefähr 170 Teilnehmer. Woran liegt das? Nun, es ist klar, dass eine neue VFP-Version immer ein guter Grund ist, eine Konferenz zu besuchen oder zumindest einen Angestellten dorthin zu senden. Da wir uns derzeit noch in einer CTP-Phase befinden, könnte dies laut Rainer der Grund für die etwas geringere Teilnehmerzahl sein.

Trotzdem steht die Zukunft der Konferenz nicht in Frage. Rainer führte aus, dass die Konferenztasche jedes Teilnehmers eine Liste mit den Daten aller dFPUG-Konferenzen in der Vergangenheit und der Zukunft enthält. Sie beginnt mit der ersten Konferenz 1994 und reicht über die diesjährige Konferenz (die 13.) bis zur 21. Konferenz im Jahr 2014. Mit Hilfe dieser Liste können Sie Ihren nächsten Besuch dieser Konferenz planen.

Mehr noch: Die 21. Konferenz wird auf jeden Fall stattfinden. Sie überschneidet sich mit Rainers 50. Geburtstag und es gibt für ihn vermutlich keinen besseren Platz, dieses Ereignis zu feiern als in der Community.

Best Practices Error Handling – Rick Schummer

Rick begann seine Session mit einer Demonstration des inspirierenden Designs seiner Dialoge für die Fehlerbehandlung in seiner Real-World-keine-Demo-Anwendung namens „Bug“. Auch wenn es unglaublich klingt, handelt es sich um eine Anwendung, die Rick entwickelt hat und die in einem Unternehmen eingesetzt wird, das sich auf das Entfernen der echten und lebenden Bugs spezialisiert hat.

Der erste Teil der Session konzentrierte sich auf die Theorie der Fehlerbehandlung.

Als erstes erkannten alle Teilnehmer an, dass Fehler in unseren Programmen auftreten. Wir sind nur Menschen und Fehler zu machen ist Teil unserer Natur. Außerdem erkannte jeder an, dass es ein System geben sollte, das mit diesen Fehlern umgeht. Alle Fehler müssen behandelt werden, unabhängig davon, ob es sich um einen Fehler im Code oder lediglich um eine nicht erwartete Folge von Ereignissen handelt, die in der Anwendung auftritt und ein nicht erwartetes Verhalten auslöst.

In VFP gibt es fünf grundlegende Methoden für die Behandlung von Fehlern. Die erste Methode besteht darin, keine Fehlerbehandlung einzusetzen, indem der Befehl `ON ERROR *` verwendet wird. Glauben Sie es oder nicht, das ist alles, was Sie benötigen, um Ihre Anwendung fehlerfrei zu machen! Unabhängig davon, wie bequem diese Lösung sein mag, ist sie in der Regel in seriösen Anwendungen nicht zu akzeptieren. Das gilt auch für die zweite Methode, die darin besteht, den vorgegebenen Errorhandler von VFP mit den klassischen Schaltflächen Abbrechen, Wiederholen, Ignorieren und Hilfe zu benutzen. Es können in echten Anwendungen daher nur die globale Fehlerbehandlung mit dem Befehl `ON ERROR` gemeinsam mit Ihrer eigenen Fehlerbehandlungsroutine, die strukturierte Fehlerbehandlung mit `TRY...CATCH` oder das Ereignis `OnError` in Objekten verwendet werden.

Rick erklärte ausführlich die Ziele und Nachteile jeder Methode, zeigte einige Beispiele und sprach über die üblichen Kombinationen dieser Techniken. Außerdem sprach er über die in kommerziellen Frameworks häufig verwendete Fehlerbehandlungsstrategien.

So weit der erste theoretische Teil der Session. Der zweite Teil konzentrierte sich auf unterschiedliche Techniken, die helfen, Fehler festzuhalten, sie den Entwicklern mitzuteilen, sie zu verfolgen usw. Um es kurz zu machen beleuchtete die Session die Ecken, die wir als Entwickler manchmal gerne übersehen. Das Hauptziel der Session bestand darin, uns daran zu erinnern, was wir tun sollten, wenn nicht für unsere Kunden, dann wenigstens für uns.

Objektorientierung – Michael Niethammer



Dieser aus zwei Sessions bestehende Block war Teil des Tracks „Einführung in VFP 9.0“ und gab einen vollständigen Überblick über die Grundlagen der Objektorientierung. Michael erledigte einen großartigen Job, als er versuchte, alles so einfach wie möglich zu erklären, so dass jemand, der bislang noch keine Berührung mit der Objektorientierung hatte, über ein solides Verständnis dieses Themas verfügte, nachdem er diese Session verfolgt hatte. Die Session wurde häufig durch verschiedene Fragen der Teilnehmer unterbrochen. In diesen Fällen behielt

Michael immer seinen aktuellen Gedanken im Hinterkopf und fuhr an dieser Stelle fort, wenn er sich sicher war, dass die Frage vollständig beantwortet war.

Am Beginn der ersten Session erläuterte Michael die grundlegende Terminologie, also die Klasse, die Klassenbibliothek, die Instantiierung, das Objekt usw. und zeigte alles über die visuellen Klassen von VFP. Später behandelte er die Vererbung, die Container, die Eigenschaften und Methoden und am Ende anhand leicht verständlicher Beispiele die Ereignisse.

In der zweiten Session konzentrierte er sich auf die weitergehenden Details der bereits angesprochenen Themen wie Methoden und Ereignisse sowie auf neue Themen wie Access- und Assign-Methoden. Er kratzte außerdem an der Oberfläche vieler anderer Themen wie das Design der Komponenten und die Designmuster und erläuterte viele der klassischen Regeln der objektorientierten Entwicklung anhand anschaulicher Beispiele.

Da sich die Session eindeutig an Entwickler wandte, die noch immer den Großteil ihrer Zeit mit der strukturierten Programmierung verbringen, wandte Michael einen Teil der zur Verfügung stehenden Zeit für die Erklärung aller Vorteile der Objektorientierung auf, aber er behandelte auch die Nachteile, also dass die Umstellung auf die Objektorientierung eine steile Lernkurve erfordert. Ist diese Aufgabe erst einmal gemeistert, gibt die Objektorientierung aber nur noch Vorteile.

More than just your typical office automation – Marcia Akins



Visual FoxPro ist eine sehr mächtige Programmiersprache. Stellen Sie sich vor, Sie verfügen über einen Server, auf dem eine FoxPro-Anwendung läuft, die Word, Excel oder andere Anwendungen steuert. Der Anwender greift auf Ihre Anwendung zu und Ihre Anwendung steuert, was der Anwender mit Word tun darf, wo er die Dokumente speichern kann, ob er die Schrift ändern darf und andere Dinge.

Marcia begann nicht mit einem Vortrag über Automation Server, sondern sie zeigte uns ein wunderbares Video, das einer ihrer Freunde gedreht hat, so dass während der gesamten Session eine tolle Atmosphäre herrschte.

Wenn Sie Angst hatten, dass das Thema zu schwierig wird, lagen Sie falsch, da Marcia wusste, dass nicht alle Teilnehmer Professionals sind. Sie begann mit den Grundlagen der Automation. Am Anfang müssen wir eine neue Instanz erstellen (`CREATEOBJECT("Word.application")`). Marcia führte zwei Dinge aus. Nach dem Erstellen einer neuen Instanz müssen wir unser Objekt sichtbar machen, da das Objekt standardmäßig unsichtbar erstellt wird. Noch wichtiger ist, dass das Erstellen immer neuer Instanzen die Anwendung verlangsamt und dass es besser ist, bestehende Instanzen mit dem Befehl `GETOBJECT()` erneut zu verwenden.

Marcia zeigte uns, wie der Objektbrowser verwendet wird, um die Eigenschaften und Methoden eines Automation Servers zu untersuchen. Sie erzählte, dass der Objektbrowser unser bester Freund ist. Damit hat sie Recht. Der Objektbrowser zeigte uns den Speicherort einer Hilfedatei und half uns, die Methoden von Outlook auf sehr einfache Weise zu verwenden. Durch ein Drag-and-Drop aus dem Objektbrowser in Ihr Programmfenster wird eine `#DEFINE`-Anweisung generiert.

Am Besten ist aber, dass FoxPro, wenn Sie eine neue Instanz erstellen, in der Lage ist, IntelliSense zu verwenden und dass Sie das Ergebnis sofort erhalten, beispielsweise das Ändern des Titels einer Grafik in Excel oder das Ändern der Schrift oder des Schriftgrads. Mit IntelliSense haben Sie eine einfache Möglichkeit, mehr über Automation Server zu erfahren. Anschließend erwähnte Marcia noch das Event Binding sowie andere Themen.

Die Session war absolut großartig, voll mit vielen hilfreichen Themen und Beispielen, die Marcia bis ins kleinste Detail erklärte. Wir warten jetzt gespannt auf den nächsten Teil von „More than Office Automation“.

More than jus your typical office automation – Marcia Akins

Nicht lange danach hielt Marcia den zweiten Teil Ihrer Session. Wie gehofft begann Marcia mit einer netten Geschichte.

Die zweite Session begann mit einem anderen Microsoft-Programm, mit Map Point. Marcia zeigte uns, wie mit Map Point ein Automation Server erstellt wird. Zunächst müssen Sie mit `CREATEOBJECT()` eine Instanz erstellen und anschließend können Sie Methoden wie `FindAddress()` oder `FindPlaceResult()` verwenden. Marcia erklärte den Unterschied zwischen Route (auf WayPoints basierend), WayPoint (auf PushPins basierend) und PushPin (das über einen eigenen Ort verfügt). Wir erfuhren außerdem, wie eine Route optimiert (wofür mindestens vier WayPoints erforderlich sind) oder berechnet werden kann.

Nachdem wir Map Point erfolgreich absolviert hatten, beschäftigten wir uns mit Visio. Visio ist eine großartige Möglichkeit, effektiv Workflow-Diagramme zu generieren. Der Prozess, einen Automation Server zu erstellen, ist der gleiche wie vorher mit der Ausnahme, dass Visio von Anfang an sichtbar ist. Das Erstellen eines neuen Dokuments in Visio unterscheidet sich aber und der Entwickler muss `oVisio.Documents.Add()` verwenden. Marcia sagte, dass es so funktioniert.

Marcia erzählte uns, dass Visio nicht das erste Programm ist, das automatisiert werden sollte. Sie zeigte auf, dass die Arbeit mit diesem Objektmodell sehr schwierig sein kann.

Am Ende gab uns Marcia eine kurze Einführung in Outlook und wie es möglich ist, eine E-Mail mit Hilfe der Automation zu versenden. Dabei betonte sie, dass die Automation Server langsam arbeiten.

Die Session war wie die erste Session sehr interaktiv. Sie war voll mit eindrucksvollen Beispielen, die zeigen, wie es wirklich funktioniert, und natürlich auch mit Marcias Humor und Enthusiasmus.

Debugging Essentials – Rick Schummer



Jeder Programmierer auf dieser Welt versucht, eine perfekte Anwendung zu entwickeln. Die Anwendungen sind in der Regel sehr kompliziert und meist treten Fehler auf, die nicht erwartet waren und im besseren Fall beseitigen wir diese Fehler, bevor wir die Runtime erstellen. Im schlechteren Fall finden wir diese Fehler in der erstellten Runtime. Das effektive Debugging war Ricks Thema. Rick versuchte, uns zu erklären, dass das Debugging und das Testen hilfreich und wichtig sind. Je mehr wir debuggen, desto besser werden wir darin.

Wir begannen mit FoxPro für DOS, da dies hilft, das Erbe dieses Werkzeugs zu verstehen und weil es immer noch von vielen Entwicklern auf der gesamten Welt verwendet wird.

Ein weiteres Thema dieser Session war, uns einige Grundlagen über den Debugger nahe zu bringen. Rick begann mit Befehlen, die wir im Debugger verwenden können, beispielsweise:

- CLEAR DEBUG
- DEBUG OUT
- CLOSER DEBUGER

Rick vergaß nicht, den Dialog Optionen im Menü Extras zu erwähnen, wo Sie in der Lage sind, Ihren Vordergrund, Ihren Hintergrund sowie die Schriften zu ändern. Die Rückkehr zu den Ursprungswerten stellt ebenfalls kein Problem dar mit Ausnahme der Fonts, die Sie nicht zurücksetzen können. Sie können Code in einem Trace-Fenster markieren und ihn anschließend einfach in das Befehls-Fenster ziehen.

Versteckte Tipps waren ein anderes Thema, über das Rick sprach. Einige dieser Tipps nutzen Sie beim Einsatz des Trace-Fensters, wo Sie einige Probleme in Ihrem laufenden Code entdecken können, aber Rick führte auch aus, dass Sie keine Informationen erhalten, wenn Sie sie nicht finden. Er erwähnte außerdem das Watch-Fenster sowie das Locals-Fenster, die Ihnen alle Speichervariablen zeigen, die sich im Gültigkeitsbereich des aktuell ausgeführten Codeteils befinden.

Ein anderes wichtiges Feature des Debuggers sind die Haltepunkte. Haltepunkte dienen dem Anhalten eines laufenden Codes auf einer bestimmten Zeile und Rick zeigte uns, wie dieses mächtige Werkzeug effektiv genutzt werden können. Aber die Haltepunkte sind nicht alleine. Der Coverage Profiler ist ein anderes hoch performantes Werkzeug, das Ihnen wichtige Daten über die Geschwindigkeit Ihrer Anwendung liefern kann.

Diese Session war für diejenigen das Richtige, die den Debugger nicht nutzen und etwas darüber erfahren wollten. Sie war vollgepackt mit Tipps, Tricks und Techniken, die für das Entwickeln von fehlerfreien Anwendungen hilfreich sind.

Best Practices with Data – Andy Kramek



Was ist das Wertvollste für uns? Für manchen ist es Geld, für FoxPro-Programmierer sind es Daten. Sichere und konsistente Daten sind die besten Daten. Andy Kramek kam nach Frankfurt, um uns mitzuteilen, ob die DBF nach wie vor eine sichere und effiziente Stelle für die Speicherung Ihrer Daten ist.

Ist es sinnvoll, eine DBF als primären Speicherort zu verwenden? Wenn Sie meinen, zu diesem Thema gäbe es nur wenige Argumente, dann liegen Sie vollkommen daneben. Andy nannte einige Argumente dafür, beispielsweise:

- Geschwindigkeit
- Integration in die GUI
- Flexibel und einfach zu warten.

Es gibt aber auch einige Nachteile:

- Es ist einfach, gute Designregeln zu brechen
- Es ist zu einfach für ein Produkt eines Drittherstellers, die Daten zu lesen
- Zu viele Operationen erfordern das exklusive Öffnen der Datei

Andy wies außerdem darauf hin, dass das DBF-Format nicht sicher ist. DBF ist ein Standard, den viele andere Programme öffnen lesen und in den sie auch schreiben können.

Wo sollten wir DBFs verwenden? Andy empfahl uns, DBF in lokalen Netzen zu verwenden, wo wir die Datenbank problemlos kontrollieren können.

Ein anderes Thema der Session waren Namenskonventionen. Hier fanden wir viele interessante Dinge heraus, beispielsweise wann es sinnvoll ist, für Tabellen und Felder Namen im Plural zu wählen, Feldnamen auf zehn Zeichen zu beschränken und dass Abkürzungen entweder konsistent oder gar nicht verwendet werden sollten. Ein weiterer Tipp war, Suffixe zu verwenden, beispielsweise `_det` (für Detailtabelle) oder `_typ` (für Typendefinition).

Irgendwann ist die DBF nicht mehr ausreichend und Sie benötigen eine Datenbank (DBC). Andy legte uns sehr kurz die Vorteile des DBC dar, beispielsweise die Verwendung langer Tabellennamen oder die Möglichkeit, lokale Ansichten zu verwenden, was sehr hilfreich ist.

Während des Rests der Session redeten wir über andere Themen, beispielsweise über das Aktualisieren von Tabellen mit SQL Update, Locate oder Seek. Andy vergaß aus nicht zu erwähnen, wann wir Indizes verwenden sollten und wann nicht.

Zum Schluss zeigte Andy uns einige Beispiele, anhand derer er demonstrierte, wie schnell ein Update mit und ohne Index sein kann.

Advanced SQL Queries with VFP 9.0 – Andy Kramek

Jeder von uns hat irgendwann die SQL-Sprache verwendet, aber niemand von uns kann sich die wahre Stärke dieser Sprache vorstellen. In dieser Session lernen Sie einige erweiterte Techniken kennen, an denen SQL beteiligt ist und die wir in Visual FoxPro 9.0 einsetzen können.

Andy begann mit der Arbeit mit Cursors. Cursor sind das Ergebnis von SQL-Abfragen. Andy führte aus, dass unser Cursor, der als Ergebnis einer SQL-Abfrage erstellt wurde, schreibgeschützt ist, zeigte uns aber im Anschluss die Lösung, die Verwendung des Befehls USE AGAINST. Außerdem gibt es noch alternative Ausgabemöglichkeiten, beispielsweise in eine Datei, auf den Bildschirm oder auf den Drucker.

Anschließend wandten wir uns den Joins zu. Andy bezeichnete den Inner Join als „natürlich“. Dieser Join ist auch der Standard-Jointyp. Anschließend sprach er über den Outer Left Join, den Outer Right Join und den Full Outer Join, der eine Kombination der beiden davor genannten darstellt. Der Full Outer Join besteht aus drei Untermengen der Daten.

Eine andere hilfreiche Sache, die Andy demonstrierte, war UNION. Dieser Befehl kann zwei Ergebnismengen aus zwei Abfragen zusammenfassen. Wir kennen zwei Arten von UNION:

- UNION ALL
- UNION (entfernt doppelte Datensätze)

Das nächste Thema in Andys Session waren die Unterabfragen. Er erklärte, dass eine Unterabfrage eine Abfrage ist, die in eine Abfrage eingebettet ist und die dazu dient, einen oder mehrere Werte zu erhalten. Es gibt vier Unterabfragetypen:

- Nicht korrelierende Unterabfragen (in der Klausel WHERE definiert)
- Korrelierende Unterabfragen (in der Klausel WHERE definiert)
- Abgeleitete Tabelle (in der Klausel FROM definiert)
- Berechnetes Feld (in der Klausel SELECT definiert)

Am Ende der Session zeigte Andy uns einige Beispiele für Abfragen und Unterabfragen. Für diejenigen, die immer noch nicht überzeugt waren, hatte Andy einige spezielle Beispiele vorbereitet. Hier ein Beispiel zur Illustration:

- Die Aufgabe besteht darin, eine Abfrage zu erstellen, die: Alle Produkte auflistet, die ein Kunde jemals bestellt hat, die aber NICHT in seiner letzten Bestellung enthalten sind. Diese Liste soll auch das Datum enthalten, an dem jedes Produkt zuletzt bestellt wurde.
- Felder: Kundename, Produktname, Datum
- Tabellen: Customer, OrderHdr, OrderDet, ProdMain
- Erforderliche Informationen: Das letzte Bestelldatum, das vor der letzten Bestellung liegt, für alle Produkte, die jemals bestellt worden sind.

Freitag, 10. November

Von Jan Vit und Jan Král

Windows Communication Foundation – Craig Berntson



Damit die Teilnehmer das Thema dieser Session und auch die Technologie dahinter schätzen lernten, begann Craig mit einer kleinen Geschichte über einen Menschen namens Bob. Er war sehr intelligent und eines Tages erfand er ein nettes kleines Programm. Wie es häufig der Fall ist, wuchs das Unternehmen und die neuen Versionen des Programms mussten immer mehr kommunizieren. Teils mit anderen Anwendungen auf dem gleichen Rechner oder im lokalen Netzwerk, teils mit einer vollständig anderen Anwendung in einem anderen Netzwerk hinter Firewalls. Als Bob mehr und mehr

Telefonanrufe mit der Forderung nach weiteren Features erhielt, erweiterte er seine Software, so dass sie wie gewünscht funktionierte und ging zu einer Party, seiner Lieblingsbeschäftigung. Als er später die Zeit fand, sich all die Änderungen anzusehen, die er an der originalen Software vorgenommen hatte, stellten wir fest, dass er etwas neues erstellt hatte, das heute Windows Communication Foundation genannt wird. Armer Bob, jetzt können wir ständig feiern.

Nun, nach 20 Minuten, begann Craig mit seinem Überblick.

Windows Communication Foundation, kurz WCF, ist eine neue Technologie, die im .NET Framework 3.0 zur Verfügung steht, das früher in dieser Woche veröffentlicht wurde. Sie wurde auf der Basis der Erfahrungen mit den aktuellen Kommunikationsprotokollen und Techniken erstellt, auch auf der Grundlage von MSMQ, Webdiensten und COM+. Ziel der neuen Technologie ist, die aktuellen Technologien vollständig abzulösen.

Die Welt der Kommunikation hat sich vom Aufruf von Methoden (strukturierte Programmierung – 70er Jahre) über den Aufruf von Objekten (Objektorientierung – 80er Jahre) zum Aufruf von Komponenten (90er Jahre) entwickelt. Jetzt machen wir den nächsten Schritt. Wir senden Nachrichten (Serviceorientierung). Es wäre ein Fehler, zu meinen, dass damit die Objektorientierung überflüssig geworden sei. Der neue Schritt in der Entwicklung bedeutet lediglich, dass die Kommunikation wie wir sie kennen sich geändert hat und dass die Objekte sich direkt aufrufen.

Die Webdienste können als Beginn der serviceorientierten Techniken betrachtet werden. Sie sind aber nach wie vor nicht ideal und können definitiv nicht alle Probleme lösen.

Der Einsatz der WCF ist geradlinig und entspricht in einigen Dingen dem, was wir von den Webdiensten kennen. Sie benötigen eine Oberfläche, den sogenannten Servicekontrakt, die die Operationen und Methoden definieren, die zur Verfügung stehen und selbstverständlich den aktuellen Dienst, der die Oberfläche implementiert.

Die WCF ist sehr gut skalierbar und trotzdem einfach zu verstehen und einzusetzen. Craig führt aus, dass, so weit er informiert ist, es von Microsoft ein offizielles Versprechen gibt, dass diese Technologie in der Zukunft aus Visual FoxPro heraus verwendet werden kann. Bis dahin bleibt die Technologie außen vor und steht lediglich für Experimente mit Hilfe von Visual Studio zur Verfügung.

Windows Presentation Foundation – Craig Berntson

Diese Session war ein Startpunkt in eine neue und überraschende Technologie namens Windows Presentation Foundation, kurz WPF. Auch wenn WPF erst in dieser Woche veröffentlicht wurde und daher wirklich neu ist hat Craig viele Beispiele mitgebracht, die mit früheren CTPs erstellt waren.

Wird über WPF geredet, geht es meist um XAML. Diese neue Sprache ähnelt XML und ermöglicht es den Entwicklern, sämtliche Präsentationsschichten zu entwerfen und sie in Windows- oder Webumgebungen gemeinsam zu nutzen. Es hat den Anschein, als würden mit der Einführung dieser Technologie die Windows- und Webformulare veraltet seien.

Wenn Sie mögen, was Sie bis hierhin kennen gelernt haben, wollen Sie es eventuell selbst ausprobieren. Alles was Sie dafür benötigen ist das Visual Studio 2005, das .NET Framework 3.0 und das Windows Vista SDK. Dies alles erhalten Sie auf einer speziellen Website, die sich mit den neuen Technologien rund um das .NET Framework 3.0 befasst. Es handelt sich um www.netfx3.com.

Ein weiteres hilfreiches Werkzeug ist Cider. Cider erstellt Ihren XAML-Code mit Hilfe des klassischen Drag-and-Drop, wie Sie es von Visual Studio kennen. Auch dieses Werkzeug erhalten Sie auf der bereits erwähnten Website.

Unternehmen, die über ein großes Entwicklungsteam verfügen, oder die Teile der Softwareentwicklung an andere Unternehmen vergeben, wird es gefallen, dass der XAML-Code vollständig vom Programmcode getrennt ist, so dass das Oberflächendesign der Anwendung durch professionelle Designer erledigt werden kann, nicht durch Programmierer. Im Gegensatz zu Programmierern mögen es die Designer, ihre Zeit mit Oberflächenentwürfen zu verbringen. Ob Sie es glauben oder nicht – durch das Aussehen Ihrer Benutzeroberfläche verkaufen Sie Ihre Anwendung als Erstes.

Da die Session auf einer Visual FoxPro-Konferenz gehalten wurde, zeigte Craig auch Möglichkeiten auf, dieses coole neue Feature von Visual FoxPro aus zu verwenden. Grundlage dafür ist ein ActiveX-Steuerelement, das auf einem VFP-Formular platziert wird.

Mit Hilfe von Visual FoxPro zeigte Craig verschiedene graphisch ansprechende Beispiele, beispielsweise eine Uhr, die wie ein Gadget von Vista aussieht oder ein hilfreiches Werkzeug zum Zoomen. Jeder war überrascht, dass das letzte Beispiel eine einfache Listbox verwendete, die mit WPF erstellt war. Das war nur zur Demonstration gedacht: „Hey, Sie können nicht nur die schönen Dinge machen, auch ganz normale Aufgaben lassen sich mit XAML erledigen.“

Application Architecture for Multiple UIs – Beth Massi



Diese Session zielte vor allem auf den Einsatz mobiler Geräte, auf das Erstellen von UIs für die Geräte sowie auf die Verwaltung des Datenzugriffs dieser Geräte.

Das Problem mit den mobilen Geräten besteht darin, dass sie in der Regel mit ihren Eigentümern auf die Reise gehen und dass dabei ihre Konnektivität häufig wechselt. Selbstverständlich könnten wir es vermeiden, eine spezielle Software für die mobilen Geräte zu erstellen, indem wir alles, was durch unsere mobilen Anwender benötigt wird, auf eine Website stellen. Diese

Website kann problemlos mit jedem Gerät angesehen werden. Leider reicht dieses Vorgehen nicht in jedem Fall aus. Manchmal ist es erforderlich, die Daten auch offline zur Verfügung zu haben und diese mit jeder sinnvollen Verbindung synchronisieren zu können. Um dieses Problem dreht sich diese Session.

Damit das Gerät in der Lage ist, unsere Software auszuführen, ist das Compact Framework erforderlich. Es handelt sich dabei um eine kleinere Version des klassischen .NET Framework, etwa 28 % davon. Es wurde so verkleinert, dass es weniger Speicher benötigt, ohne dass zu viele Features herausgenommen wurden.

Zunächst müssen wir entscheiden, welchem Teil der Anwendung den Zugriff zur Verfügung stellen wollen. Die vermutlich einfachste Möglichkeit besteht darin, auf der Basis eines bestehenden Geschäftsobjekts in der mittleren Schicht eine Fassade zu errichten. Dadurch erreichen wir eine klare Trennung der Geschäftsseite der Anwendung von der Präsentationsseite. Außerdem sind wir dadurch vor allen möglichen Änderungen geschützt, die Microsoft in seiner Kommunikationssoftware (beispielsweise WCF) vornehmen könnte. Die Übertragung unseres Systems auf einen anderen Kommunikationsstandard sollte ebenfalls durch eine geringfügige Änderung innerhalb der Fassade möglich sein.

Nach der Entscheidung, welcher Kommunikationsstandard verwendet werden soll, müssen wir unser Nachrichtenformat definieren, den Kontrakt. Die meisten Entwickler werden für kleine Datenmengen XML wählen oder ein Binärformat, beispielsweise für große Bilder. Die einfachste Möglichkeit besteht aber vermutlich darin, ein serielles Datenset zu versenden, mit dem wir auf vielen unterschiedlichen Plattformen arbeiten können, auch in Visual FoxPro.

Damit das „gelegentlich verbundene Szenario“ funktioniert, benötigen wir auch irgendeine Software, die die Daten auf der Clientmaschine hält, auch wenn es sich um ein kleines Mobiltelefon handelt. Die Lösung dieses Problems ist eine vereinfachte Version des SQL Server, SQL mobile.

In ihrer Präsentation zeigte Beth nicht nur viele Beispiele mit Hilfe des Geräteemulators von Visual Studio 2005, sondern sie hatte auch ein Smartphone dabei, mit dem jeder Teilnehmer nach der Session herumspielen konnte. Es war einfach, ihrer Präsentation zu folgen, die mit aktuellen Informationen aus der Welt der mobilen Geräte angefüllt war.

Installing Applications Using Inno Setup – Doug Hennig



Das Entwickeln von Software ist eine Sache, sie auszuliefern eine andere. Wir haben alle bereits eine Anwendung entwickelt und wollten sie eventuell mit dem Windows Installer ausliefern. Was wäre, wenn ein anderer Installer existieren würde? Ein Installer, der effizienter, schneller und einfacher zu bedienen ist? Doug versuchte, uns eine andere Lösung zu zeigen, wie wir unsere Software ausliefern, ohne dafür den Windows Installer zu verwenden.

Doug stellte Inno Setup vor. Dieser Setup-Assistent enthält viele hervorragende Features. Eines davon ist, dass Inno erheblich kleinere Dateien als der Windows Installer generiert. Sowohl Installation als auch Deinstallation sind erheblich schneller als mit dem Windows Installer. Außerdem ist er absolut kostenlos.

Nach einer kurzen Vorstellung von Inno Setup teilte Doug uns mit, was wir mit dem Inno Setup Wizard alles anstellen können. Wenn wir wollen, können wir unseren Kunden im Setup Wizard mehrere Sprachen anbieten (Englisch ist die Standardsprache) oder wir können die Installation durch ein Passwort schützen. Das Erstellen eines Symbols auf dem Desktop ist absolut kein Problem. Lizenzen sind ein weiteres Feature, das es uns ermöglicht, einen eigenen Lizenztext in die Installation aufzunehmen. Wenn Sie dies wünschen, können Sie mit Ihrer Anwendung einen Uninstaller installieren, der Ihr Programm sicher deinstalliert.

Die Definition des Verzeichnisses, in das die Anwendung installiert werden soll, ist eine weitere Einstellung, die Sie problemlos vornehmen können. Sie können den Anwender entscheiden lassen, wohin die Anwendung installiert wird oder Sie können festlegen, welches Verzeichnis verwendet wird.

Doug zeigte auf, dass Inno Setup ohne größere Probleme durch Skripte gesteuert werden kann. Wenn Sie über die notwendige Erfahrung verfügen, können Sie Inno Setup an Ihre eigenen Vorlieben anpassen. Inno Setup-Skripte sind einfache Textdateien, die weitgehend den .INI-Dateien entsprechen.

In dieser Session lernten wir eine großartige Lösung für die Installation und den Vertrieb unserer Software kennen. Sie ist einfach, schnell und kostenlos und Doug vergaß auch nicht, uns einen Link zu zeigen, von dem wir Inno herunterladen können, zusammen mit anderen hilfreichen Ressourcen.

Metadata to the Maximum – Andy Kramek

Ein Albtraum eines Entwicklers ist, wenn er herausfindet, dass er etwas in seiner Anwendung ändern muss. Das Problem: Können Sie sich vorstellen, Ihre Anwendung zu ändern, ohne dabei den Code zu ändern? Können Sie sich vorstellen, Ihre Anwendung im laufenden Betrieb zu ändern? Andy Kramek ist der lebende Beweis, dass dies möglich ist, auch wenn dafür eine trickreiche Vorgehensweise für die Änderungen an der Anwendung gewählt werden muss.

Zunächst definiert Andy drei Datentypen:

- Rohdaten – diese Daten müssen mit Ihrer Anwendung geliefert werden
- Verarbeitungsdaten – Informationen, die Sie nicht aus Ihren Rohdaten ziehen können
- Metadaten

Metadaten sind Daten, die das System unterstützen. Sie sind kein Teil einer Anwendung. Andy betonte diesen Punkt besonders. Er führte aus, dass Metadaten Daten über die Daten sind. Sie können in drei Gruppen aufgeteilt werden:

- In der Konfiguration werden die Voreinstellungen des Anwenders gespeichert, beispielsweise die Einstellungen für die Anwendung. Die Konfiguration verwendet dafür eine .INI-Datei oder die Registry von Windows. Dieser Metadatatyp kann zur Laufzeit geändert werden.
- Die Definition speichert Daten, die zur Laufzeit benötigt werden. Diese Metadaten werden zur Laufzeit nur selten geändert.
- Verarbeitungsdaten bestehen aus einem Codeblock, der „on the Fly“ ausgeführt wird.

Dann bleibt nur noch eine Frage. Sollen wir uns mit Metadaten belasten? Andy versuchte uns zu überzeugen, dass die Metadaten dies wert sind. Mit Hilfe von Metadaten können Sie Ihre Anwendung zur Laufzeit ändern. Sie ändern nicht den Code, sondern die Daten. Das ist das wichtigste Feature der Metadaten.

Das Problem besteht darin, dass Sie eine Anwendung herunterfahren müssen, damit Sie eine Änderung am Code dieser Anwendung vornehmen können, da der Code der Anwendung zur Laufzeit schreibgeschützt ist. Eine typische Übung für die Verwendung von Metadaten in der realen Welt ist der Einsatz in einem Krankenhaus. Dort können die Anwendungen nicht einfach beendet werden.

Auf der einen Seite sind die Metadaten ein großartiges Werkzeug für die Änderung der Verhaltens Ihrer Anwendung zur Laufzeit, auf der anderen Seite erfordert dieses Vorgehen ein anspruchsvolleres Codedesign. Außerdem ist es schwieriger wartbar und zu dokumentieren.

Die Session war wie üblich voll mit Beispielen, die sowohl den konkreten Prozess erklärten als auch andere hilfreiche Tipps und Tricks enthielten.

Datenanbindung – Michael Niethammer



Bei uns dreht sich alles um Daten. Wir überlegen, wie wir sie speichern, wie wir sie ändern und wie wir sie möglicherweise löschen. Michael weiß das uns bereitete daher diese interaktive Session vor, die sich auf die Arbeit mit Daten in Visual FoxPro-Formularen konzentrierte. Diese Session basierte weitgehend auf einfachen Beispielen, die die Grundlagen dieses Themas demonstrierten.

Zunächst erstellte Michael mit dem Befehl `CREATE PROJECT` ein Projekt und in diesem Projekt erstellte er ein Formular. Anschließend gab er uns einige Informationen über die Datenumgebung, in die wir später Tabellen einfügen. Der Sprecher vergaß auch nicht, einige wichtige Ereignisse zu erwähnen, beispielsweise `BeforeOpenTables` und `AfterCloseTables`. Für die Liebhaber von Methoden erwähnte Michael `OpenTables()` und `CloseTables()`.

Wir fahren mit dem `CursorAdapter` fort. Wir begannen mit dem `CursorAdapter-Generator` und Michael erklärte, wie er funktioniert. Michael zeigte auf, dass wir, wenn wir unsere Daten im `CursorAdapter` aktuell halten wollen, im Generator das Feld `AutoUpdate` anklicken müssen.

Wir verlassen den `CursorAdapter` und die Datenumgebung und beginnen mit den an die Daten gebundenen Steuerelementen. Diese Steuerelemente verwenden die Eigenschaft `ControlSource` und arbeiten mit Tabellen, die sich in unserer bereits gefüllten Datenumgebung befinden. Zu diesen Steuerelementen gehören:

- `TextBox`
- `ListBox`
- `PageFrame`
- `Grid` usw.

Am Ende der Session redeten wir über die Tabellenpufferung (pessimistisch oder optimistisch), sowie über die Befehle `REPLACE`, `DELETE` und `APPEND`. Die Datensitzung ist ein anderes Thema und Michael betonte, dass die Datensitzung nichts mit der Datenumgebung gemeinsam hat.

Die Session war für Anfänger gedacht, die versuchen, mit der Datenumgebung und den datengebundenen Steuerelementen zu arbeiten. Dank der großartigen Beispiele war es kein Problem, zu verstehen, die die Datenumgebung, der `CursorAdapter` und anderes funktionieren.

Visual FoxPro Tools und Assistenten – Michael Niethammer

Den Tools von Visual FoxPro war eine andere Session auf der heutigen Liste gewidmet. Michael wollte und in die Mystik der Visual FoxPro-Tools einführen und er begann mit dem Projekt-Manager.

Zunächst erstellten wir ein neues Projekt und Michael beschrieb alle Teile dieses Tools. Er führte aus, dass, wenn Sie ein verschlüsseltes Projekt haben wollen, Sie nicht vergessen dürfen, dies im Projekt-Manager anzuklicken. Der Projekt-Manager gibt Ihnen auch weitere Möglichkeiten, beispielsweise Ihrer Anwendung ein Icon zuzuordnen oder das Aufnehmen von Dateien in Ihr Projekt, angefangen mit Programmen, Formularen und Klassen bis hin zu Datenbanken oder einzelnen Tabellen.

Anschließend gingen wir über zum Erstellen von Menüs mit dem Menü-Designer. Aus Gründen der Einfachheit erstellten wir ein Standardmenü und in diesem Modell zeigte uns Michael jede mögliche Einstellung, beispielsweise die Arbeit mit der Ergebnisbox, der Optionsbox usw. Er führte aus, dass die Arbeit in einem Menü nicht objektorientiert ist und er teilte uns mit, dass die Menüdateien die Endung .MPR aufweisen.

Michael enttäuschte uns nicht und sprach über unseren Lieblingsdebugger, der es uns ermöglicht, Bugs in unserer Anwendung zu lokalisieren und zu beseitigen, Variablen zu beobachten, Haltepunkte zu setzen und vieles mehr. Er betonte, dass die Fehlerbehandlung für Entwickler sehr wichtig ist, ich würde sogar meinen notwendig. Die Haupt-Fehlerbehandlung wird durch den Befehl ON ERROR, durch das Ereignis Error oder durch TRY...CATCH...FINALLY erledigt.

Ein weiteres zweifellos hilfreiches Werkzeug ist der Klassen-Katalog, der sich für die Steuerung und Administration von Visual FoxPro-Klassen eignet. Auch der Objekt Browser ist ein gutes Tool. Meist wird er für die Untersuchung externer Klassenbibliotheken eingesetzt, beispielsweise von Word oder Outlook.

Die Session war mit vielen anderen Tools wie dem Coverage Profiler und dem Document Wizard angefüllt, aber die Behandlung all dieser Werkzeuge würde mindestens zwei Stunden in Anspruch nehmen. Diese Session gab uns den erforderlichen Überblick über die Tools, die in Visual FoxPro verwendet werden.

Filter, Indizes, Rushmore & Co – Jürgen Wondzinski



Jürgen Wondzinski, weltweit als „wOOdy“ (wie von den Passwörtern bekannt, müssen Sie auch hier auf die Groß- und Kleinschreibung achten), präsentierte den Zuhörern seine Session über „das Verständnis von Visual FoxPro durch das Wissen, wo es herkommt).

wOOdy erfreute seine Zuhörer mit einer gut durchdachten Geschichte der unterschiedlichen Elemente von Visual FoxPro, aber auch der lockere Stil seiner Vorführung war unterhaltsam. So bezeichnete er sich selbst als FOXIL (sprechen Sie dies laut und

langsam aus), der seit den späten 80er Jahren mit der Domestizierung des „Fuchs“ beschäftigt ist. Wikipedia definiert den Begriff Domestizierung folgendermaßen: „Domestizierung oder Domestikation ist ein innerartlicher Veränderungsprozess von Wildtieren oder -pflanzen, bei dem diese über Generationen hinweg genetisch isoliert von der Wildform vom Menschen gehalten werden.“

Beginnend mit der Geschichte über das Erstellen von Vulcan erläuterte wOOdy, dass ursprünglich geplant war, ein System für das Beziehen von Daten zu entwickeln, das in den 70er Jahren in der Raumforschung im California Institute of Technology's Jet Propulsion Laboratory verwendet werden sollte. Dies bildete so etwas wie ein roter Faden, den wOOdy an unterschiedlichen Stellen verwendete, um das spezielle Verhalten von Visual FoxPro zu erläutern.

Anschließend erklärte wOOdy die grundlegenden Unterschiede zwischen xBase und SQL. Das Datensammlensystem von JPL wurde zu dBase II (die Geschichte können Sie unter http://www.foxprohistory.org/people_began.htm nachlesen), das sich auf dem aufstrebenden PC-Markt gut verkaufen ließ und seine ursprüngliche Aufgabe aufgrund seiner Datensatzorientierung besser als alle anderen Produkte erfüllte: das Sammeln und Verwalten von Daten.

Die Wurzeln von SQL gehen auf den Aufsatz „A Relational Model of Data for Large Shared Data Banks“ zurück, der in den 70er Jahren von Dr. Edgar F. Codd publiziert wurde, der eventuell die Blaupause für die IBM Datenbankentwicklung wurde. Da IBM die Großunternehmen bediente, konzentrierten sich IBM und andere auf große skalierbare RDBMS-Systeme und SQL wurde die Sprache dieser RDBMS, die auf Großrechnern ausgeführt wurden. Dies ist auch der Grund, weshalb SQL größere oder kleinere Datenblöcke auf einmal bewegt. Die bekannte Konsequenz ist, dass Sie sich, um SQL vollständig zu verstehen, die Theorie des Datensets studieren müssen. Oder wie es wOOdy anschaulich darstellte: Im schlimmsten Fall können Sie mit einem xBase-Befehl eine Tabelle löschen, mit einem falsch auf eine SQL-Datenbank verwendeten SQL-Befehl können Sie die gesamte Datenbank löschen.

Außerdem funktionieren xBase-Befehle immer auf einzelnen Arbeitsbereichen, SQL-Befehle, die auf eine SQL-Datenbank angewendet werden, führen immer zu einem Datenset. Aus dem Blickwinkel der Historie sind xBase-Befehle Datensatz- und Tabellenorientiert, SQL-Befehle sind Sektororientiert.

wOOdy erläuterte dies anhand der Entwicklung der Anzahl der Arbeitsbereiche von xBase / Visual FoxPro. Zunächst gab es gerade zwei Arbeitsbereiche. Als die PCs größer wurden und über mehr Speicher sowie mehr Festplattenplatz verfügten, bestand der nächste Schritt aus zehn Arbeitsbereichen, A bis J genannt. Dies ist auch der Grund, weshalb es bis heute nicht ratsam ist, Tabellen mit einem einzelnen Buchstaben zu benennen. Im nächsten Schritt wurde das Limit auf stolze 225 Arbeitsbereiche erweitert. Heute sind wir bei 32.767 Arbeitsbereichen.

Immer noch gibt es lediglich eine Tabelle je Arbeitsbereich und alle Befehle von Visual FoxPro (auch die VFP-internen SQL-Befehle) funktionieren nur im aktuell gewählten Arbeitsbereich.

wOOdy wandte sich den „guten“ und „schlechten“ Befehlen zu. Das Kriterium, das diese beiden Gruppen unterscheidet, ist der implizite Wirkungsbereich dieser Befehle. Wird dieser Wirkungsbereich nicht sorgfältig beachtet, kann das Ergebnis überraschend sein.

Am Schluss erklärte wOOdy das Geheimnis der Geschwindigkeit des Datenzugriffs, die bei Visual FoxPro nach wie vor ungeschlagen ist: die Rushmore-Technologie. Durch den intelligenten Einsatz einfache Bit-Schemata, die im Speicher gehalten werden, wurde dieses Wunder Wirklichkeit.

Keynote – Yair Alan Griver



Alan Griver, zu dessen Aufgaben bei Microsoft auch gehört, dass er Group Manager für Visual FoxPro ist, begann seine Keynote mit einem Beitrag von Kevin Regsdale, der einen Videoclip erstellt hat, der jedem begeisterten Visual FoxPro-Programmierer und Visual FoxPro-Fan aus dem Herzen sprach. Visual FoxPro ist nicht nur am Leben, sondern Visual FoxPro wächst und gedeiht und ist die erste Programmierumgebung, an die Sie denken sollten. Das Video zeigt VFP in einer Weise, die jede professionelle Produktion über ein Konsumerprodukt neidisch werden lässt.

Sie sollten sich Kevins VFP-Video einmal ansehen. Sie finden es unter der URL <http://weblogs.foxite.com/kevinregsdale/archive/2006/07/02/1939.aspx>. Außerdem haben Sie dort die Möglichkeit, bei der Promotion von Visual FoxPro zu helfen. Besuchen Sie die Site und lesen Sie!

Außerdem haben wir einen neuen Slogan für Visual FoxPro: Neben „Fox Rocks!“ haben wir jetzt auch „**VFP Connects!**“ Anschließend zeigte Yag, was Visual FoxPro verbindet und wie dies geschieht: „**VFP Connects! Sie und Ihre Daten!**“

Der Berichts-Generator wurde in Visual FoxPro 9.0 komplett überarbeitet. Aber auch das ist nicht die letzte Stufe der Evolution dieses vitalen Elements von Visual FoxPro.

Erweiterungen des Berichtssystems sind in Vorbereitung. Die Laufzeit ist sowohl Decorator als auch Visitor. Sie können einen Bericht vorverarbeiten, um Teile zu entfernen, die nicht angezeigt werden sollen - beispielsweise können Sie einen Bericht während des Ladens auswechseln, Links für HTML erstellen, aber verhindern, dass der Inhalt gedruckt wird (wenn diese Kombination angeraten ist), die Eigenschaften des Berichts können dynamisch zur Laufzeit behandelt werden, Berichtsobjekte können rotiert werden und einige ähnliche Dinge werden helfen, „den Anwender mit seinen Daten zu verbinden“.

VFP Connects! Sie mit dem SQL Server!

Visual FoxPro verfügt über sein eigenes robustes Datenbanksystem, aber SQL Server sind verbreitet und erledigen ihre Arbeit. Viele Menschen mögen sie und vermutlich wird es beim SQL Server bleiben. Das Visual FoxPro-Team hat also eine weise Entscheidung getroffen, indem sie eine der Grundregeln des Erfolgs übernommen hat: „Wenn du sie (diese SQLs) nicht bekämpfen kannst, schließe dich mit ihnen zusammen (das sollte hier eigentlich heißen: integriere sie). Aus diesem Grund hat der SQL Upsizing Wizard eine grundlegende Überarbeitung erfahren. Er ist jetzt ein großartiges Werkzeug, um die Datenwelt zu erobern. Yag zeigte eine Demo, die viele dieser überzeugenden Features verwendet. Dabei zeigte er auch, wie „visuell“ (=sichtbar) ein Visual Wizard sein kann.

Die einfache Bedienung ist eines der wichtigsten Features. So kann der SQL Upsizing Wizard mit den Standardeinstellungen gestartet werden. In einem erneuten Durchlauf können die Einstellungen optimiert und an die eigenen Bedürfnisse angepasst werden. Und für die Geschwindigkeits-Junkies: Die Performance beim Upsizing auf den SQL Server wurde verbessert. Doug Hennig hat eine Tabelle mit über 300.000 Datensätzen getestet. Früher benötigte das Upsizing dieser Tabelle zwei Stunden, jetzt noch elf Minuten.

VFP Connects! Uns mit allen anderen!

Yag ist im Rahmen seiner Anstellung bei Microsoft in zwei Bereichen aktiv: Er behält den erfolgreich den Überblick über die phänomenale Implementierung der Datenwerkzeuge in Visual Studio und er arbeitet zusätzlich mit den Entwickler-Communities.

Anschließend erscheint CodePlex auf der Bühne, Microsofts Website für Entwickler. CodePlex ist eine Community-Plattform, die es Entwicklern ermöglicht, neue Projekte zu Themen zu beginnen, an denen sie interessiert sind, um ihre Ideen und Fragen mit gleichgesinnten Entwicklern auf der gesamten Welt zu diskutieren. CodePlex ermöglicht es Entwicklern, andere Entwickler zu treffen, die ihre Projekte bereits in CodePlex gestartet haben und das zu verwenden, was sie von CodePlex-Projekten gelernt haben. Außerdem können die Entwickler hier ihr Feedback leisten.

Unter dieser Prämisse promotet Yag VFPX und VFPY, eine Art Visual FoxPro-Sektion innerhalb von CodePlex. Es handelt sich um eine Initiative der Visual FoxPro-Community, um Open Source Add-Ons für Visual FoxPro 9.0 zu erstellen. In seiner Keynote zeigte Yag einige Erweiterungen aus VFPX. Innerhalb von VFPX stehen der Community der Code, die Klassen und die Klassenbibliotheken zur Verfügung. Die folgenden Projekte werden aktuell weiterentwickelt:

- ClassBrowserX
- Code Analyst
- Ctl32 Statusbar
- GDIPlusX
- JustBehave
- Ersatz für die Dialoge Neue Eigenschaft und Neue Methode
- OOP Menu

Die Ergebnisse dieser Arbeit werden Microsofts Arbeit an der Verbesserung und Erweiterung von Visual FoxPro, Codename Sedna, vervollständigen.

VFP Connects! Sie mit Visual Studio und .NET!



Die Bibliothek besteht aus einer Collection von COM-Klassen, die eine Untermenge des .NET Framework 2.0 wrappen. Das .NET Framework ist eine umfangreiche Collection mit Namespaces und APIs, die eine umfassende Sammlung mit Funktionalitäten enthält, die Entwickler nutzen können, um Anwendungen zu erstellen, die auf der .NET Plattform laufen. Visual FoxPro enthält zwar auch eine umfangreiche Bibliothek mit APIs, aber einige Features sind in den Bibliotheken von Visual FoxPro entweder nicht vorhanden oder schwieriger zu verwenden als im Framework. NET4COM umfasst eine kleine

Untermenge des .NET Framework – eine Collection häufig verwendeter APIs, durch die Visual FoxPro auf Funktionalitäten zugreifen kann, die andernfalls nicht vorhanden wären.

Die Bibliothek MY für Visual FoxPro entspricht NET4COM. MY ist nativ für Visual FoxPro implementiert und stellt wie NET4COM hierarchisch häufig verwendete Funktionalitäten bereit, die einfach zu entdecken sind und in denen einfach navigiert werden kann. Das Sedna CTP Setup installiert die Bibliothek MY sowie die dazu gehörenden Dateien im Verzeichnis MY unterhalb des Zielverzeichnisses der Installation.

MY wurde mit Visual Basic DotNet eingeführt. Der Namespace MY ist eine native Klassenbibliothek von Visual FoxPro, die Wrapper für SYS()-Funktionen, die Windows API sowie für Eigenschaften und Methoden des Windows Script Host enthält. Er ermöglicht Anweisungen wie die folgende:

```
MY.Computer.FileSystem.SpecialFolder.MyDocuments
```

Außerdem liefert er Informationen über verschiedene Verzeichnisse wie das temporäre Verzeichnis, den Desktop und andere. MY gibt Verzeichnisse und Dateien als Objekte zurück und gibt Ihnen die Möglichkeit, diese Objekte zu ändern.

NET4COM und MY sind in der Microsoft Visual FoxPro Sedna CTP enthalten.

VFP Connects! Manchmal wirklich überraschend!

Und dann ist da noch Jeannine Johnson. Kennen Sie nicht? Nun, es gab den „Start Something Amazing Award“, ein Wettbewerb, bei dem die einzigartigen und kreativen Methoden ausgezeichnet wurden, mit denen Menschen Windows und andere Microsoft-Produkte verwenden. Jeannine Johnson aus Puyallup, Washington, USA, ist die Gewinnerin des ersten Preises im Bereich Sports & Games.

Trotz einer physischen Behinderung (Jeannine hatte bei der Geburt deutliches Untergewicht), die ihre Teilnahme an den traditionellen Sportarten verhinderte, hat sie mit dem Drag Racing ihre Nische gefunden, in der sie ihre Liebe zur Geschwindigkeit ausleben und Erfolge feiern kann.

Jeannine vertraut auf eine Windows-basierende Technologie, die ihr hilft, die Performance ihres Fahrzeugs zu überwachen und verwendet eine Technologie wie die Software Crew Chief Pro, um sicherzustellen, dass sich ihr Fahrzeug auf dem neuesten Stand befindet. Jeannine verbindet ihren Windows-PC mit dem Data Acquisition System ihres Autos, das die Drehzahl des Motors, die Temperatur und andere Parameter misst. Die Ergebnisse dieser Messungen werden durch das Softwarepaket ausgewertet.

Nebenbei bemerkt – Crew Chief Pro ist von Don Higgins in Visual FoxPro geschrieben worden. Diese sowie andere Software, die Don Higgins erstellt hat, werden in der nordamerikanischen Rennszene häufig eingesetzt.

Auch sonst verlässt sich Jeannine auf Windows-basierende Technologie. Neben anderen Dingen hat sie Microsoft Windows Movie Maker verwendet, um einen Film über ihre Rennen zu erstellen und sie verwendet Microsoft Office Word und Microsoft Office PowerPoint, um die Sponsoren ihres Rennsports zu verwalten.

„Ohne diese Windows-Technologien wären meine Möglichkeiten eingeschränkt, nicht nur in meiner Bildung, sondern auch im Hinblick auf mein Hobby und den Rennsport.“

VFP Connects! Uns mit anderen!

Im Web gibt es viele Sites, die die Mitglieder der Community verbinden. Hier einige Beispiele:

- VFP DevCenter
- MSDN-Foren
- Universal Thread
- Foxite
- Deutschsprachige FoxPro User Group
- Yags Blog

Samstag, 11. November

Von Jan Vit und Jan Král

Designing mit UML – Craig Berntson

Da diese Session als Einführung in UML gedacht war, begann Craig mit einer kurzen Einführung in die grundlegenden Diagrammtypen, beispielsweise in den Use Case, das Sequenzdiagramm, das Klassendiagramm, das Komponentendiagramm und das Paket. Für das Klassendiagramm verwendete er etwas mehr Zeit, da dies am häufigsten verwendet wird und er erklärte kurz, wie wir Elemente wie eine Schnittstelle, eine Klasse, die Vererbung, Container und ähnliches zeichnen.

Wir können UML in vielen Situationen verwenden. Das erste und offensichtliche Einsatzgebiet ist der Prozess des Designen einer neuen Anwendung. Vermutlich erstellen Sie zunächst einen Use Case, damit Ihre Anwender verstehen, was die Anwendung macht, wenn sie fertig gestellt ist. Ist der Use Case durch die zukünftigen Anwender abgesegnet, fahren Sie fort und erstellen die Klassen- und Sequenzdiagramme. Dieses Vorgehen ist selbstverständlich nicht das allein richtige. Sie können die Diagramme auch während der Codierung erstellen. Die letzten Diagramme, die Sie erstellen, sind das Komponenten- und das Paketdiagramm, die Ihnen einen Überblick über Ihre Anwendung geben.

Ein anderer Grund, UML einzusetzen, könnte sein, wenn Sie eine bestehende Anwendung überarbeiten und nur einen Teil davon ändern wollen oder wenn Sie mit einer vollständig unbekannt Anwendung arbeiten und sich zumindest einen Überblick verschaffen wollen, wie sie intern funktioniert.

Um UML erzeugen zu können, benötigen Sie ein Werkzeug. Auf dem Markt werden heute viele Tools angeboten, beispielsweise Rational Rose, Visual UML, Visio oder die Unterstützung von UML in Visual Studio 2005. Ein großartiges Werkzeug, das sich Craig kürzlich angesehen hat, ist Sparx Enterprise Architect. Diese Software hat er auch für seine Beispiele verwendet. Leider gibt es heute kein UML-Werkzeug für Visual FoxPro. Einige der bestehenden Werkzeuge können aber erweitert werden. Sie für VFP verfügbar zu machen erfordert nur wenig Arbeit.

Getting the most out of Reporting – Lisa Slater Nicholls



Ein Großteil dessen, was Lisa in ihrer Session zeigte, zielte nicht auf die neueste Version von VFP und könnte daher auch mit älteren Versionen verwendet werden. Lisa erklärte außerdem, dass diese Session einige Features von VFP bis an ihr Limit belastet und dass daher durchaus ein Crash vorkommen kann.

Sie erklärte, dass Report Listener nicht immer die beste Technik seien. Manchmal funktionieren die alten Methoden genauso gut oder haben eine noch bessere Performance, auch wenn sie schwieriger zu codieren sind.

Im Service Pack werden einige Probleme bei der Arbeit mit GDI+-Handles beseitigt und wir erhalten eine verbesserte Kontrolle darüber, wie viele Seiten und welche Teile der Seiten verarbeitet werden, indem einige neue Eigenschaften für Report Listener eingeführt werden:

- PageLimit
- PageTopLimit
- PageTailLimit
- PageLimitInsideRange

Die wichtigste Sache beim Erstellen eines Berichts ist, die richtigen Daten für den Bericht zu erhalten. Diese Daten können entweder in einer denormalisierten Form oder relational vorliegen. Die denormalisierte Form ist im Bericht sehr einfach zu verwenden, bewirkt aber, dass viele redundante Daten über das Netz gezogen werden müssen (wenn die Daten beispielsweise von einer SQL-Datenbank auf dem Server kommen). Da alle Daten heute von externen Quellen kommen können, ist es empfehlenswert, die relationalen Daten zu verwenden, auch wenn dies bedeutet, dass während der Programmierung mehr Arbeit erledigt werden muss.

Da die direkte Platzierung der Daten in den Bericht als schlecht angesehen wird, sollten einige spezielle Datenklassen erstellt werden. Anschließend sollten sie so erweitert werden, dass sie als Datenquellen für Berichte verwendet werden können.

Die Session enthielt noch viele weitere interessante Tipps und Tricks, beispielsweise das Generieren korrekt formatierten HTMLs, indem der Bericht zweimal ausgeführt wird oder das Beziehen der gesamten Seitenzahl des Berichts, um anschließend die Papiergröße anzupassen, so dass der gesamte Bericht eine Seite ausfüllt.

Best Practices for the Middle Tier – Craig Berntson



Da sich diese Session um das beste Vorgehen in der mittleren Schicht dreht, begann Craig mit einer kurzen Einführung, was derzeit eine mittlere Schicht ist und was ihre Aufgaben sind. Anschließend fragte er die Zuhörer, ob sie ihm bis zu diesem Punkt folgen konnten. Mit Ausnahme von Igor im Hintergrund, der vorgab, nichts verstanden zu haben, antwortete jeder so schnell und richtig, dass Craig mit seiner Präsentation begann.

Die wichtigste Sache der mittleren Schicht ist, dass sie sich in der Mitte befindet. Dadurch ist si-

chergestellt, dass die Benutzeroberfläche und die Daten im Backend niemals direkt miteinander zu tun haben. Die Unterteilung in Schichten kann logisch (nur durch die Software) vorgenommen werden oder die Schichten können jeweils auf einem eigenen Rechner laufen.

Auch wenn es früher in dieser Woche einen anderen Anschein hatte, empfiehlt Craig COM+ als beste Wahl für das Erstellen der mittleren Schicht. Diese Technologie bietet standardmäßig viele Vorteile, beispielsweise Sicherheitsfunktionalitäten, die Unterstützung von Transaktionen sowie eine einfache Installation und Administration.

Bevor Craig mit COM+ und seinen Beispielen begann, was die Grundlage seiner Session bildete, erwähnte er einige andere gute Vorgehensweisen bei der Erstellung der mittleren Schicht. Zunächst einmal ist es in der Regel das Beste, unsere Komponenten statuslos zu belassen, um die Menge des benötigten Speichers zu minimieren. Benötigt aber unsere Anwendung den Status, versuchen Sie, die Informationen irgendwo anders unterzubringen, beispielsweise im SQL Server. Um die Zeit der Verbindung zu optimieren, versuchen Sie, Methoden mit mehr Parametern zu verwenden, statt viele Eigenschaften in Ihrem COM+-Objekt zu setzen.

Eine andere wichtige Sache ist die Entscheidung, in welcher Form die Daten gesendet werden sollen. Die VFP-Cursor stehen außer Zweifel, da VFP mit ihnen umgehen kann und da sie sich im Arbeitsspeicher befinden. Eine benutzerdefinierte Objektlösung klingt erheblich besser, benötigt aber zu viel Overhead. Die beste Möglichkeit ist vermutlich der Einsatz von XML oder ADO Recordsets, am Ende in XML.

Bei der Implementierung der mittleren Schicht in Visual FoxPro achten Sie darauf, die Klasse Session als Basisklasse Ihrer Objekte verwenden, da diese Klasse über ihre eigene Datensitzung verfügt und für den Einsatz in der mittleren Schicht optimiert ist. Anschließend müssen Sie Ihre Objekte nur noch als OLE PUBLIC kennzeichnen und als Multi-threaded DLL erstellen, schon ist die mit VFP erstellte DLL fertig für den Einsatz in COM+.

In einem sehr deutlichen Beispiel zeigte Craig den gesamten Prozess des Erstellens einer DLL, die für COM+ geeignet ist und zeigte, wie sie von der Benutzeroberfläche aus angesprochen wird. Er sprach über viele weitere Themen, die mit diesem Themenkomplex zusammenhängen, beispielsweise über die Unterstützung von Transaktionen, die Auslieferung der COM+-Anwendung auf den Server und den Client und er erläuterte, wie sich der COM-Albtraum, das Debugging, etwas einfacher gestalten lässt.

Bis Windows Communication Foundation wirklich an den Start geht, ist COM+ DIE Möglichkeit für das Erstellen der mittleren Schicht, mit der Visual Basic, Office und .NET genau wie Nicht-Microsoft-Clients auf einfache Weise kommunizieren können.

Adding IntelliSense to Your Applications – Doug Hennig

IntelliSense ist definitiv das beste Feature, das Microsoft jemals hinzugefügt hat. Es verbessert unsere Produktivität und erleichtert deutlich die Anwendungsentwicklung. Doug wusste dies und entschied, uns etwas über das Hinzufügen von IntelliSense zu unseren eigenen Anwendungen zu erzählen. Dabei taucht eine Frage auf. Ist es erforderlich, einer Anwendung IntelliSense hinzuzufügen? Wird der Anwender sie nutzen?

Die IntelliSense wurde mit Visual FoxPro 7.0 eingeführt und jeder Entwickler mochte sie. In Visual FoxPro 9.0 haben wir jetzt die Möglichkeit, unsere eigene IntelliSense in ausgelieferten Anwendungen einzusetzen.

Doug begann damit, die IntelliSense für die Laufzeit einzurichten. Zunächst versuchte er, uns zu erläutern, wie die IntelliSense funktioniert. Dafür wird das in `_CODESENSE` angegebene Programm ausgeführt. Diese Systemvariable verweist auf `FoxCode.app` usw. In unserer Laufzeitumgebung funktioniert IntelliSense an zwei Stellen – in einem Codefenster und in einem Memofeld.

Probleme gibt es überall, so auch hier. Doug meinte damit die Anzeige aller Elemente einer Klasse mit Ausnahme des Elements, das er verwenden will.

Am Ende der Session sprach Doug über FFI, die Registrierung von Klassen mit der FFI oder den Einsatz des Favoriten als IntelliSense-Registrierungseditor, wo Sie einen Namespace ändern oder auch Methoden und Eigenschaften wählen können.

In dieser Session erhielten Sie eine vollständige Beschreibung, wie Sie IntelliSense in Ihre Anwendung integrieren können. Es war eine überwältigende Fülle von Tipps und Tricks, mit denen sich Ihre Anwendung anwenderfreundlicher gestalten lässt. Oder wie Doug sagte: „Die Anwender werden Sie dafür lieben.“

Extending the Data Explorer – Rick Schummer

Wäre es nicht schön, über ein Werkzeug zu verfügen, das uns die Komponenten in Visual FoxPro, in SQL Server-Datenbanken oder über eine ADO-Verbindung in anderen Datenbanken anzeigt? Für die Antwort auf diese Frage müssen wir nicht lange nachdenken. Es wäre schön. Rick Schummer zeigte uns die Grundlagen des Data Explorer und nicht nur die Grundlagen. Er zeigte uns auch, wie der Data Explorer erweitert werden kann.

Der Data Explorer entspricht weitgehend dem SQL Server Enterprise Manager und auch dem Query Analyzer. Er hat aber einen großen Vorteil. Der Data Explorer ist in der Lage, sehr schnell auf die Daten zuzugreifen und er funktioniert mit allen Daten. Im Grunde untersucht der Data Explorer Ihren Rechner und Ihr Netzwerk nach Komponenten und Datenbanken. Rick führte aus, dass, wenn der Entwickler über ein großes und langsames Netzwerk verfügt, der Data Explorer langsamer startet aus auf einem schnelleren Netzwerk.

Das vielleicht beste Feature des Data Explorer, das Rick vorstellte, ist seine Einfachheit. Wenn Sie einfach eine Tabellenstruktur des SQL Server sehen wollen, müssen Sie achtmal klicken, um sie anzuzeigen. Im Data Explorer sind drei Klicks ausreichend. Damit spart der Data Explorer Ihre Zeit, schont Ihre Maus und schützt Sie vor einem Tennisarm.

Mit dem Data Explorer können Sie sehr schnell Abfragen erstellen und das Beste daran ist, dass Sie Ihre eigenen Funktionen hinzufügen können. Nichts ist perfekt. Rick zeigte uns, wie eine Funktion mit Drag-and-Drop eingefügt wird. Er zog eine Tabelle vom Data Explorer und ließ sie auf ein Formular fallen. Jetzt würden Sie erwarten, dass ein Grid erscheint. Der Grid erscheint auch, aber ohne Daten. Er wiederholte den Prozess mit einem Feld der Tabelle und es funktionierte perfekt.

Nach der Tour durch den Data Explorer wandte sich Rick der Erweiterbarkeit des Data Explorers zu. Als Beispiel ermöglichen Ihnen die Funktionalität Document Database sowie der Add-In Manager, den Data Explorer zu erweitern.

Am Ende der Session erzählte uns Rick über die neuen Features in Sedna, fasste anschließend die Session zusammen und gab uns zum Schluss einen Rat: „Der Schlüssel ist, dass Sie überall mit der rechten Maustaste klicken können.“

The Power of Cursor Adapters – Venelina Jordanova



Stellen Sie sich vor, Ihr Kunde verwendet eine Anwendung, die mit einer Visual FoxPro-Datenbank arbeitet. Eines Tages entschließt er sich, einen Server zu kaufen und darauf den SQL Server zu installieren, aber weiterhin Ihre Anwendung zu nutzen. In dieser Situation haben Sie ein Problem, da zwei unterschiedliche Datenbanken vorhanden sind und Ihre Anwendung nur in der Lage ist, mit einer Visual FoxPro-Datenbank zu arbeiten. Venelina zeigte in ihrer Session eine mächtige Lösung, den Einsatz des CursorAdapters.

Der CursorAdapter ist die erste Visual FoxPro-Klasse, die es uns ermöglicht, unsere Anwendung mit einer anderen Datenquelle zu verbinden, einfach durch das Auswechseln der Verbindung zur Laufzeit. Venelina zeigte auf, dass der Einsatz des CursorAdapters der Weg ist, eine Anwendung zu entwickeln, die mehr als eine Datenbank verwendet, beispielsweise den SQL Server oder Oracle.

Anschließend beschrieb Venelina die Arbeitsweise des CursorAdapters, wie er sich mit einer nativen, ADO- oder ODBC-Datenquelle verbindet und in einen Cursor von Visual FoxPro konvertiert. Anschließend werden die Daten im Cursor geändert und die Datenquelle auf dem Server wird aktualisiert, aus der wir die Daten gelesen haben.

Der CursorAdapter verfügt über viele unterschiedliche Ereignisse. Venelina erwähnte die wichtigen von ihnen. Hier einige Beispiele:

- BeforeCursorClose
- AfterCursorClose

Ein anderes Thema der Session waren die Einstellungen für den Datenzugriff. Dafür können Sie den CursorAdapter Builder verwenden, der sehr hilfreich ist, da er Ihnen hilft, diese Einstellungen schnell und einfach vorzunehmen.

Venelina wollte noch einige Bemerkungen über das Aktualisieren von Daten machen. Wenn Sie mit dem CursorAdapter arbeiten, können Sie die Pufferung einsetzen. Selbstverständlich können Sie zwischen der optimistischen und pessimistischen Pufferung wählen. Um die geänderten Daten in die Datenbank zu schreiben, sollten Sie die Funktion TableUpdate() verwenden.

Am Ende der Session sprach Venelina über Konflikte bei der Verwaltung der Daten. Was geschieht beispielsweise, wenn die Daten durch mehr als einen Anwender geändert werden. Die Eigenschaft ConflictCheckType legt fest, wie das Objekt Konfliktprüfungen beim Update oder Löschen durchführt.

In der Session lernten wir eine komplexe Lösung für die Verwendung von mehr als einer Datenquelle in einer Anwendung zur Laufzeit kennen. Venelina demonstrierte alles anhand einfacher Beispiele. Am Ende zeigte sie ein Video und verabschiedete sich mit den Worten: „Geben Sie nicht auf und hängen Sie nicht auf.“

Best practices for class design – Marcia Akins

Die Zeit ist unbarmherzig und wir sind bereits am letzten Tag unserer Konferenz angelangt, aber auch der letzte Tag soll großartig sein. Wir haben uns entschieden, Marcias letzte Session zu besuchen und wir haben es nicht bereut. Marcias Thema war das beste Vorgehen beim Klassendesign und wie üblich begann Marcia mit einem Video und diesmal hatte sie auch eine Menge süßer Preise für Entwickler vorbereitet.

Eine Frage: Wann sollten wir Klassen erstellen? Die Antwort ist einfach. Marcia teilte uns mit, dass wir Klassen nur dann erstellen müssen, wenn sie wieder verwendbar sind, dass der Code also mehr als einmal verwendet wird. Ein Beispiel dafür ist eine Klasse, die einen Kalender repräsentiert. Ein anderer Grund ist, dass der Aufwand gerechtfertigt sein muss. Wenn es keinen Grund gibt, eine Klasse zu erstellen, lassen Sie es, denn dann ist es Zeitverschwendung. Als Beispiel führte sie Entwickler an, die für alles Klassen haben, auch für Schriften. Sie nannte dies die Klassenkrankheit.

Marcia führte aus, was wir in einer Klasse benötigen und was sinnlos ist. Sie unterschied drei Gruppen:

- Muss durch das Objekt erledigt werden (verschieben des Datensatzzeigers)
- Kann durch das Objekt erledigt werden (Anzeigen einer Meldung)
- Sollte durch das Objekt erledigt werden (Sicherstellen, dass eine Tabelle geöffnet ist)

Anschließend sprach Marcia über die Verwaltung von Hierarchien. Wir haben zwei Hierarchien:

- Die Klassenhierarchie (Probleme, wenn die Funktionalität zu umfangreich ist; weitgehend ein Thema zur Designzeit)
- Die Objekthierarchie (Probleme, wenn ein Objekt Referenzen auf ein anderes Objekt hält, weitgehend ein Problem zur Laufzeit)

Sie erwähnte auch einige Faustregeln, beispielsweise Code in Methoden zu schreiben, nicht in Ereignissen usw.

In der Session erhielten wir viele Tipps für das effektive Klassendesign sowie für die Arbeit mit Klassen. Marcia achtete darauf, dass uns nicht langweilig wurde, sie hielt uns die gesamte Zeit konzentriert und wer eine intelligente Frage stellte oder eine intelligente Antwort gab, erhielt ein Bonbon. Ich muss zugeben, dass Marcias Erfahrungen bewundernswert sind.

Best practices for remote data access – Andy Kramek



Haben Sie sich jemals überlegt, Ihre Daten an einer anderen Stelle zu halten als in Ihrer Anwendung? Einige von uns hatten diesen Gedanken bereits, andere nicht. Diese Session behandelt die Speicherung der Daten abseits Ihrer Anwendung, die Unterschiede zwischen ODBC und OLE DB sowie viele andere Themen.

Zunächst teilte Andy uns mit, was remote Daten sind. Es handelt sich um Daten, die von der Anwendung getrennt sind. Das bedeutet, dass die Daten im SQL Server, in Oracle oder eventuell in MySQL gespeichert sind. Für dieses Vorgehen gibt es einige Pros und Kontras.

- Gute Integrität der Daten – Pro
- Sicherheitsfunktionalitäten – Pro
- Kein Werkzeug für Endanwender – Kontra
- Kein direkter Datenzugriff
- Vieles anderes

Wir haben eine Stelle, an der unsere Daten gespeichert sind und wir haben unsere Anwendung. Das einzige, was uns fehlt, ist die Verbindung. Andy beschrieb uns die Grundlagen der Verbindung mit einer Datenbank. Wir haben zwei grundlegende Möglichkeiten:

- OLEDB
- ODBC

Diese beiden Möglichkeiten unterscheiden sich in der Implementierung und im Ziel. ODBC ist gut geeignet für relationale Daten und die Verwendung von ODBC mit VFP gibt einen VFP-Cursor zurück. Andy erklärte, dass ODBC durch die nativen Funktionen von Visual FoxPro perfekt unterstützt wird. Für OLEDB gibt es in Visual FoxPro keine direkte Unterstützung.

Bleibt nur noch die Frage: Was ist besser? Andy erwähnte einige der Mythen, beispielsweise, dass OLE DB schneller ist als ODBC oder dass ODBC tot ist. Er meinte aber, dass es keine richtige Antwort gibt. Eines ist aber offensichtlich. Visual FoxPro ist für die Arbeit mit ODBC entworfen worden.

Anschließend sprach Andy über die Werkzeuge für die Arbeit mit remoten Daten. Er erwähnte Werkzeuge wie den CursorAdapter, die remote Ansicht (nur mit ODBC verwendbar) oder SQL PassThrough (SPT). Remote Ansichten sind für den schnellen und einfachen Zugriff auf die Da-

ten geeignet. Der CursorAdapter ist flexibler als die remote Ansicht und es ist möglich, ihn über bestehenden Code zu implementieren. SQL PassThrough hat gute Fähigkeiten für den Ad-Hoc-Zugriff auf die Daten.

Am Schluss sprach Andy über andere Dinge, beispielsweise die Befehlsausführung mit remoten Daten, das Transaktionsmanagement und am Ende über gespeicherte Prozeduren.

In der Session verbreitete Andy eine gute Laune und die Session war voll mit Tipps und Tricks, wie wir unsere Anwendungen ohne größere Probleme mit remoten Daten verbinden. Außerdem verwendete Andy viele Beispiele und gestaltete die Session sowohl interessant als auch informativ.

Letzte Worte und Verlosung – Rainer Becker

Bevor die Preise verteilt wurden gab Rainer neben vielen Scherzen noch einige organisatorische Details bekannt.

Die vielleicht wichtigste Information war, dass die DevCon im nächsten Jahr bereits geplant und der Zeitpunkt festgelegt ist. Wenn Sie den Bericht über die Willkommenssession gelesen haben, wissen Sie, dass auch die Termine für die folgenden sieben Konferenzen bereits feststehen, um Ihnen damit zu helfen, die folgenden Jahre zu planen. Unter anderem können Sie im nächsten Jahr die folgenden Themen erwarten:



- Service Pack 2 mit einer langen Liste mit Fehlerbeseitigungen
 - NET4COM
 - My.Funktionen
 - Vista Toolkit
 - Erweiterungen am Berichtswesen
 - Upsizing zum SQL Server
 - Weitere Sessions über Backend-Datenbanken (nicht nur SQL Server, sondern auch zu DB/2, Oracle und MySQL)
 - Severbasierte verteilte Verarbeitung mit VFP
 - Verbesserungen der Produktivität / Tipps und Tricks
 - Mappoint / Geo-Coding / geographische Berechnungen
 - Mobile Geräte, Datensynchronisierung
 - Design Patterns, Softwareergonomie
- Vieles, vieles mehr...

In der Zwischenzeit machen Sie weiter mit Ihrer Programmierung, finden Sie weitere ungewöhnliche und besonders coole Möglichkeiten, Ihre täglichen Entwicklungsprobleme zu lösen und – das ist das wichtigste – kommen Sie im nächsten Jahr wieder! Wir sehen uns im nächsten Jahr wieder.

Nach dieser Session, dem offiziellen Ende der diesjährigen Konferenz, nahm jeder noch einen letzten Snack und fuhr heim.

Letzte Worte und Danksagungen

Wir, die Reporter, wollen abschließend noch unseren Dank aussprechen, besonders an:

- Rainer Becker, den Leiter der deutschsprachigen FoxPro User Group und den Veranstalter dieser Visual FoxPro DevCon. Er hat gut für uns gesorgt und hat uns immer gern jede mögliche Unterstützung gewährt. Danke, Rainer.
- Michel Gournier, der die Idee hatte, von der Konferenz zu berichten, für seine Unterstützung und sein Verständnis, da wir häufig die Deadlines vollständig ausgenutzt haben.
- Ruth Mohr und ihr Sohn Raphael für die vielen schönen Bilder im Archiv.
- Yuri Shutenko, der immer engagiert und fleißig seine Arbeit an der Übersetzung der VFP-Hilfe für die russische VFP-Community erledigt und trotzdem immer relaxt ist und sich über jede Konversation freut, für seine moralische Unterstützung sowie für einige Bilder, die Sie im Bildarchiv finden. Danke, Juri.
- Eugen Wirsing für viele schöne Bilder im Archiv. Danke, Eugen.
- Tina, den wahren Engel der Konferenz, die dafür gesorgt hat, dass alles wie geplant lief und das jeder alles zur Verfügung hatte, was er benötigte. Danke, Tina.

Wir möchten uns auch bei allen anderen bedanken, die uns mit einer nicht enden wollenden Hilfe versorgt haben und die es uns ermöglicht haben, Ihnen den Bericht so schnell wie möglich zugänglich zu machen.

Wir möchten Sie bitten, Ihre Erfahrungen von der Konferenz auch mit anderen zu teilen, und Bemerkungen zu der Berichterstattung sowie Ideen für die Verbesserung von Universal Thread im UT-Forum zu veröffentlichen. Wir freuen uns immer über Feedback von Ihnen.

Bis zum nächsten Mal!

Jan Vit



Jan Vit ist Student an der University of Economics in Prag. Seit 1993 programmiert er aktiv und beschäftigt sich zusätzlich mit anderen Bereichen der Informationstechnologie. Seine Computererfahrung begann mit seinem ersten 80386er PC mit FoxPro 1.02, gefolgt von höheren Versionen für DOS. Seit 1999 verwendet er die jeweils neueste Version von Visual FoxPro zusammen mit anderen Programmiersprachen und Werkzeugen. Derzeit verbringt er den Großteil seiner Zeit mit Visual FoxPro 9.0, MS SQL Server und C# in WinForms wie auch in WebForms. Seit 2004 ist er regelmäßiger Sprecher auf der Czech Visual FoxPro DevCon und ist Koautor vieler Berichte der DevCons in Prag und Deutschland. Derzeit entwickelt er vorwiegend Software für Apotheken.

Jan Kral



Ich heiße Jan Kral und studiere an der Jaroslav Seifert's High School in Prag. Meine Interessen sind Computertechnologie und die Medizin. In meiner freien Zeit entwickle ich Anwendungen mit Visual FoxPro 7, 8 und neuerdings mit VFP 9.0, die meisten für die eigene Verwendung. Ich war Koautor UT-Berichte über die tschechische DevCon. Außerdem entwickle ich Webseiten mit PHP, JavaScript, CSS usw. Sie können mich gerne per E-Mail kontaktieren.