WHITE PAPER

FoxInCloud

WHITE PAPER

March 2011

EXPAND YOUR OPPORTUNITIES

Exposing your application in the cloud secures your existing client base and eases customer acquisition through instant trial and SaaS offerings.

Your application's current value is limited by being 'thick' client, which drives a heavy work load for pre-sales trial and demos, after-sales deploying and maintenance.

A Web interface will make it accessible to any potential client from all around the world, who will be immediately able to evaluate it in real operating conditions.

Easy access and SaaS commercial offerings will provide your business additional flexibility, reduce perceived risk and accelerate favorable decisions.

Clients already equipped with your 'thick' client version may slide to the 'thin' version at their own pace, while saving on TCO and maintenance.

Be first in the cloud

With FoxInCloud, your application will be 'cloudified' long before your competitors'; you will gain a real differential advantage on a decisive market segment.

STYLE YOUR INTERFACE

Style increasingly matters

When technology, price and functionality are comparable, management decision is often made on user interface look and feel.

One known limitation of Visual FoxPro applications is their ‘old fashioned’ user interface, and styling capabilities dependent on Windows settings.

Offer your Visual FoxPro application a fresh Web 2.0 look

Based on open web technologies, HTML and CSS, FoxInCloud opens a wide for creativity and graphical innovation; your application’s Web version will provide a fresh design that your clients will notice.

While FoxInCloud generates HTML templates based on your application’s original Visual FoxPro forms, you keep control over graphic behavior through CSS classes, custom HTML rendering or Javascript, bringing you almost unlimited capabilities.

Interface and application code being clearly separated, you easily integrate graphic design from a web design studio.

You also benefit from a wide array of styles and Javascript libraries, many of them being free or low-priced: jQuery, Sencha/Ext.js, Telerik, etc.

JUMPSTART INTO THE WEB

Offer your team a jump-start into the Web

You planned moving to the web for years and your team worries keeping developing in Visual FoxPro?

With FoxInCloud, flip to the web on the Bleeding Edge

FoxInCloud removes the pain of acquiring a new server technology, rewrite your application, testing, and burden of maintaining 2 different versions simultaneously.

Skill your team on fast-moving, vivid technologies

After several weeks of adaptation, you can start testing your application in web mode, explore possibilities offered by this new environment.

Your team can concentrate on pure Web development: design, look and feel, customization, etc.; it acquires HTML, CSS and Javascript skills as far as needed, while continuing regular application maintenance.

PRESERVE YOUR VFP CODE

Your Visual FoxPro application has a single real handicap: its thick client interface whereas market demand strongly leans to thin, web interface.

FoxInCloud address this demand: only the visual surface is moved to the web client, the rest of your application works just as now: events, business code, database, etc. Your code can keep the same structure without complex layering into n-tier or design patterns.

You build additional value from your client-validated code base instead of 'migrating' to another server technology: you can avoid costs, shorten time to market, and lower the risk of functional regression.

Adapt your code without changing it

In your code base, some instructions need be adapted for your application to operate in web mode. These adaptations are light and without functional impact; using an alternate syntax, moving code into another method, calling methods inherited from FoxInCloud classes.

FoxInCloud Adaptation Assistant guides you during the adaptation process: Identifies instructions to adapt, Provides a shortcut to code snippet, Explains how manual adaptation should be performed, Redefines native classes into FoxInCloud classes, Performs single instruction adaptations automatically

Invest in durable technologies

With FoxInCloud, you adopt a new technology while preserving your acquired skills. Your learning investment focuses on universal web technologies, the use of which is growing continuously: HTML, CSS Javascript. Your development is entirely reusable whenever you switch to another server technology.

BET ON DURABLE TECHNOLOGIES

FoxInCloud gives your application an additional 10-year lifetime and you invest in standard, open client technologies, compatible with any client and server technology.

Target the 'right fox'

Having invested years of development on Visual FoxPro software, since the march 2007 ‘message to the community’, you seek a durable evolution path.

Everything would be so simple if some magic wand could translate your tens of thousand lines of code into another language. This instrument remaining unfound, migration remains a costly and risky operation, without any client benefit.

Moreover, server technology landscape keeps moving: .Net gets somewhat unpredictable (some expert view it as a moving target), free solutions such as LAMP are increasingly appraised by professional markets.

Visual FoxPro support is guaranteed until 2015; beyond it seems obvious that Microsoft will keep its backward compatibility policy in order to maintain the huge market of corporate workstations. And whenever a compatibility break would occur, the Windows Server support scheme would extend the application life time by an additional 10-year.

Choosing FoxInCloud today gives your web application a 15-year lifetime… such a delay seems well enough to choose and implement a new server strategy, based on the most mature and productive technology.

Bet on durable technologies (...)

What is current Visual FoxPro status?

Three years after its latest evolution, Visual FoxPro still yields nearly 1 million Google requests per month worldwide, to be compared with 14 million for C# and 40 millions for Java.

Many community initiatives such as VFPx, Guineu and VFPCompiler demonstrate community's involvement and faith in Visual FoxPro.

Many articles relate Microsoft's efforts to bring Visual FoxPro applications and developers towards .Net platform, with such hypotheses as integrating Visual FoxPro syntax into Visual Studio.

Visual FoxPro remains on Microsoft's radar screens, knowing that such a community maintains tens of thousands applications, meaning hundreds of thousands Windows licenses under attack by growing competition.

Server: take advantage of your Visual FoxPro application

Running on a web server, your application overcomes most criticisms usually addressed to Visual FoxPro :

- No more dependency on client hardware and software
- Visual FoxPro data are protected against undesired access and unexpected shutdown
- All data requests are local, much faster than on LAN
- Web user interface benefits from the latest graphic capabilities: HTML 5, CSS 3, etc.

Bet on durable technologies (…)

Visual FoxPro is well known for being tremendously fast in managing data and strings, both giving excellent capabilities for running as a Web server. Encapsulated in a COM object, mute and invisible, a Visual FoxPro application resembles any other one and raises very few objections.

Client: invest on durable technologies

The main area where a cloudified application defers from its LAN equivalent is the user interface: to be as comfortable as a thick client interface, web interface should be 'rich', IOW able to activate, hide, move, modify the page's content. This result may be obtained through 2 types of solutions:

- **Proprietary** solutions require a browser extension: Adobe Flash and Microsoft Silverlight fall into this category
- **Open/standard** solutions requires no more than the browser: HTML, CSS, Javascript

Proprietary solutions rely on their support by device manufacturers, such as mobile devices.

As of now, Apple has no plan to support Flash and Silverlight in Safari, the browser for iPhone and iPad, two device very popular among managers and influential experts. Similarly, whereas its 'Androd' operating system has a growing adoption, Google made no announcement in that sense. Silverlight support by Nokia's Symbian was announced back in 2008; beta test phase began in mid 2010…

An application relying on a proprietary technology would miss more than a quarter of the mobile devices market, already prominent in youth's Internet usage patterns.

Bet on durable technologies (…)

Without real visibility on proprietary solution, the standard HTML/CSS/Javascript technology appears to be the most pertinent choice; it guarantees a universal support and benefits from a wide and lively offering of skills and tools.

LAN / Web: your client choose

Web applications address a growing demand of most of your clients, maybe not all of them: each client may have a specific road map.

FoxInCloud addresses this contrasted expectations: with an application common to LAN and Web environments, any client may slide to the web whenever he wishes, with the very same functionality, ergonomics and data. He may even move users groups one by one.

Instead of a migration, FoxInCloud provides a gradual evolution path with possible fallback to 'traditional' operations.

LIGHT CLIENT AND SERVER

FoxInCloud server relies on a standard Windows environment, without requirement on version or extension; its memory and CPU consumption is similar to your Visual FoxPro application's, you know how sober itis…

FoxInCloud server: agile as the fox himself

FoxInCloud server relies on the wConnect framework, the reference Visual FoxPro web extension for over 10years; compatible with all Windows Server versions, it does not depend on .Net framework, supports up to 32servers per site and starts automatically at first request.

The secured Web administration console supports most maintenance operations: live app. update, log browsing, application parameters update, etc. Any applicative error is logged and triggers a mail to administrator.

The FoxInCloud server hosts your application, generates HTML, manages users state, processes requests. Compiled in Visual FoxPro, it uses less than 2Mb memory, .2 seconds and 10kb bandwidth per request.

HTML client: powerful, extensible and lightweight

Your Visual FoxPro forms are 'cloned' into HTML / CSS code and Javascript widgets: spinner, grid, slider, etc. Generated code is stored in files you can easily integrate to your site with standard include mechanisms: SSI, ASP, PHP, etc.

You can adjust each object's style by means of: standard CSS selector (#id, .class, etc.), custom HTML rendering or Javascript with your preferred library: prototype/scriptaculous, jQuery, Ext, etc.

Light client and server (…)

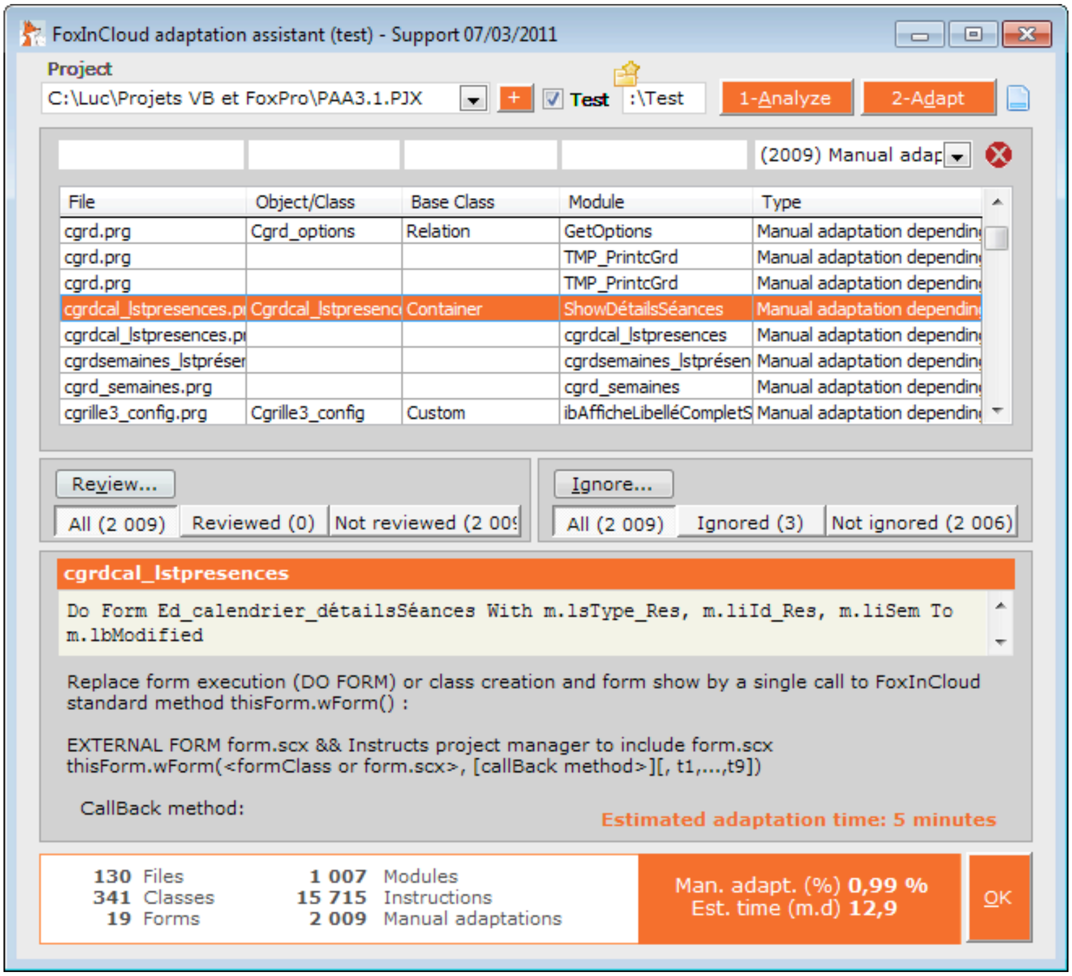
Each event implemented in the original Visual FoxPro form is also implemented in its HTML 'clone'. A standard FoxInCloud Javascript method receives user events, transmits them by AJAX to the server for execution by the matching Visual FoxPro event method, updates the HTML page according to the orders received back.

Trading only granular data, a FoxInCloud request consumes less than 10kb bandwidth in average.

FoxInCloud.js and support library (less than 300kb) are loaded once for all at first page.

Analyse your code FREE

Download, install and execute FoxInCloud Adaptation Assistant.



Project: C:\Luc\Projets VB et FoxPro\PAA3.1.PJX ☑ Test :\Test

(2009) Manual adap

File	Object/Class	Base Class	Module	Type
cgrd.prg	Cgrd_options	Relation	GetOptions	Manual adaptation dependin
cgrd.prg			TMP_PrintcGrd	Manual adaptation dependin
cgrd.prg			TMP_PrintcGrd	Manual adaptation dependin
cgrdcal_lstpresences.pı	Cgrdcal_lstpresenc	Container	ShowDétailsSéances	Manual adaptation dependin
cgrdcal_lstpresences.pı			cgrdcal_lstpresences	Manual adaptation dependin
cgrdsemaines_lstprésen			cgrdsemaines_lstprésen	Manual adaptation dependin
cgrd_semaines.prg			cgrd_semaines	Manual adaptation dependin
cgrille3_config.prg	Cgrille3_config	Custom	ibAfficheLibelléCompletS	Manual adaptation dependin

All (2 009) | Reviewed (0) | Not reviewed (2 00

All (2 009) | Ignored (3) | Not ignored (2 006)

cgrdcal_lstpresences

```
Do Form Ed_calendrier_détailsSéances With m.lsType_Res, m.liId_Res, m.liSem To
m.lbModified
```

Replace form execution (DO FORM) or class creation and form show by a single call to FoxInCloud standard method thisForm.wForm() :

EXTERNAL FORM form.scx && Instructs project manager to include form.scx
thisForm.wForm(<formClass or form.scx>, [callBack method>][, t1,...,t9])

CallBack method:

Estimated adaptation time: 5 minutes

130	Files	1 007	Modules
341	Classes	15 715	Instructions
19	Forms	2 009	Manual adaptations

Man. adapt. (%) 0,99 %
Est. time (m.d) 12,9

FoxInCloud Adaptation Assistant identifies instructions needing adaptation and details the type of adaptation to perform: automated, manual, tips ...

Note: FoxInCloud support scope expands rapidly! Make sure to execute Adaptation Assistant on a regular basis to have an up-to-date status of your application.

Adapt your code FREE

AUTOMATED ADAPTATIONS

During automatic adaptation, FoxInCloud Adaptation Assistant performs the following sequence of operations:

- Copies FoxInCloud public source code into folder [VFP]\Tools\AB; see list of copied files,
- Sub-classes FoxInCloud base classes into a user-defined vcx,
- When running in test mode (see below), copies project files into the test directory you choose,
- Replaces references to natives base classes by FoxInCloud base classes derived from Visual FoxPro native classes,
- Adapts code according to details enumerated by Adaptation Assistant.

Automated adaptation may be performed in test or production mode. In test mode, adaptation occur on a copy of your project that Adaptation Assistant creates before hand into the directory you specify; furthermore, Adaptation Assistant identifies absolute paths (e.g. c:\…) in copied files and redirects them into test directory.

After performing automated adaptation in test mode, your application works the same in test directory.

> Obviously, it is strongly advisable to perform in-depth tests BEFORE performing automated adaptation on production source code.
>
> Similarly, you should back up your production source code and freeze its evolution until adapted production code has been thoroughly tested, and to keep this backup during a sufficient delay to overcome possible flaws.

MANUAL ADAPTATIONS

Some manual adaptations requires having performed automated adaptations beforehand, some are independent.

Adaptation Assistant indicates what type of adaptation is expected instruction by instruction; you may filter adaptations to batch adaptations of same type.

You can open source code by clicking the ‘review’ command button or double clicking the list; to ease progress follow-up, you can mark an adaptation as ‘done’.

Please make sure to test your application after each large sequence of adaptations.

On top of adaptations indicated by Adaptation Assistant, you will need to perform the following:

- In your set of FoxInCloud base classes sub-classes, implement wUserGet() and wUserSet() methods for user identification during requests
- Separate long processes such as statistics or heavy reporting into asynchronous programs on top of West-Wind wConnect wwAsyncWebRequest class

Publish your site

[this step requires source code under FoxInCloud licence]

CREATE YOUR FOXINCLOUD APPLICATION SERVER

Create your Server and Application Host classes:

```
DEFINE CLASS myServer AS awServer OF awServer.prg OLEPUBLIC && OLEPUBLIC for COM mode
...
ENDDEFINE
DEFINE CLASS myAppHost AS awAppHost OF awServer.prg
...
ENDDEFINE
```

Set up some properties of the above classes:

```
cServerSets: module (class or procedure) SETting server's environment
cProcessSetsClass: Class SETting requests environment
cForms: First level forms, displayed on site pages (*.scx/classes)
cProcessClass: (optional, alternative to tools such as ASP or PHP):
wConnect process class responding to conventional requests (GET, POST,
etc.) with Visual FoxPro code.
```

Cloudify your app in 3 steps (...)

Create the two modules SETTING environment for server (SET PATH, PROCEDURE, CLASSLIB, etc.) and requests (SET DELETED, EXACT, etc.)

Setup server's parameters in an .ini file located in the project folder:

– Application name and title, administrator name and email, etc.

– Virtual and physical paths of site directories: root, images, CSS, script, etc.

Setup FoxInCloud server according to wConnect documentation.

Create your main FoxInCloud server program for development mode execution:

```
* myServer.prg
AW()  loads FoxInCloud resources
awServerStart("myServer")  starts your FoxInCloud server
DEFINE CLASS myServer as awServer of awServer.prg
...
ENDDEFINE
```

CREATE YOUR SITE

Create site pages (HTML, CSS) with your favorite tool: Notepad++, PSPad, Dreamweaver, etc.

Include into the application pages:

- HTML/CSS/Javascript code produced by FoxInCloud from your application's Visual FoxPro forms
- The FoxInCloud CSS and Javascript libraries: FoxInCloud.CSS.inc FoxInCloud.JS.inc...

... by mean of an include directive such as:

```
ASP/PHP: 'Server Side Include' directive:
<!-- #INCLUDE [FILE|VIRTUAL]="your file" #-->

Wconnect: substitution in a '' tag executed by .expandTemplate():
<%=FiletoStr('your file')%>
```

TEST YOUR APPLICATION

Navigate to some of your site's page including one of your application forms, and test the implemented events:

```
http://localhost/my/page.xxx    && where 'xxx' depends on your site
                                technology: asp, php, etc.)
```

BELIEFS

Most clients want web apps, at **low cost**

Your code bears value, not technology

Maintaining **2 versions** of the same software is a nightmare

On the client, **HTML / CSS / Javascript** are tomorrow's winners

Visual FoxPro is **solidified** on a Web server.

Fox InCloud

www.foxincloud.com ı contact@foxincloud.com