



# Session V-AFP

## Active FoxPro Pages

Jochen Kirstätter

---

### Abstrakt

Entwickeln Sie Ihre Webanwendungen mit Ihrer gewohnten Programmiersprache. Das echte Multithreading der AFP reduziert den Speicherbedarf erheblich, während ein ausgeklügeltes, verbessertes Sessionmanagement die problemlose Skalierbarkeit auf mehrere Server ermöglicht. Das AFP ControlCenter zeigt Ihnen alle wichtigen Informationen wie die Ausführungszeit oder die gerade bearbeitete Seite. PlugIns erlauben eine nahezu unbeschränkte Erweiterbarkeit der AFP. Das Debuggen war noch nie einfacher: Mit Visual FoxPro können Sie AFP-Seiten unter Echtbedingungen im Debugger testen. Ein ErrorHandler für Ihre AFP-Seiten gibt Ihnen die volle Kontrolle über die Ausführung Ihrer AFP-Anwendungen. Und natürlich wurden auch viele Wünsche realisiert, wie zum Beispiel ein Dateiupload, das Beenden einzelner AFP-Threads oder die bessere Trennung von Daten und Variablen bei unterschiedlichen Applikationen. Verpassen Sie nicht die neue AFP.

---

### Einführendes Tutorial

Active FoxPro Pages (AFP) ermöglichen es Ihnen dynamische Webseiten zu erstellen. AFP-basierte Webseiten werden wie normale HTML-Seiten behandelt. Sie erstellen und bearbeiten AFP-Dokumente genau wie normale HTML-Seiten.

---

### Was wird benötigt?

In diesem einführenden Tutorial setzen wir einfach mal voraus, dass Ihr Webserver bereits Unterstützung für die AFP bietet und dass alle Dateien mit der Erweiterung *.afp* von der AFP verarbeitet werden. Auf den meisten Webservern ist dies die Standarderweiterung, aber im Zweifelsfälle befragen Sie bitte Ihren Server-Administrator. Wenn Ihr Webserver bereits volle AFP- Unterstützung bietet, brauchen Sie nichts weiter zu tun. Erstellen Sie einfach Ihre *.afp* Dokumente, legen Sie diese im Webverzeichnis ab und der Server wird diese automatisch parsen und interpretieren. Sie müssen weder etwas kompilieren noch benötigen Sie zusätzliche Tools. Sehen Sie diese AFP-Dokumente wie normale HTML-Dokumente - nur eben mit ein paar weiteren, mächtigen Tags, die Ihnen viel Arbeit erleichtern.

---

---

# Ihr erstes AFP-Dokument

Erstellen Sie eine *hello.afp* mit folgendem Inhalt:

```
<html>
<head>
<title>Hello World</title>
</head>
<body>

<%
? "Hello World!"
%>

</body>
</html>
```

Beachten Sie, dass Sie kein CGI-Skript erstellen. Die Datei muss weder ausführbar noch sonstige Spezialitäten haben. Behandeln Sie die AFP-Dokumente wie HTML-Dateien mit speziellen Tags, welche Ihnen die Möglichkeit für phantastische Dinge geben.

Dieses Programm ist sehr einfach und Sie werden sicherlich kein AFP benötigen, um so eine Seite zu erstellen. Als Resultat erhalten Sie: **Hello World**

Falls Sie das Beispiel probiert haben sollten und ein Fehler aufgetreten ist, so stehen die Chance nicht schlecht, dass der Webserver zurzeit keine AFP-Funktionalität bietet. Befragen Sie Ihren Administrator zu einer Installation der AFP.

Das obige Beispiel stellt die Verwendung der speziellen AFP Tags dar. Wir benutzen `<%` um einen AFP Tag zu starten, schreiben den eigentlichen AFP-Code und verlassen den AFP-Modus mittels `%>` als schließendem Tag. Sie können jederzeit und immer eine Vermischung von HTML- und AFP-Modus durchführen. Jedoch achten Sie auf 'Wohlgeformtheit' - Öffnen und Schliessen der Tags, dann den nächsten.

---

## Etwas nützlicheres

Wollen wir mal etwas Nützlicheres probieren. Wir werden im nächsten Beispiel ermitteln welchen Browser diejenige Person verwendet, die sich diese Seite anschaut. Um dies zu ermöglichen, fragen wir den "User Agent" ab, welcher der Browser immer als Teil seines Requests an den Server sendet. Diese Information `HTTP_USER_AGENT` geben wir direkt wieder aus. Dazu verwenden wir folgenden einfachen Code:

```
<%? REQUEST.ServerVariables("HTTP_USER_AGENT") %>
```

Für Ihren aktuellen Browser erhalten Sie diese Ausgabe:

**Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7; .NET CLR 1.0.3705; .NET CLR 1.1.4322; .NET CLR 2.0.50727) Gecko/20060909 Firefox/1.5.0.7**

Es viele weitere Variablen, die automatisch vom Webserver zur Verfügung gestellt werden. Sie erhalten eine vollständige Liste, indem Sie eine Datei mit folgendem Inhalt erstellen:

```
<%? SERVER.Info() %>
```

Danach rufen Sie diese Datei in Ihrem Browser auf und die AFP wird Ihnen eine Seite mit vielen hilfreichen Informationen über sich selbst und allen zur Verfügung stehenden Variablen erzeugen.

Sie können auch mehrere AFP Anweisungen innerhalb eines AFP Tags erstellen und somit einen Programmblock, der wesentlich mehr als nur etwas ausgeben kann, erzeugen. Im nächsten Beispiel, überprüfen wir auf den Internet Explorer:

```
<%
if "MSIE"$REQUEST.ServerVariables("HTTP_USER_AGENT")
? "You are using Internet Explorer<br>"
endif
%>
```

So, hier sehen Sie bereits einiges an neuen Konzepten. Wir haben da eine "if"-Anweisung. Falls Sie sich ein wenig mit Programmiersprachen wie etwa C oder VFP auskennen sollten, dann erscheint Ihnen das auf Anhieb logisch. Falls nicht, konsultieren Sie bitte die Doku oder nehmen die ersten Kapitel eines Programmierhandbuchs zur Kenntnis.

Das zweite Konzept ist der \$ Funktionsaufruf. \$ ist eine eingebaute Funktion der AFP, welche einen Zeichenausdruck innerhalb eines weiteren Zeichenausdrucks sucht. In unserem Beispiel suchen wir den Ausdruck "MSIE" innerhalb des HTTP\_USER\_AGENT. Wenn der Ausdruck gefunden wird, liefert die Funktion ein True und wenn nicht, ein False. Falls ein True zurückkommt, wird die nächste Anweisung ausgeführt.

Wir wollen das nun ein wenig verbessern und Ihnen zeigen, wie Sie problemlos in und aus dem AFP-Modus kommen, sogar innerhalb eines AFP-Blockes:

```
<center><b>
<%
if "MSIE" $ REQUEST.ServerVariables("HTTP_USER_AGENT")
%>
You are using Internet Explorer
<%
else
%>
You are not using Internet Explorer
<%
endif
%>
</b></center>
```

Statt dass wir eine Ausgabeanweisung verwenden, verlassen wir einfach den AFP Modus und senden direkt HTML an den Browser. Der entscheidende und zugleich wichtige Aspekt dabei ist, dass der logische Verlauf des Skriptes absolut aktiv bleibt. Lediglich ein HTML-Block wird letztendlich an den Betrachter zurückgesandt. Wenn wir das Beispiel direkt ausführen, erhalten Sie folgendes Ergebnis:

### **You are not using Internet Explorer**

Um die gleiche Funktionalität mit weniger Codezeilen zu erstellen, bietet sich die IIF-Funktion der AFP an. Die Anweisung würde dann so aussehen:

```
IIF("MSIE" $ REQUEST.ServerVariables("HTTP_USER_AGENT"), ;
    "You are using Internet Explorer", ;
    "You are not using Internet Explorer")
```

Bitte beachten Sie die Strichpunkte, die zum Erweitern einer Anweisung über mehrere Zeilen genutzt werden.

---

## Arbeiten mit Formularen

Eine der stärksten Eigenschaften der AFP ist das Verarbeiten und Erstellen von HTML-Formularen. Es ist wichtig das Basiskonzept zu verstehen, dass Ihnen jedes Formular-Element innerhalb des AFP-Dokumentes als Variable zur Verfügung stehen kann. Dabei wird der gleiche Name wie im HTML-Formular verwendet. Dies erscheint auf den ersten Blick vielleicht ein wenig verwirrend, aber ein kleines Beispiel hilft da gleich. Nehmen wir an, Sie haben eine Seite mit folgendem Formular:

```
<form action="action.afp" method="post">
  Your name: <input type="text" name="name">
  Your age: <input type="text" name="age">
  <input type="submit">
</form>
```

Da gibt's nichts Spezielles zu entdecken - es ist ein pures HTML Formular ohne jegliche spezielle Erweiterungen oder Tags. Wenn der Benutzer nun die Felder füllt und das Formular abschickt, wird die *action.afp* aufgerufen. In diesem AFP Dokument haben Sie dann folgende Zeilen:

```
Hi <? REQUEST.Form("name")%>.
You are <? REQUEST.Form("age")%> years old.
```

Es sollte offensichtlich sein, was nun tatsächlich passiert.

## Active FoxPro Pages IntelliSense Extension

Seit Mitte des Jahres steht eine Erweiterung für Active FoxPro Pages zur Verfügung, die das vorhandene IntelliSense von Visual FoxPro mit den verfügbaren Objekten der AFP erweitert und auf diese Weise die Programmierung erleichtert. Man hat sozusagen die komplette Dokumentation aller Objekte inklusive Eigenschaften und Methoden samt Parameter direkt an den Fingerspitzen. Zur Aktivierung der Active FoxPro Pages IntelliSense Extension (aie) führen Sie die Datei *AFP3IE.exe* im AFP3-Verzeichnis aus.

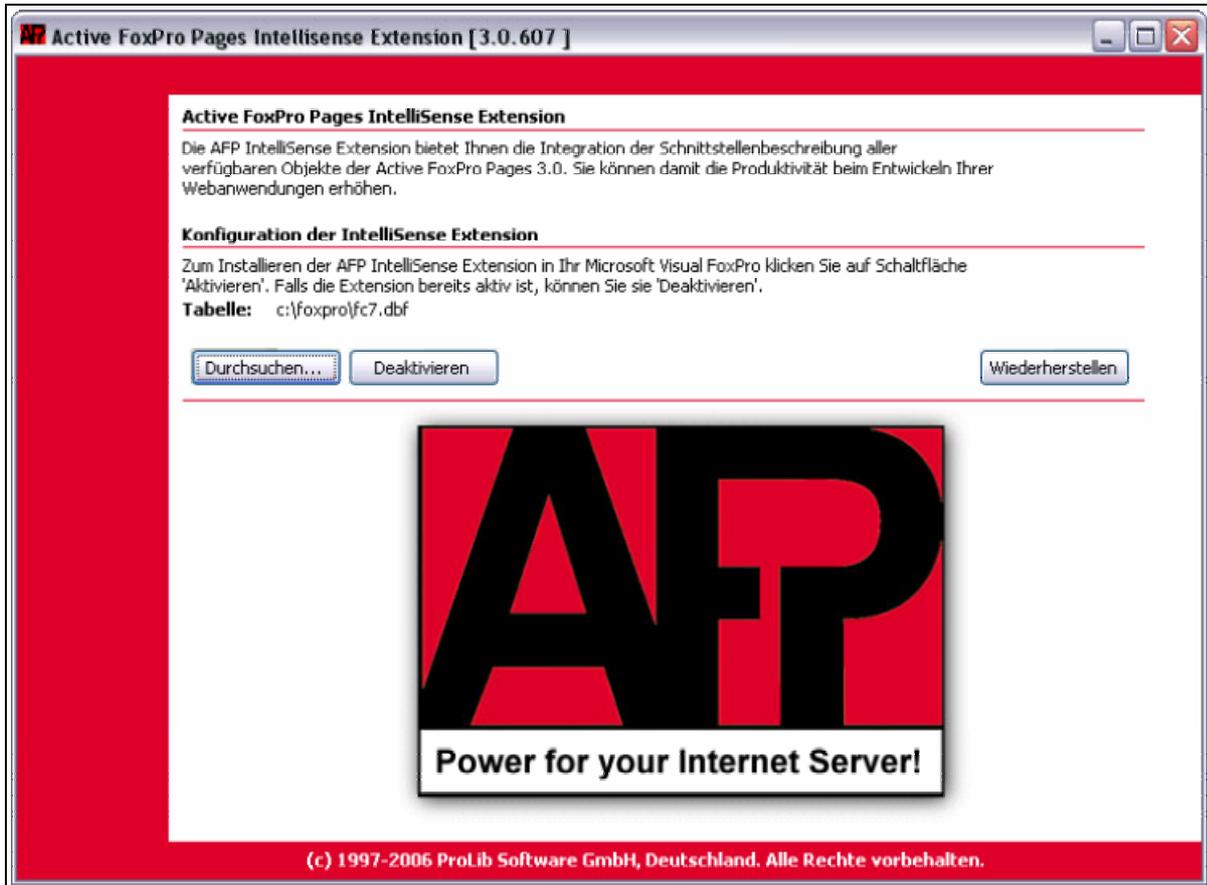
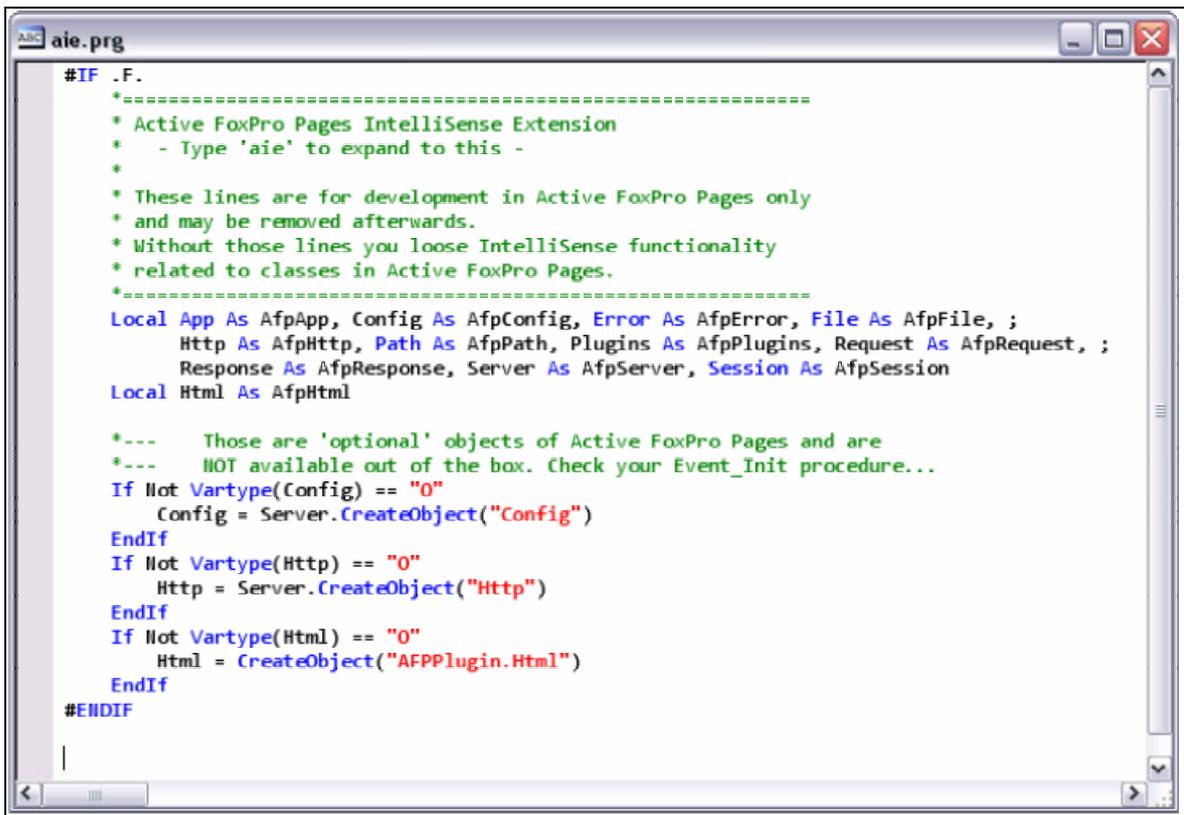


Abbildung 1: Konfiguration der Active FoxPro Pages IntelliSense Extension

Mittels der Aktivierung wird die VFP\_FoxCode-Tabelle mit den Typen der AFP erweitert. Danach kann man durch das Kürzel **aie** die Extension sozusagen in VFP auslösen und nutzen:



```
#IF .F.
*-----
* Active FoxPro Pages IntelliSense Extension
* - Type 'aie' to expand to this -
*
* These lines are for development in Active FoxPro Pages only
* and may be removed afterwards.
* Without those lines you loose IntelliSense functionality
* related to classes in Active FoxPro Pages.
*-----
Local App As AfpApp, Config As AfpConfig, Error As AfpError, File As AfpFile, ;
    Http As AfpHttp, Path As AfpPath, Plugins As AfpPlugins, Request As AfpRequest, ;
    Response As AfpResponse, Server As AfpServer, Session As AfpSession
Local Html As AfpHtml

*--- Those are 'optional' objects of Active FoxPro Pages and are
*--- NOT available out of the box. Check your Event_Init procedure...
If Not Vartype(Config) == "0"
    Config = Server.CreateObject("Config")
EndIf
If Not Vartype(Http) == "0"
    Http = Server.CreateObject("Http")
EndIf
If Not Vartype(Html) == "0"
    Html = CreateObject("AFPPlugin.Html")
EndIf
#ENDIF
```

Abbildung 2: Active FoxPro Pages IntelliSense Extension in Aktion

Der erzeugte Code wird bedingt durch die Präprozessor-Anweisungen nicht in das Kompilat integriert und kann daher beliebig häufig in Funktionen und Methoden genutzt werden.

---

## Literatur und Ressourcen

Die Produktdokumentation steht vollständig online zur Verfügung.

[Active FoxPro Pages](#)

[AFP FAQ](#)

---

## Über den Autor

Die ersten Anwendungen im Jahr 1995 entstanden im Bereich Internet. Diese basierten auf Linux, einer Skriptsprache wie Perl oder PHP, und MySQL als Datenbank. Mit der Anstellung im Frühjahr 1999 bei der Firma Kheops GmbH, Kaiserslautern, verlagerte sich der Schwerpunkt auf Anwendungsentwicklung in Visual FoxPro, Microsoft SQL Server und seit 2002 auf Webentwicklung mit [Active FoxPro Pages](#). Die [LiveFAQ](#) der dFPUG basiert auf dem kostenfreien [AfpWiki](#) von ihm.

Jochen Kirstätter ist Microsoft Certified Professional (MCP) und Microsoft Most Valuable Professional (MVP) 2006 für Visual FoxPro. Er betreut seit dem Frühjahr 2004 den [dFPUG-Stammtisch in Speyer](#) (Region Rhein-Neckar) und ist durch seine Community-Aktivitäten seit 2003 Mitglied im Microsoft CLIP Programm sowie Regionalvertreter der [SQL PASS in Rhein-Neckar](#). Sie erreichen ihn unter [jochenk@prolib.de](mailto:jochenk@prolib.de) (MSN) und ICQ 20326330.