



FoxInCloud

A-Z tutorial, Session 1

Requires FAA & FAS V2.0+

Session 1

Understanding FoxInCloud

This session targets users who wish to see the big picture before starting hands-on experience...

it describes how FoxInCloud Application Server (FAS) operates your application on a Web server and interacts with the client browser.

Reproducing steps of this session requires setting break points at various locations of FAS source code (licensed users only).

Though you may follow the steps by browsing the coming slides and applying the steps on your own machine, you may get a better understanding by attending a FoxInCloud workshop or training session; please feel free to mail contact@foxincloud.com for workshops and training sessions to come.

FAS installs the files used in this session into `<VFP9>\Tools\AB\AW\Samples\FIC\FICtuto\`

Understanding FoxInCloud, Agenda



Episode	Subject	Slides
	FoxInCloud Architecture	1
1	Connecting Web Server and FAS	11
2	Displaying app's main form in Browser	17
3	Interacting with a form in Browser	12
		41



FoxInCloud Architecture

Web server (IIS, Apache, ...):
Hosts Web site (i.e. www.mysite.com)

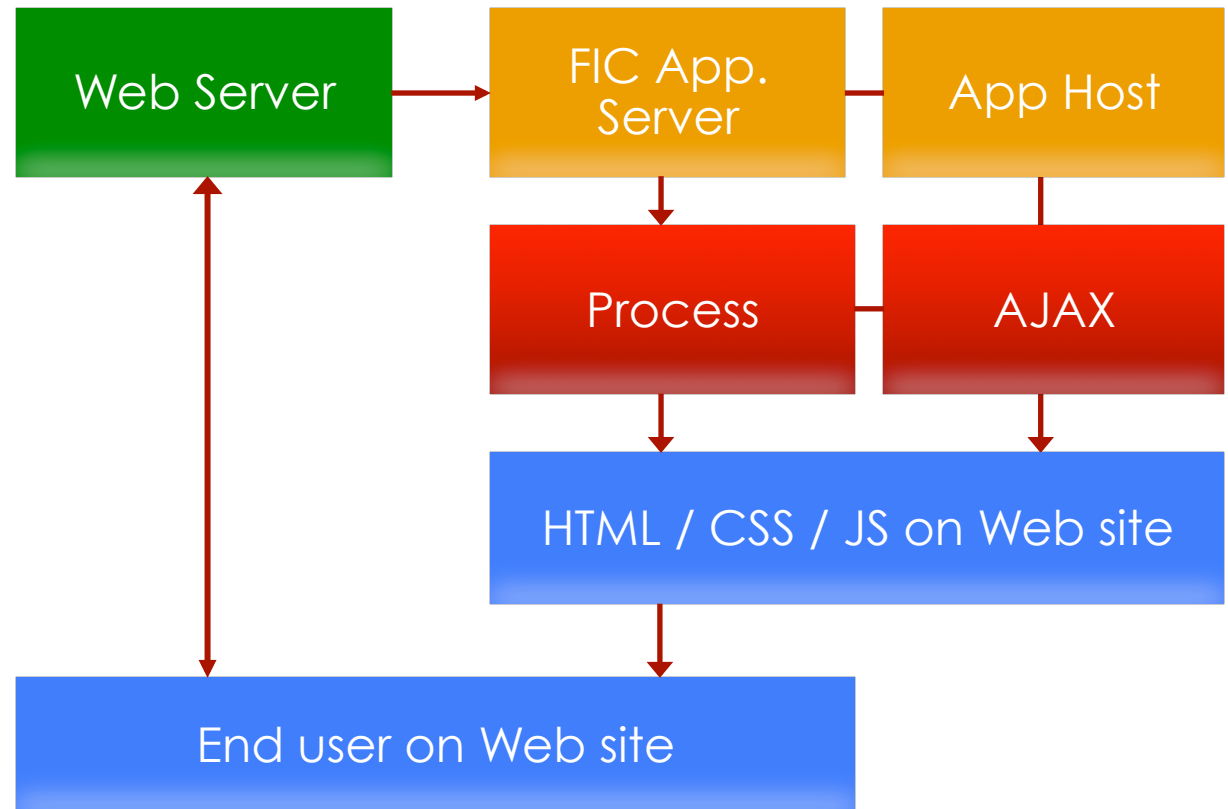
FIC App. Server (FAS):
VFP/wConnect-based
FoxInCloud Application Server

App Host:
Private session where adapted
application runs

HTML/CSS/JS gen:
clones your app's forms into
HTML/CSS/JS which becomes
the designated website (i.e. www.mysite.com)

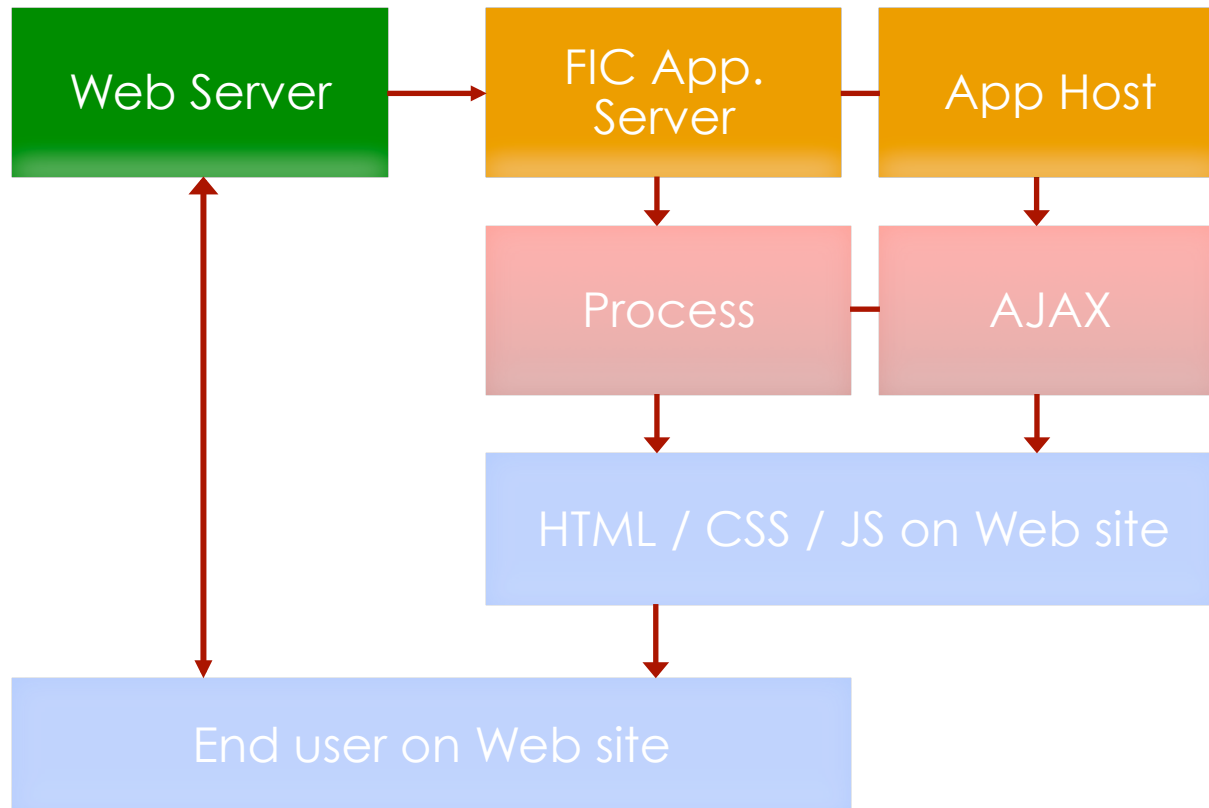
Process:
Processes HTTP requests
received from the site

AJAX:
Processes user actions on forms
thru AJAX requests



Session 1, episode 1

Connecting Web server to FAS



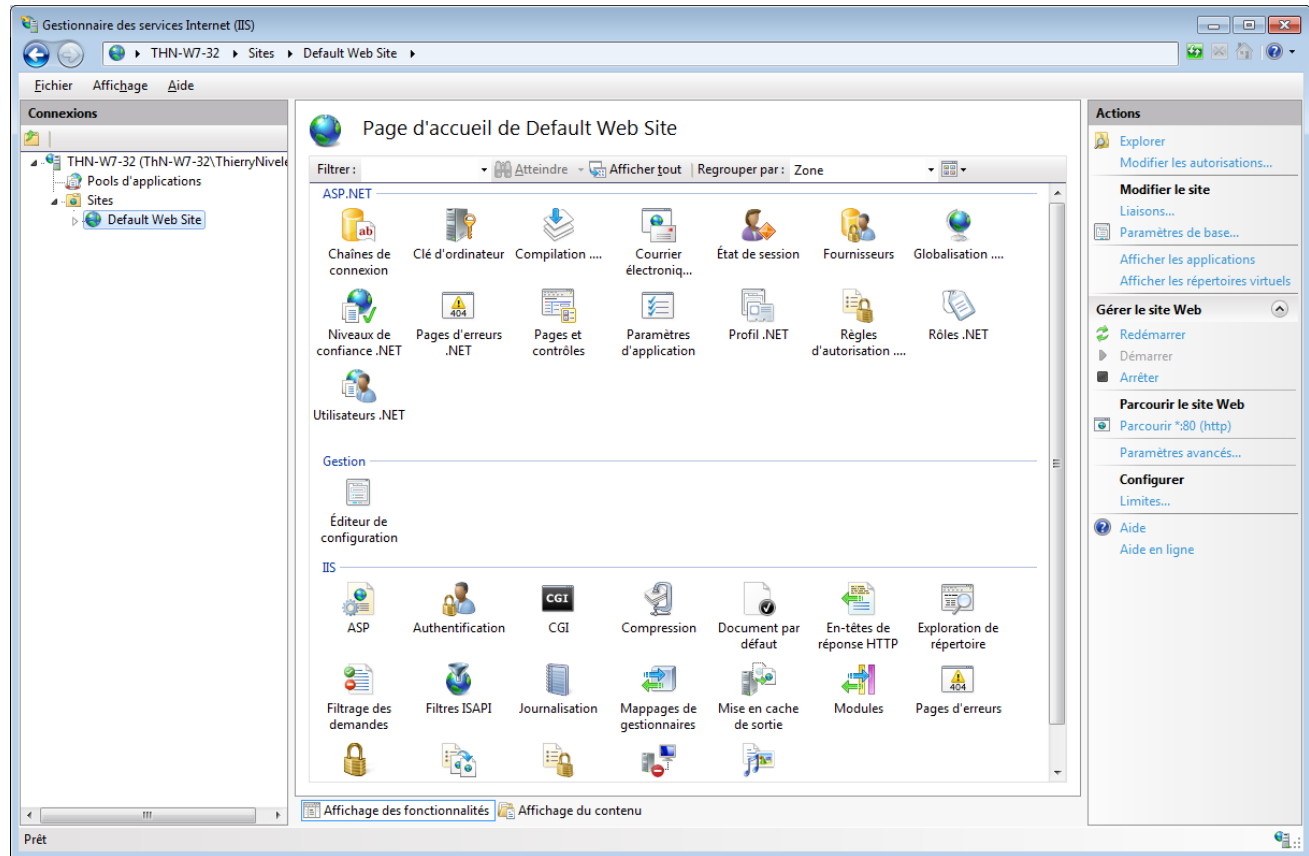


Grabbing requests from localhost web server

IIS comes with most Windows editions

Note: the free FoxInCloud Adaptation Assistant (FAA) automates all IIS settings to come;

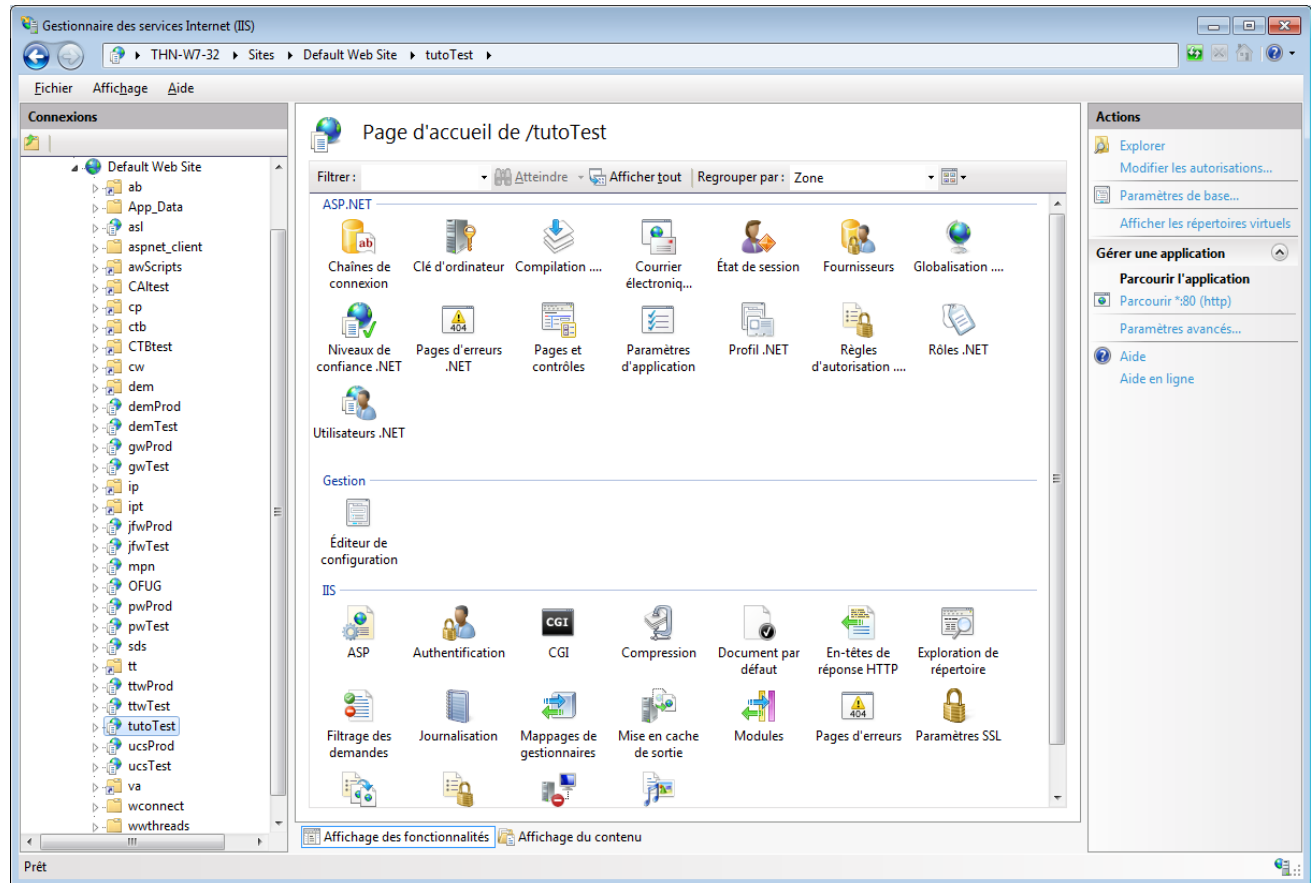
Session 3 demonstrates this automated installation.





Adding an IIS application

An “IIS application” is a virtual directory where an application can run





Mapping an URL extension to your FAS

IIS allows URLs with a given extension to be processed by a given module (.exe or .dll)

The screenshot shows the IIS Manager interface for the 'tutoTest' site. The 'Mappages de gestionnaires' (Script Mappings) tab is active, displaying a list of mappings. Two dialog boxes are open over the interface:

- Modifier le mappage de scripts (Modify Script Mapping):** Shows the configuration for the mapping with the extension `*.tuto`. The executable path is `C:\Program Files\VF9\Tools\AB\AW\Samples\FIC\FicTuto\site\bin\w` and the name is `tutoTest`. A 'Restrictions des demandes...' button is visible.
- Restrictions des demandes (Request Restrictions):** Shows the 'Accès' (Access) tab with the 'Exécution' (Execute) radio button selected, indicating that the mapped module can execute the requested script.

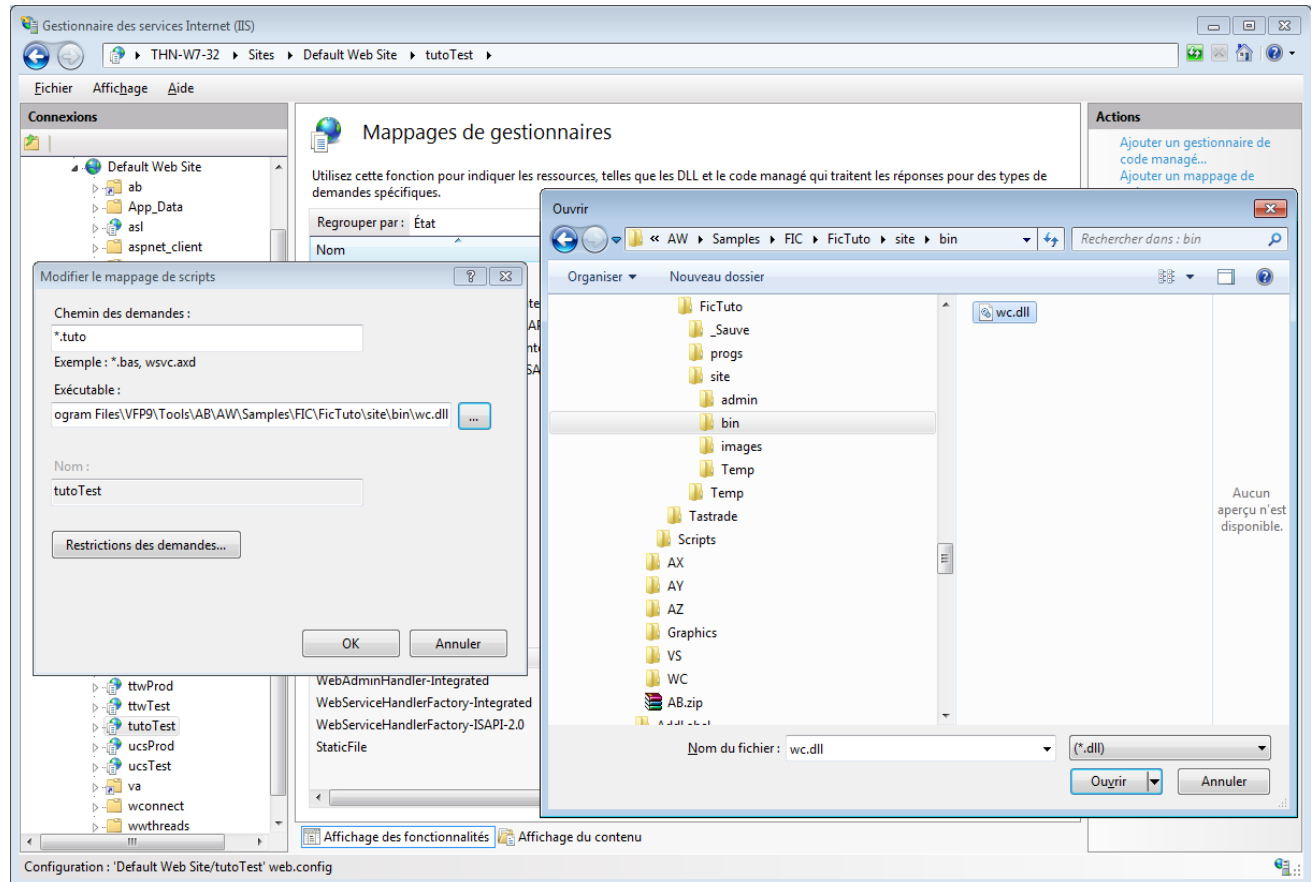
Nom	Chemin d'accès	État	Type de chemin	Gestionnaire
WebAdminHandler-Integrated	WebAdmin.axd	Activé	Non défini	System.Web.Ha
WebServiceHandlerFactory-Integrated	*.asmx	Activé	Non défini	System.Web.Ser
WebServiceHandlerFactory-ISAPI-2.0	*.asmx	Activé	Non défini	IsapiModule
StaticFile	*	Activé	Fichier ou dos...	StaticFileModule



Target module is wConnect's “wc.dll”

FAS relies on
“West-Wind
web
connect”
(wConnect)

wc.dll is the
IIS extension
that
wConnect
provides.





wc.dll reads settings from wc.ini

Main setting is the folder where **wc.dll** and FAS trade request and response files

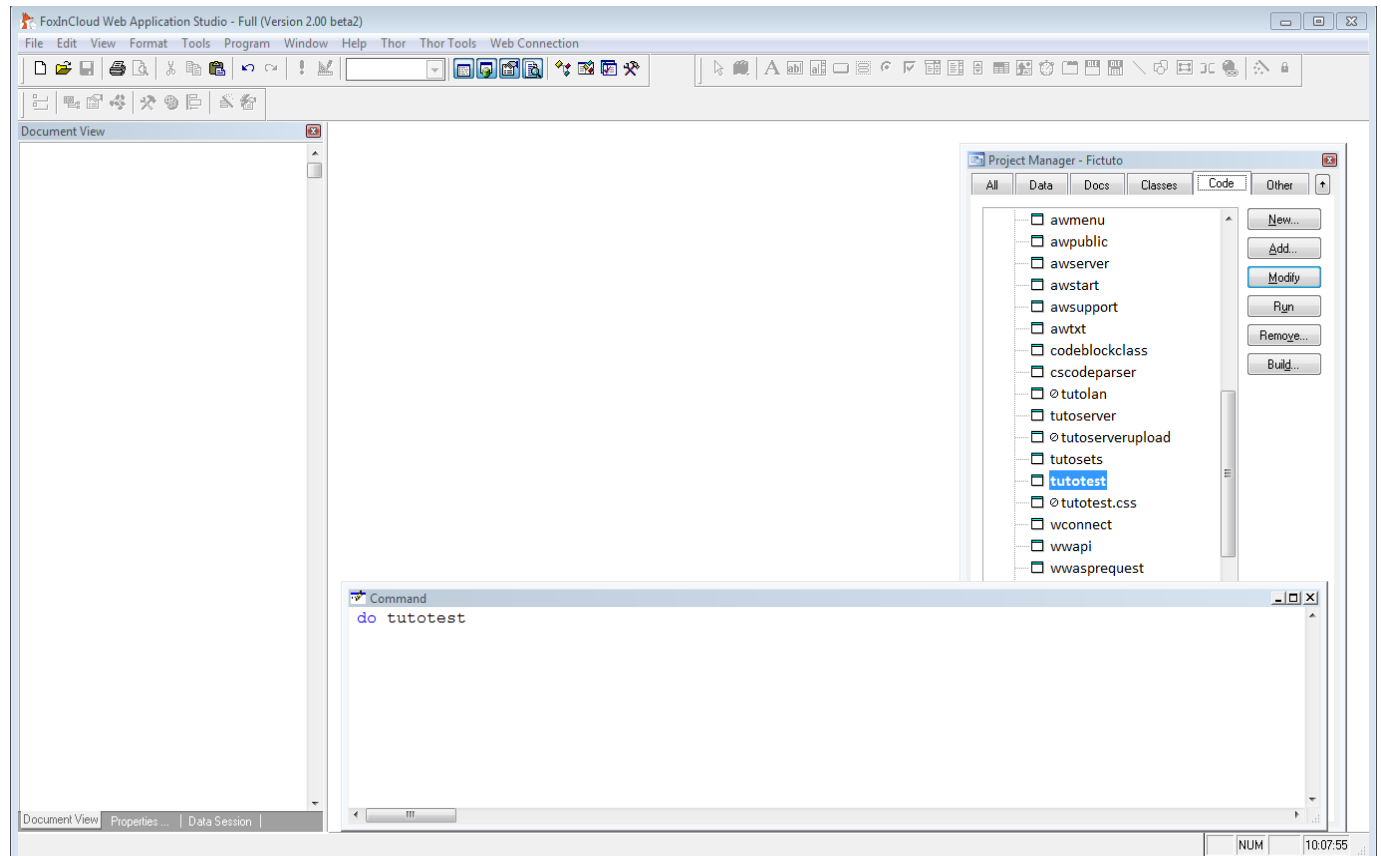
When getting an HTTP request from IIS, **wc.dll** writes request details into a file in this folder.

```
C:\Program Files\VFP9\Tools\AB\AW\Samples\FIC\FicTuto\site\bin\wc.ini - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
wc.ini
1 [wvcgi]
2 ; Directory where wc.dll and FAS trades files
3 Path=c:\Temp\FICtuto\
4
5 ;*** Time to allow request to finish
6 ;*** Process will be terminated after number of secs
7 ;*** specified here.          REQUIRED
8 Timeout=600
9
10 ;*** Specify how often wc.dll polls for 'completion' message
11 ;*** Specify in milliseconds.  REQUIRED FOR FILE BASED
12 PollTime=100
13
14 ;*** Message File Template (1st 3 letters)
15 ;*** Default is "wc_"
16 ;*** Only needed if using a different template
17 Template=wc_
18
19 ;*** Messaging Mechanism of the DLL:  REQUIRED
20 ;*** File      - Original WEB Connection Logic of
21 ;              file based messaging
22 ;*** Automation - Use OLE Automation Server Interface
23 ;*** Interactive - Call up the VFP development environment via Automation
24 Mechanism=File
25 ;Mechanism=Automation
26
27 ; PostMethod - URLEncoded or INI - Match POSTMETHOD in WCONNECT.H
28 PostMethod=UrlEncoded
29
30 ;*** Account for Admin tasks  REQUIRED FOR ADMIN TASKS
31 ;***      NT User Account  - The specified user must log in
32 ;***      Any              - Any logged in user
33 ;***
34 AdminAccount=
3 lines, 68 characters selected
Tab Size: 4
Plain Text
```



Starting FoxInCloud Application Server (FAS)...

Open
ficTuto.pjx
and run
main
program
tutoTest.prg





FAS initializes application

awAppHost
is app's
session
object.

Both
desktop
and Web
versions of
your app
share the
xxxSets
Settings
class.

```
Visual FoxPro Debugger [break] - tutosets.prg;tutosets.init
File Edit Debug Tools Window Help

Call Stack
+ 9 tutosets.init
  8 awapphost.init_app_
  7 awapphost.init_app
  6 awapphost.init
  5 tutoservertest.setserverpropertie
  4 wwserver.init
  3 tutoservertest.init
  2 awserverstart
  1 tutotest.prg

Trace
Object: tutosets Procedure: Init
44 * =====
45 * -----
46
47 PROCEDURE INIT
48
49 DODEFAULT()
50
51 SET PATH TO '..\..\_Data\; ..\_Data\; ..\Classe\; ..\Progs\; ..\Progs\Bitr
52
53 IF NOT wlAutomation()
54   SET CLASSLIB TO ficSample ADDITIVE
55 ENDIF
56
57 && code for both desktop and web modes
58
59 THIS.SET('SYSFORMATS', 'ON') && THIS.SET() restores original SETting whe
60 THIS.SET('TALK', 'OFF')
61 THIS.SET('SAFETY', 'OFF')
62
63 THIS.SET('COLLATE', 'machine')
64 THIS.SET('EXCLUSIVE', 'OFF')
65 THIS.SET('MULTILOCKS', 'ON')
66 THIS.SET('REPROCESS', -2) && AUTOMATIC
67 THIS.SET('DELETED', 'OFF')
68 THIS.SET('EXACT', 'OFF')
69
70 IF m.this.wlLAN
71
```



FAS enters wait state

FAS waits
for timer
events

```
Visual FoxPro Debugger [run] - awstart.prg:awserverstart
File Edit Debug Tools Window Help
Call Stack Trace
Object: Procedure:
, CreateObject(m.tcServerClass);
)
llResult = Vartype(m.goWCserver) == 'O'
ASSERT m.llResult MESSAGE cAssertMsg(Textmerge( [<<Proper(Program())>> - Could not
IF m.llResult;
AND NOT (Vartype(m.tuFormsLaunch) == 'L' AND m.tuFormsLaunch) && CSS generation,

* Put server live online in file polling mode
* =====
ON SHUTDOWN DO ShutIt IN awStart.prg && Allows system shutdown
READ EVENTS
ON SHUTDOWN
* =====
ENDIF

RELEASE goWCserver
ON ERROR

IF m.llResult
CLOSE DATABASES
CLOSE TABLES && referme les tables de log et de session (wConnect les laisse
IF Program(-1) <= 2 && lancé directement ou par xxxTest|Prod.prg
CLEAR ALL
RETURN
ENDIF
ENDIF
RETURN m.llResult

* =====
PROCEDURE ShutIt && Allows system shutdown
RELEASE goWCserver
```



FAS timer fires

In dev. mode, FAS waits for files written by the Web server, containing HTTP request details

```
Visual FoxPro Debugger [break] - wvisual.vcx\tutoverstart.tmfiletimer.timer
File Edit Debug Tools Window Help

Call Stack
+ 3 tutoverstart.tmfiletimer.timer
  2 awserverstart
  1 tutotest.prg

Trace
Object: tmfiletimer Procedure: Timer

*** called with the contents of the message file as a
*** parameter.
*****
LOCAL lcCommandLine, loServer

loServer = THIS.Parent

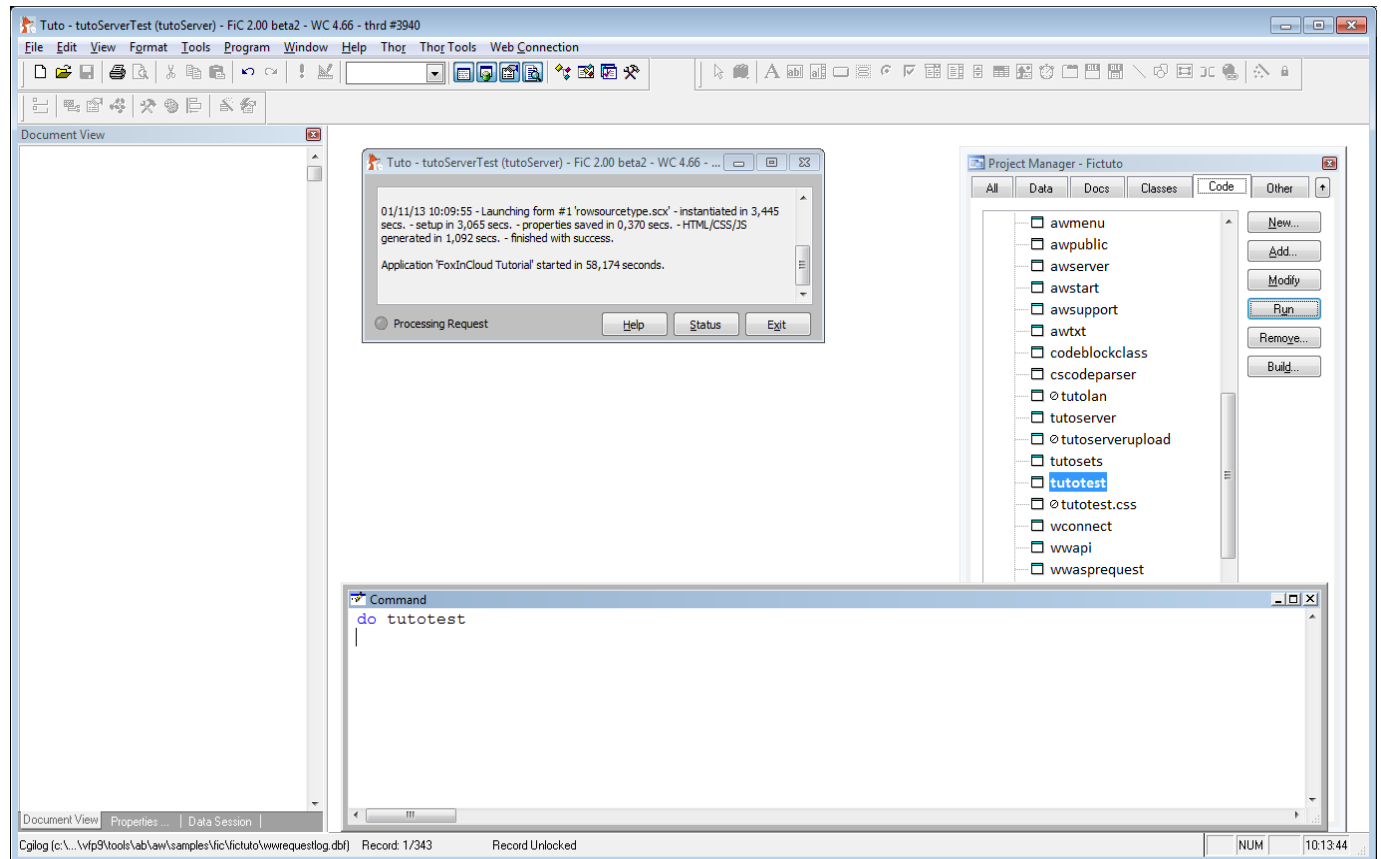
#IF SHAREWARE
  IF SECONDS()-loServer.nSWSeconds>SWTIMEOUT
    THIS.ENABLED=.F.    && Disable Timer
    DO FORM Sharewar
    loServer.nSWSeconds=SECONDS()
    THIS.ENABLED=.T.    && Turn back on
  ENDIF
#ENDIF

*** Checks for CGI Request TMP file
*** If found it's stuffed into Queue
lcCommandLine= THIS.CheckForMessageFile()

IF !EMPTY(lcCommandLine)
  THIS.ENABLED = .F.
  THIS.INTERVAL = 0
  loServer.ProcessHit(@lcCommandLine)
  THIS.INTERVAL = 5  && Force First request immediately
  THIS.ENABLED = .T.
ELSE
  *** Reset timer interval to normal
  IF THIS.INTERVAL = 5
    THIS.INTERVAL = loServer.nTimerInterval
  ENDIF
ENDIF
ENDIF
```



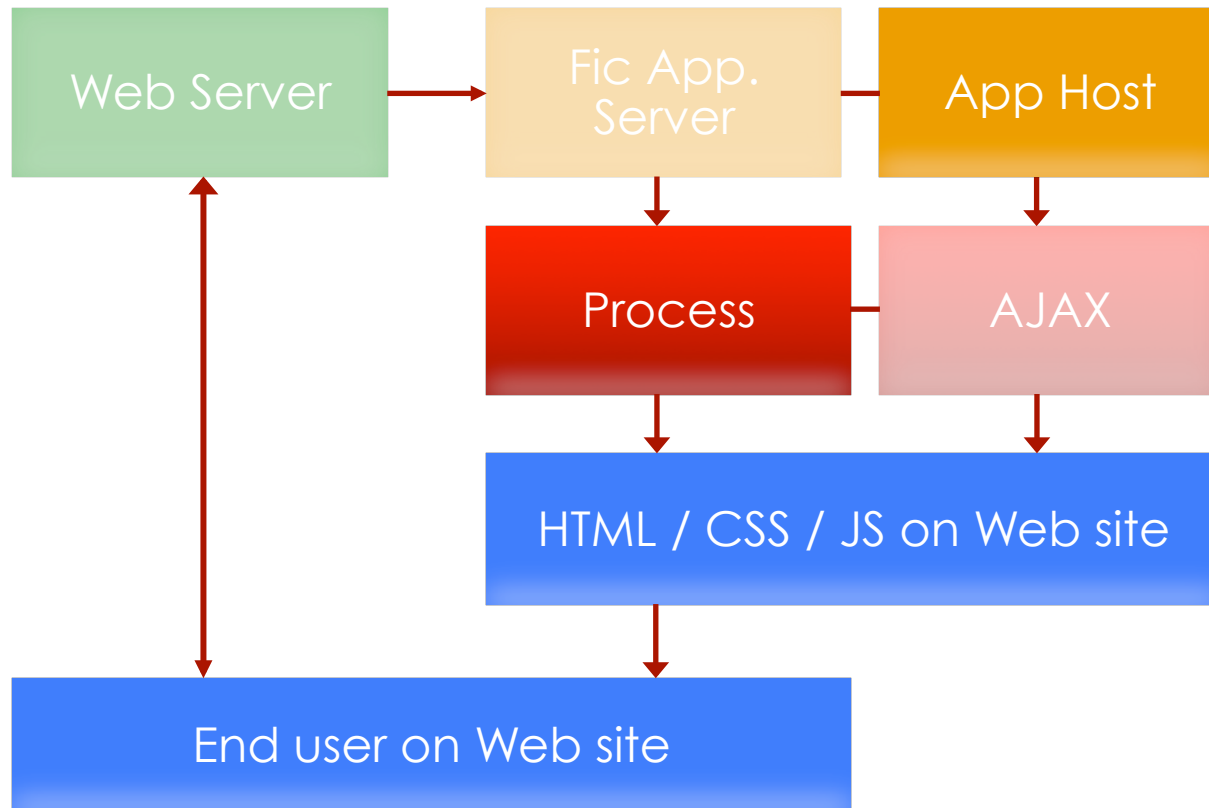
FAS: ready to serve requests!



Any question? Post with screenshot in the 'FoxInCloud' section of <http://west-wind.com/wwThreads/> or <http://www.universalthread.com/>

Session 1, episode 2

Displaying main form in browser





The form we want to display in the browser

`index.scx` is a regular VFP form with several command buttons and a background picture ...

The screenshot displays the FoxInCloud Web Application Studio (Version 2.00 beta2) interface. The main window, titled 'Form Designer - index.scx', shows a form with a light blue background. On the left is a cartoon fox illustration. The form contains the following text and buttons:

- Header: FoxInCloud Live Tutorial
- Buttons: User Events, .KeyPress(), Grid_RowColChange, RowSourceType, Modal Response, Using data, Event Log, paged Grid, EXIT
- Text: FOXinCloud, FoxPro app in the Cloud, www.foxincloud.com

The 'Properties - index.scx' window on the left shows the following properties for the form:

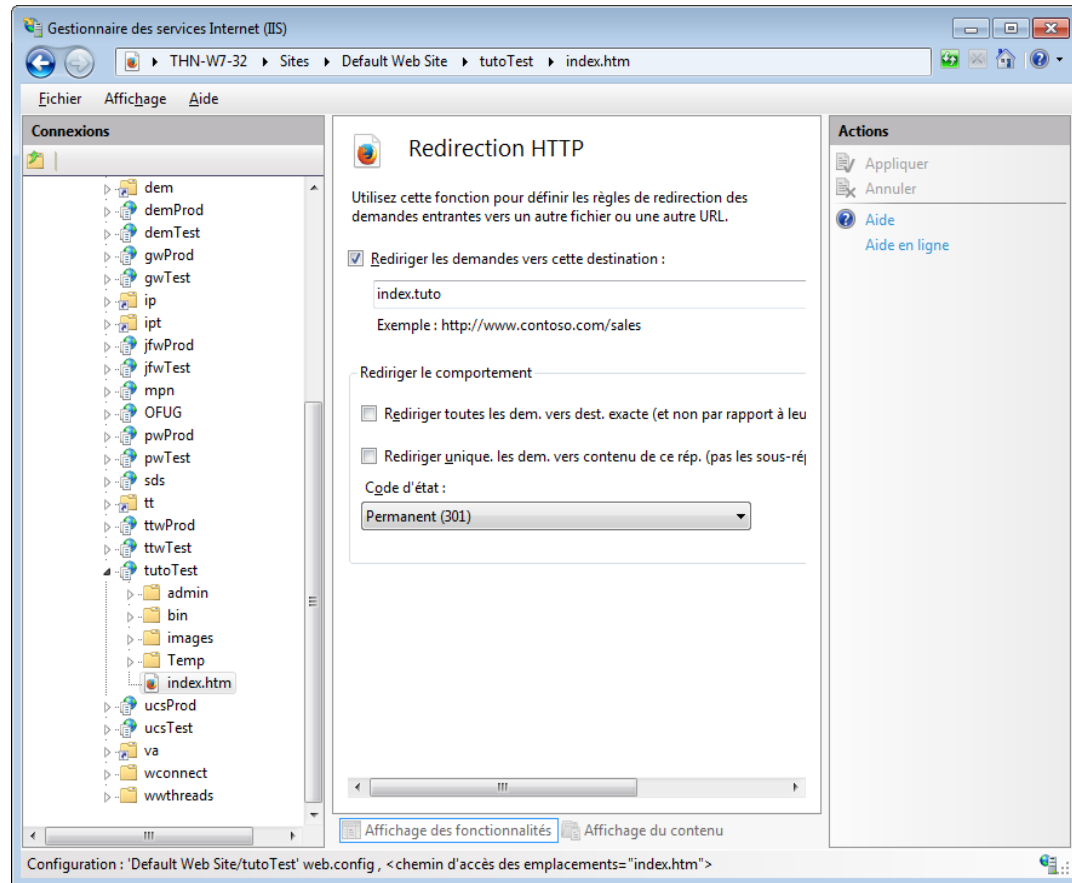
Property	Value
AutoCenter	.T. - True
BorderStyle	0 - No border
Caption	Tuto FoxInCloud
DataSession	2 - Private Data Session
Destroy	[User Procedure]
Height	338
MaxButton	.F. - False
MinButton	.F. - False
Movable	.F. - False
Name	Index
Picture	c:\Program files\vfp9\to
ShowTips	.T. - True
TitleBar	0 - Off
Width	800
WindowState	.F.

At the bottom of the properties window, there is a note: 'Specifies if the Form object is automatically centered in the main Visual FoxPro window the first time it is displayed.'



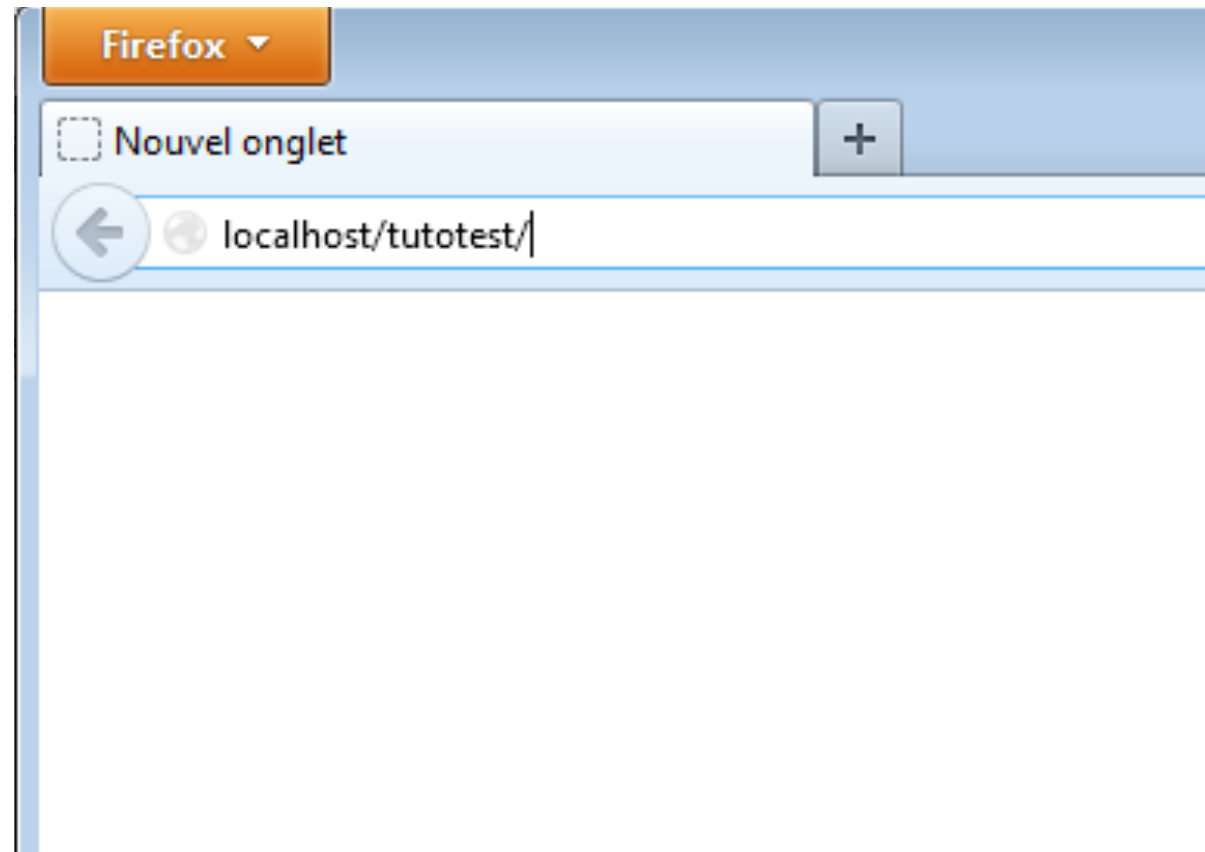
Redirecting IIS default document to FAS home

We redirect IIS's default document `index.htm` to an URL with the extension mapped to our FAS: `index.tuto`





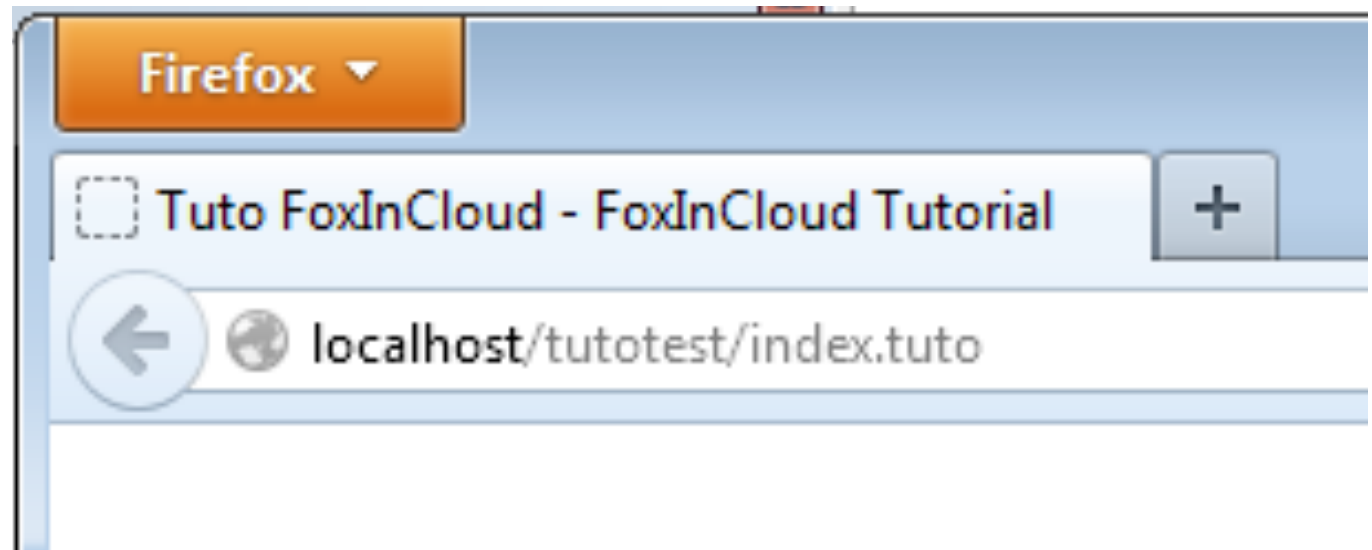
Typing application URL





IIS redirects root URL to our FAS

Due to IIS
redirection
of
`index.htm`
to
`index.tuto`,
typing
application
root URL
`http://
localhost/
tutotest/
yields
application
home page`





wc.dll gets the request from IIS and dumps it into a file

As we mapped *.tuto to wc.dll, IIS transmits the .../index.tuto request to wc.dll

```
Temp.ini
1 |&&Output+File=c:\temp\fictuto\WCDBA3.tmp&DLLVersion=Web+Connection+4%2E67+%2832+servers%29&wcCo
nfig=C%3A%5CProgram+Files%5CVFP9%5CTools%5CAB%5CAW%5CSamples%5CFIC%5CFicTuto%5Csite%5Cbin%5Cwc%2
EINI&REQUEST_METHOD=POST&QUERY_STRING=%26MethAddr%3DInitControlsInet%26ObjAddr%3Dindex%5Fscx%26P
ageInit%3Dtrue%26UserID%3D%26nReq%3D1&Executable+Path=%2Ftutotest%2FMethExec%2Etuto&CONTENT_TYPE
=application%2F%2Dwww%2Dform%2Durlencoded%3B+charset%3DUTF%2D8&CONTENT_LENGTH=19&PATH_INFO=%2Ft
utotest%2FMethExec%2Etuto&PATH_TRANSLATED=C%3A%5CProgram+Files%5CVFP9%5CTools%5CAB%5CAW%5CSamp
les%5CFIC%5CFicTuto%5Csite%5CMethExec%2Etuto&SCRIPT_NAME=%2Ftutotest%2FMethExec%2Etuto&PHYSICAL_PA
TH=C%3A%5CProgram+Files%5CVFP9%5CTools%5CAB%5CAW%5CSamples%5CFIC%5CFicTuto%5Csite%5CMethExec%2E
tuto&SERVER_PROTOCOL=HTTP%2F1%2E1&SERVER_SOFTWARE=Microsoft%2DIIS%2F7%2E5&SERVER_NAME=localhost&S
ERVER_PORT=80&REMOTE_HOST=%3A%3A1&REMOTE_ADDR=%3A%3A1&HTTP_REFERER=http%3A%2F%2Flocalhost%2Ftuto
test%2Findex%2Etuto&HTTP_USER_AGENT=Mozilla%2F5%2E0+%28Windows+NT%2E6%2E1%3B+rv%3A24%2E0%29+Gecko%
2F20100101+Firefox%2F24%2E0&HTTP_COOKIE=%5F%5Futma%3D111872281%2E777200636%2E1310508010%2E137783
1031%2E1377947599%2E174%3B+%5F%5Futmz%3D111872281%2E1310508010%2E1%2E1%2Eutmsr%3D%28direct%29%7
Cutmccn%3D%28direct%29%7Cutmcmd%3D%28none%29%3B+ip%3D3970QTW9T%3B+pus%2Didv%3DA0039d7dea3096fc47
1da81b55cbb68e9d69%3B+mpn%3D3NU0TNRAV%3B+tt%3D37F18TQ3Y%3B+ebNewBandWidth%5F%2Elocalhost%3D2247%
253A1365827628263%3B+tuto%3D3VT1AYS89%3B+%5F%5Futm%3D111872281%3B+ASPSESSIONIDSSQRBBQ%3DPFBBJJ
EBBAFHNGOCGKAOEOC&APPL_PHYSICAL_PATH=C%3A%5CProgram+Files%5CVFP9%5CTools%5CAB%5CAW%5CSamples%5C
FIC%5CFicTuto%5Csite%5C&GMT_OFFSET=3600&ALL_HTTP=HTTP_CACHE_CONTROL:no-cache
2 HTTP_CONNECTION:keep-alive
3 HTTP_PRAGMA:no-cache
4 HTTP_CONTENT_LENGTH:19
5 HTTP_CONTENT_TYPE:application/x-www-form-urlencoded; charset=UTF-8
6 HTTP_ACCEPT:text/javascript, text/html, application/xml, text/xml, */*
7 HTTP_ACCEPT_ENCODING:gzip, deflate
8 HTTP_ACCEPT_LANGUAGE:fr, fr-fr;q=0.8, en-us;q=0.5, en;q=0.3
9 HTTP_COOKIE: __utma=111872281.777200636.1310508010.1377831031.1377947599.174; __utmz=111872281.
1310508010.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); ip=3970QTW9T; pus-
idv=A0039d7dea3096fc471da81b55cbb68e9d69; mpn=3NU0TNRAV; tt=37F18TQ3Y; ebNewBandWidth_
localhost=2247%3A1365827628263; tuto=3VT1AYS89; __utmc=111872281;
ASPSESSIONIDSSQRBBQ=PFBBJJEBBAFHNGOCGKAOEOC
10 HTTP_HOST:localhost
11 HTTP_REFERER:http://localhost/tutotest/index.tuto
12 HTTP_USER_AGENT:Mozilla/5.0 (Windows NT 6.1; rv:34.0) Gecko/20100101 Firefox/34.0
```




FAS routes request to application's Process object

Inside FAS, the `.tuto` extension maps to the `tutoProcess` class.

This class processes the request.

```
Visual FoxPro Debugger [break] - awserver.prg:tutoservertest.routerequest
File Edit Debug Tools Window Help
Call Stack
6 tutoProcess.routerequest
5 tutoProcess.process
4 tutoProcess.processhit
3 tutoProcess.tmrfiletimer.timer
2 awserverstart
1 tutotest.prg
Trace
Object: tutoProcess Procedure: routerequest
707 DO CASE
708
709 CASE InList(m.tcParameter, 'FOXINCLOUD', 'FOXLANWEB', 'ZENBUYER', 'INTUICA
710 OR Ascan(m.this.aScriptMapExt, m.tcExtension, 1, -1, 1, 7) > 0 && Script
711
712 loProcess = CreateObject(m.this.cawProcessClass, m.this)
713 RETURN loProcess.Process()
714
715 CASE InList(m.tcParameter, 'WWMAINT', 'AWMAINT', 'AZMAINT')
716 loProcess = CreateObject('awMaint', m.this)
717 RETURN loProcess.Process()
718
719 CASE InList(m.tcExtension, 'WC', 'WCS', 'FXP')
720 RETURN wwScriptMaps(m.this)
721
722 CASE m.tcExtension == 'WWWSCAN'
```

Name	Value	Type
m.tcExtension	"TUTO"	C
m.ascriptmapext	(Array)	A
ascriptmapext[1]	"php"	C
ascriptmapext[2]	"pl"	C
ascriptmapext[3]	"vfp"	C
ascriptmapext[4]	"fic"	C
ascriptmapext[5]	"zip"	C
ascriptmapext[6]	"tuto"	C
m.this.cawProcessClass	"tutoProcess"	C

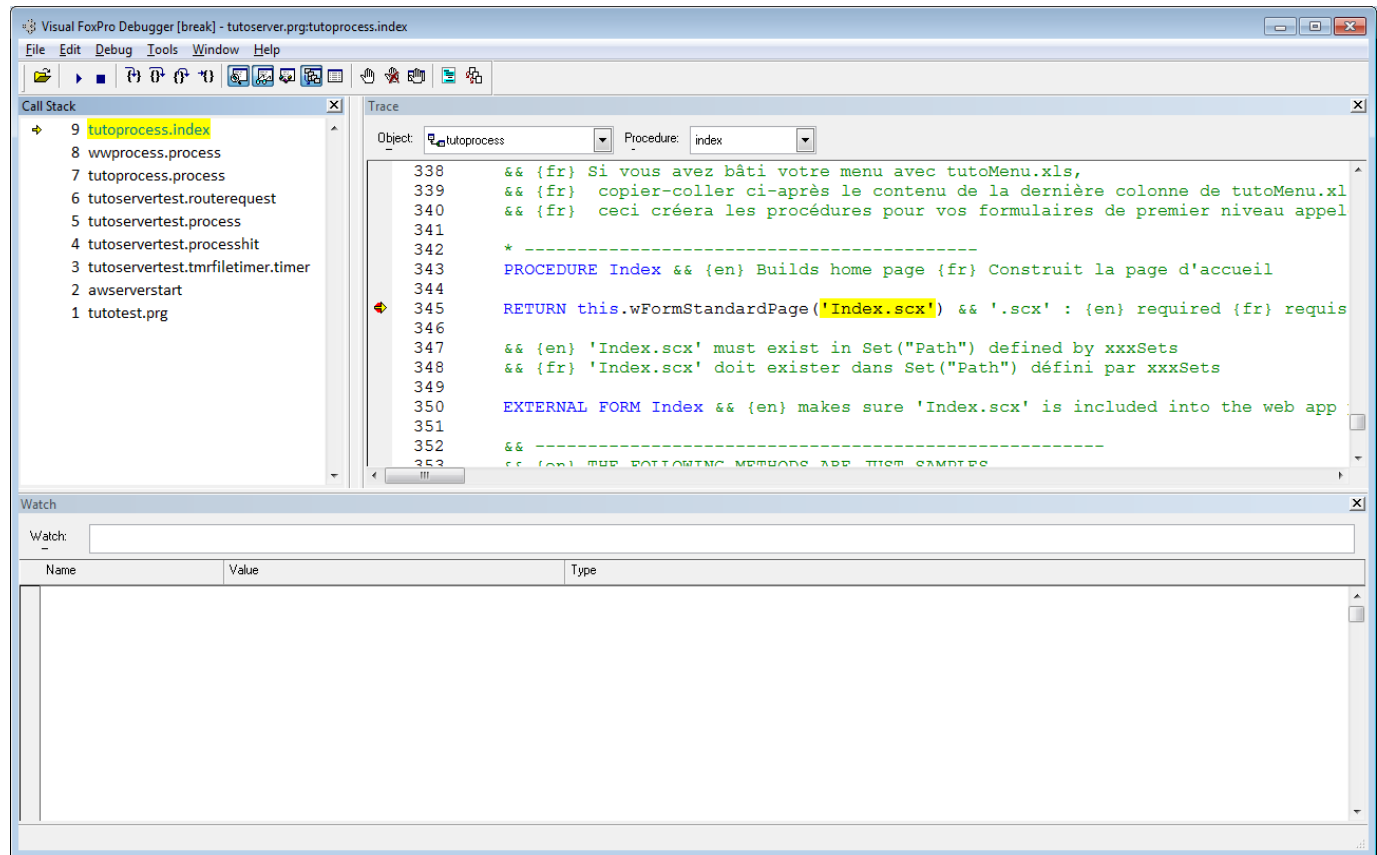


Application executes process class method

The process class executes the method named after justStem(url)

("index.tuto" executes the method 'index' of process class mapped to '.tuto')

All FoxInCloud member names begin with 'w'





Process asks application host for a reference to the form

oAppHost is FoxInCloud application host object.

Application host object keeps a pool of references to the various forms in the application

```
Visual FoxPro Debugger [break] - awserver.prg:tutoprocess.wform
File Edit Debug Tools Window Help
Call Stack
  11 tutoprocess.wform
  10 tutoprocess.wformstandardpage
  9 tutoprocess.index
  8 wwprocess.process
  7 tutoprocess.process
  6 tutoservertest.routerequest
  5 tutoservertest.process
  4 tutoservertest.processhit
  3 tutoservertest.tmrfiletimer.timer
  2 awserverstart
  1 tutotest.prg
Trace
  Object: tutoprocess Procedure: wform
  2714      )))
  2715      IF m.llResult
  2716
  2717          lcParms = wcFormParms(m.lnParm)
  2718          * =====
  2719          llResult = m.this.oAJAX.FormInitSave(m.tcForm &lcParms)
  2720          * =====
  2721          this.cResult = Iif(m.llResult, Space(0), m.this.oAJAX.cResult)
  2722          IF m.llResult
  2723
  2724              loResult = this.oAppHost.oFormOfName(m.tcForm)
  2725              llResult = Vartype(m.loResult) == 'O!'
  2726              this.cResult = Iif(m.llResult, Space(0), m.this.oAppHost.cResult)
  2727          ENDIF
  2728      ENDIF
  2729  ENDIF
  2730
  2731  = m.llResult OR this.Warning(Textmerge( [ <<Proper(Program())>>() ] - <<m.this.cRe
  2732
  2733  RETURN m.loResult
  2734
  2735  * -----
Watch
  Watch:
  Name      Value      Type
  -----
  m.tcform  "Index.scx"  C
```



First time this form is used: App Host must instantiate

App. Host
does not
hold any
reference to
the form
requested:

This form
needs to be
instantiated

Visual FoxPro Debugger [break] - awapphost.prg;awapphost.oform

File Edit Debug Tools Window Help

Call Stack

- 13 awapphost.oform
- 12 awajax.forminitsave
- 11 tutoprocess.wform
- 10 tutoprocess.wformstandardpage
- 9 tutoprocess.index
- 8 wwprocess.process
- 7 tutoprocess.process
- 6 tutoservertest.routerequest
- 5 tutoservertest.process
- 4 tutoservertest.processhit
- 3 tutoservertest.tmrfiletimer.timer
- 2 awserverstart
- 1 tutotest.prg

Trace

Object: awapphost Procedure: oform

```
1565 lnDSID_ = Set("DataSession")
1566 SET DATASESSION TO m.lnDSID
1567 ENDIF
1568
1569 * Instancier l'écran
1570 loResult = m.this.FormLaunch(;&& règle l'utilisateur et son n° de requête &
1571 @m.tuForm;
1572 , m.llSubForm;
1573 , m.lcSuffix;
1574 )
1575
1576 * Rétablir la session de données
1577 IF NOT Empty(m.lnDSID_)
1578 SET DATASESSION TO m.lnDSID_
1579 ENDIF
1580
1581 llResult = Vartype(m.loResult) == 'O';
1582 AND this.oForm_Props(m.loResult, m.llSubForm, m.tlModal, m.lnDSID, m.lcFor
1583
1584 ENDIF
1585 IF m.llResult
1586
```

Watch

Watch: |

Name	Value	Type
m.tuform	"Index.scx"	C



App Host stores a reference to the form

App. Host instantiates the form using a regular `DO FORM...NAME` command.

This form instance lives inside App Host as long as server runs.

All users share this instance.

The screenshot shows the Visual FoxPro Debugger interface. The Call Stack window on the left lists the following procedure calls from top to bottom:

- 15 awapphost.formlaunch_
- 14 awapphost.formlaunch
- 13 awapphost.oform
- 12 awajax.forminitsave
- 11 tutoprocess.wform
- 10 tutoprocess.wformstandardpage
- 9 tutoprocess.index
- 8 wwprocess.process
- 7 tutoprocess.process
- 6 tutoservtest.routerequest
- 5 tutoservtest.process
- 4 tutoservtest.processshit
- 3 tutoservtest.tmrfiletimer.timer
- 2 awserverstart
- 1 tutotest.prg

The Trace window on the right shows the following code snippet:

```
2004                                     [Form File '<<m.tcForm>>' can't be found]; && d ^
2005                                     )))
2006                                     = m.llResult OR this.Warning(m.this.cResult)
2007                                     IF m.llResult
2008                                     * -----
2009                                     *
2010                                     DO FORM (m.tcForm) NAME loResult LINKED NOSHOW
2011                                     * -----
2012                                     *
2013                                     ENDIF
2014                                     ELSE
2015                                     * -----
2016                                     *
2017                                     llResult = .T.
2018                                     liClass = 0 && this.oClass.getKey(m.tcForm)
2019                                     * -----
2020                                     *
2021                                     loResult = Iif(m.liClass > 0;
2022                                     , this.oClass(m.liClass);
2023                                     , CreateObject(m.tcForm);
2024                                     )
2025                                     * -----
2026                                     *
2027                                     ENDIF
```

The Watch window at the bottom shows the following variable:

Name	Value	Type
m.tcForm	"index.scx"	C



Form.events() execute

Form
instantiates
and fires
events just
like in
desktop:

.Load(),
.Init(),
.Show(), ...

```
Visual FoxPro Debugger [break] - ficsample.vcxindex.load
File Edit Debug Tools Window Help

Call Stack
16 index.load
15 awapphost.formlaunch_
14 awapphost.formlaunch
13 awapphost.oform
12 awajax.forminitsave
11 tutoprocess.wform
10 tutoprocess.wformstandardpage
9 tutoprocess.index
8 wwprocess.process
7 tutoprocess.process
6 tutoservertest.routerequest
5 tutoservertest.process
4 tutoservertest.processhit
3 tutoservertest.tmrfiletimer.timer
2 awserverstart
1 tutotest.prg

Trace
Object: Index Procedure: Load
1 LOCAL cSetExclu AS STRING,
2   cSetSafe AS STRING
3
4 cSetExclu = SET("Exclusive")
5 cSetSafe = SET("Safety")
6 SET EXCLUSIVE OFF
7 SET SAFETY OFF
8
9 IF !FILE('ficevent.dbf')
10  CREATE TABLE ficevent FREE (idpk I AUTOINC, who C(1), FORM C(50), NAME C(50),;
11    EVENT C(50), BASECLASS C(50), CLASS C(50), TIMESTAMP C(16), inactif L)
12  INDEX ON idpk TAG idpk CANDIDATE
13  INDEX ON TIMESTAMP TAG TIMESTAMP ADDITIVE
14 ELSE
15  USE ficevent SHARED AGAIN IN 0
16 ENDIF
17
18 IF m.cSetExclu == "ON"
19  SET EXCLUSIVE ON
20 ELSE
21  SET EXCLUSIVE OFF
22 ENDIF
23
24 IF m.cSetSafe == "ON"
25  SET SAFETY ON
26 ELSE
27  SET SAFETY OFF
28 ENDIF
29
30 RETURN DODEFAULT()
```



FAS saves form's initial state into a VFP table

As any Web application, FoxInCloud needs save application and forms state for each user, including dataSession;

Initial state is restored whenever a user opens a form.

The screenshot displays the FoxInCloud Web Application Studio interface. The main window shows a table with columns Oa, Bc, Pn, Pt, Pv, Pa, Pr, Pp, and C. The table lists various form controls and their initial states. The 'DataSession' row is highlighted in blue.

Oa	Bc	Pn	Pt	Pv	Pa	Pr	Pp	C
tutocmdrowcolchange	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
tutolanrelease	Commandbutton	ENABLED	L	Memo	T	T	F	F
tutolanrelease	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
tutocmdlogevent	Commandbutton	ENABLED	L	Memo	T	T	F	F
tutocmdlogevent	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
tutocmddata	Commandbutton	ENABLED	L	Memo	T	T	F	F
tutocmddata	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
tutorowsourcetype	Commandbutton	ENABLED	L	Memo	T	T	F	F
tutorowsourcetype	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
tutocmdpagedgrid	Commandbutton	ENABLED	L	Memo	T	T	F	F
tutocmdpagedgrid	Commandbutton	MOUSEPOINTER	N	Memo	T	T	F	F
	Form	WUPK	O	Memo	F	F	F	F
	Form	OBJECTEVENT	C	Memo	F	T	F	F
	Form	OBJECTBINDED	C	Memo	F	T	F	F
	Form	ICOUNTEREVENT	N	Memo	F	T	F	F
	Form	CAPTION	C	Memo	T	T	F	F
	Form	WCDSFORM	C	memo	F	T	F	F
	Form	WNDATASESSIONID	N	Memo	F	T	F	F
	Form	WLMODAL	L	Memo	F	T	F	F
	Form	WLSUBFORM	L	Memo	F	T	F	F
	Form	WUVALUE	X	Memo	T	T	F	F
	Form	DataSession	V	Memo	F	F	F	F

The right pane shows the VFP table structure for 'Index_scx_ini.pv':

```
<DataSession>
  <aliases>
    <awbcevent>
      <dbf>C:\PROGRAM FILES\VFP9\TOOLS\AB\AW\AWBCEVENT.DBF</dbf>
      <cols></cols>
      <recno>1</recno>
      <relations></relations>
      <data>
        <buffering>1</buffering>
        <buffer>
          <recnos></recnos>
          <deleted></deleted>
          <fldStates></fldStates>
          <rs></rs>
        </buffer>
        <rows></rows>
      </data>
      <tags></tags>
      <order></order>
      <key></key>
      <setFields></setFields>
      <filter></filter>
      <savems>0</savems>
    </awbcevent>
    <ficevent>
      <dbf>C:\PROGRAM FILES\VFP9\TOOLS\AB\AW\SAMPLES\FIC\FICTUTO\FICEVENT.DBF</dbf>
      <cols></cols>
      <recno>1</recno>
```



FAS clones form into HTML, CSS and JavaScript

FoxInCloud's `awHTMLgen` class walks through the form's containers hierarchy to generate HTML CSS JS replicating the layout and behavior of original VFP form

The screenshot shows the Visual FoxPro Debugger interface. The Call Stack window on the left lists the following frames:

- 16 awapphost.cformhtml
- 15 awapphost.formlaunch_
- 14 awapphost.formlaunch
- 13 awapphost.oform
- 12 awajax.forminitsave
- 11 tutoprocess.wform
- 10 tutoprocess.wformstandardpage
- 9 tutoprocess.index
- 8 wwprocess.process
- 7 tutoprocess.process
- 6 tutoservertest.routerequest
- 5 tutoservertest.process
- 4 tutoservertest.processshit
- 3 tutoservertest.tmfiletimer.timer
- 2 awserversstart
- 1 tutotest.prg

The Trace window shows the following code for the `cformhtml` procedure:

```
1352 *-----  
1353  
1354 FUNCTION cFormHTML && Code HTML d'un écran de l'application  
1355 LPARAMETERS ;  
1356 toForm AS awFrm OF aw.vcx,; && Écran à rendre  
1357 tlFormMain,; && [.F.] Écran top level, situé sur une page du site (sinon, écran secondaire superposé  
1358 tcScript,; && [''] @ Script de l'écran  
1359 tlStyleInLine && [.F.] Écrire les styles dans la balise des éléments (au lieu de la feuille de styles  
1360  
1361 LOCAL lcResult  
1362 STORE Space(0) TO tcScript, lcResult  
1363  
1364 IF loClass(m.toForm, 'form')  
1365  
1366 WITH m.this.oHTMLgen as awHTMLgen OF awHTML.prg  
1367 .lFormMain = Vartype(m.tlFormMain) == 'L' AND m.tlFormMain && avant d'affecter .oControl !  
1368 .lStyleInLine = Vartype(m.tlStyleInLine) == 'L' AND m.tlStyleInLine && avant d'affecter .oControl !  
1369 .oControl = m.toForm && cf. oControl_Assign()  
1370 lcResult = .getHTML()  
1371 tcScript = .cScriptJS  
1372 ENDWITH  
1373 ENDIF  
1374  
1375 RETURN m.lcResult  
1376  
1377 *-----
```

The Debug Output window shows the following text:

```
m.lcResult = <!-- Généré le 01/11/13 14:35:06 par FoxInCloud version  
2.00 beta2 -->  
<div id="index_scx" class="form awfrm ficfrm" tabindex="1">  
  
<button id="index_scx_tutocmdventabiest" class="commandbutton
```



FAS send back a full HTML page including the form

FoxInCloud's `.wFormStandardPage()` method builds a full HTML page including `<head>`, FAS standard and application-specific CSS & JS, `<body>`, etc.

```
Visual FoxPro Debugger [break] - awserver.prg:tutoprocess.wformstandardpage_output
File Edit Debug Tools Window Help
Trace
Object: tutoprocess Procedure: wformstandardpage
2665 </body>
2666 </html>
2667 ENDTTEXT
2668
2669 * Send HTML response back
2670 Response.ContentTypeHeader ( !text/html; charset=utf-8 ! )
2671 Response.FastWrite ( cUTF8 ( m.m.lcResult ) )
2672
2673 * -----
2674 PROTECTED FUNCTION wFormStandardPage_cKeyWords && <meta name="keywords" content="" /> {en} keywords for this meta tag {fr} Mots clés pour cet
2675
2676 RETURN lcCase !

Debug Output
m.lcResult = <!doctype html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" charset="utf-8">
<head>
<meta charset="utf-8"> <!-- http://www.w3.org/International/questions/qa-html-encoding-declarations
http://www.iana.org/assignments/character-sets -->
<meta name="description" content="FoxInCloud Tutorial" />
<meta name="keywords" content="FoxInCloud, SaaS, Cloud, FoxPro, Web, adapter une application client lourd au web" />
<meta name="author" content="FoxInCloud App Admin" />
<meta name="generator" content="FoxInCloud Application Server Version 2.00 beta2" />
<!--[if IE]>
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<![endif]--> <!-- IE: force rendering mode to the latest available standard -->

<title>Tuto FoxInCloud - FoxInCloud Tutorial</title>
<link type="text/css" rel="stylesheet" href="/awScripts/Window1.3/Themes/default.css" />
<link type="text/css" rel="stylesheet" href="/awScripts/Window1.3/Themes/all.css" />
<link type="text/css" rel="stylesheet" href="/awScripts/Protofish/Protofish.css" />
<link type="text/css" rel="stylesheet" href="/awScripts/AW2.6.1/styles/system/aw.css" />
<link type="text/css" rel="stylesheet" href="/awScripts/iOuvrv-ui-1.10.2/themes/base/inquerv-ui.css" />
```



Form page finally shows up in the browser ...

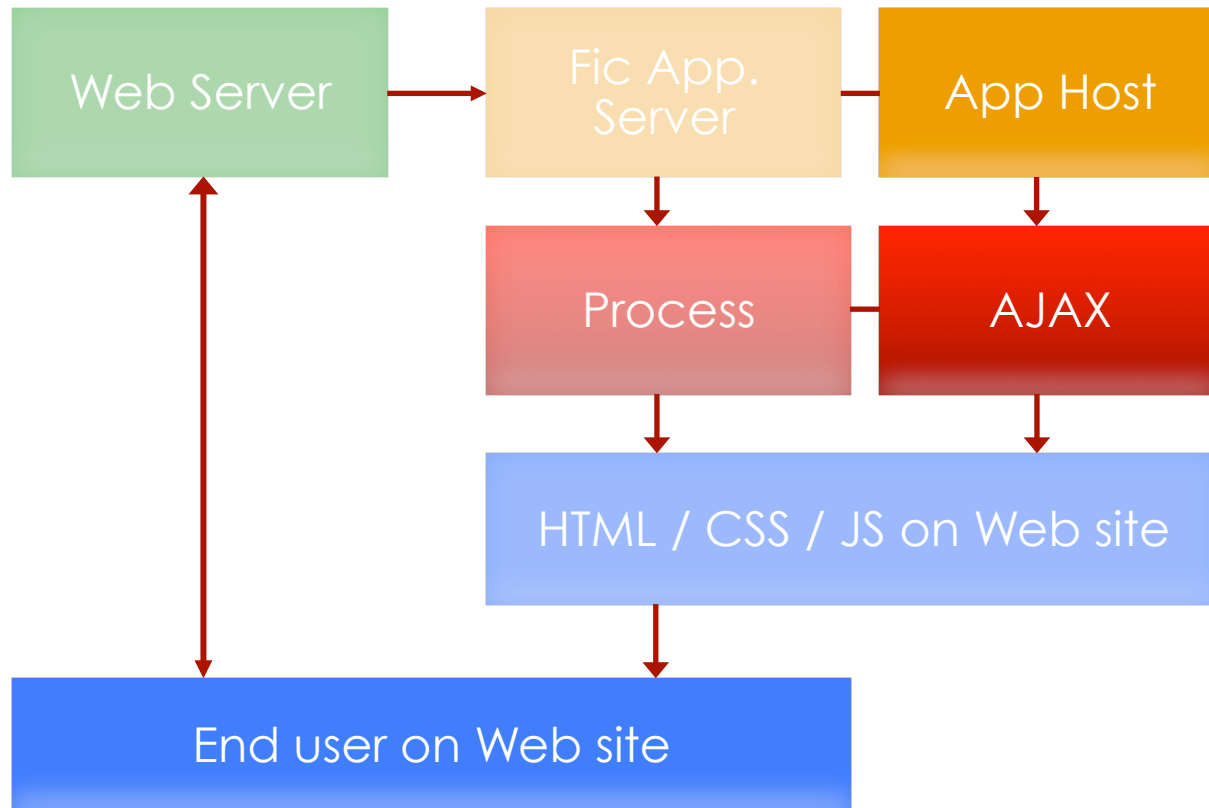
The whole process, from FAS receiving the request to response delivered takes less than a second ...

```
index.tuto -  
Launching form #1  
'index.scx'...  
Instantiated in  
.124 secs.  
Setup in  
.006 secs.  
State saved in  
.134 secs.  
HTML/CSS/JS  
generated in  
.334 secs.  
Finished with  
success in  
.914 secs
```



Session 1, episode 3

Interacting with a form





Desktop: clicking a form button opens a child form...

In the desktop application, clicking the `.rowSourceType` button opens the `rowSourceType.scx` form

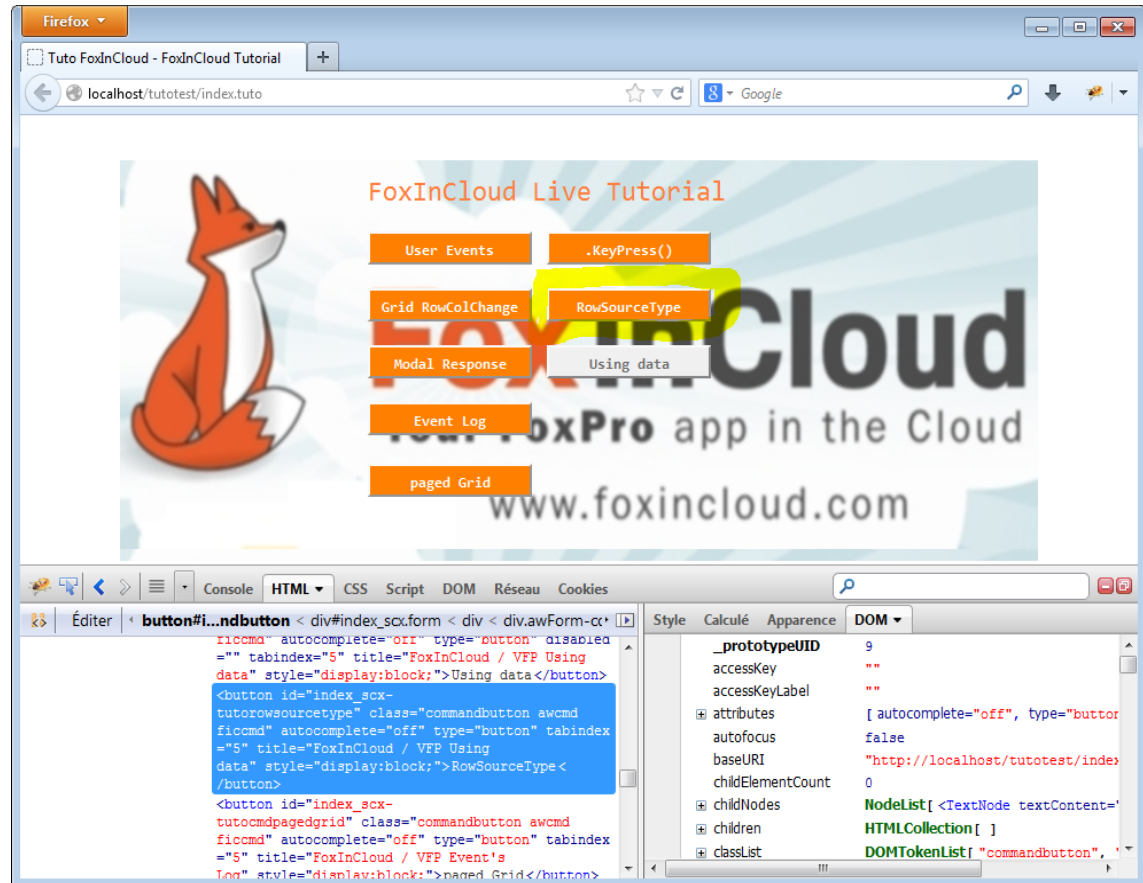
The screenshot shows the FoxInCloud Web Application Studio interface. The main window displays a desktop application with a fox logo and several buttons: 'User Events', '.KeyPress()', 'Grid RowColChange', 'RowSourceType', 'Modal Response', and 'Using data'. A child form titled 'RowSourceType processing by FoxInCloud' is open, showing a list of data sources for various RowSourceType values. The list includes 'Aux Joyeux ecclési', 'Bigfoot Breweries', 'Cooperativa de Que', 'Escargots Nouveaux', 'Exotic Liquids', 'Formaggi Fortini s', 'Forêts d'érables', 'G'day, Mate', 'Gai pâturage', 'Grandma Kelly's Ho', 'Hell Süßwaren GmbH', 'Karkki Oy', 'Leka Trading', 'Lyngbysild', and 'Ma Maison'.

RowSourceType = 4 && QPR (Parent List)	RowSourceType = 1 && Value	RowSourceType = 2 && Alias	RowSourceType = 3 && SQL Query	RowSourceType = 5 && Array
Aux Joyeux ecclési				
Bigfoot Breweries				
Cooperativa de Que				
Escargots Nouveaux				
Exotic Liquids				
Formaggi Fortini s				
Forêts d'érables				
G'day, Mate				
Gai pâturage				
Grandma Kelly's Ho				
Hell Süßwaren GmbH				
Karkki Oy				
Leka Trading				
Lyngbysild				
Ma Maison				



Let's click the same button in web browser ...

FAS has generated the same button in HTML



Any question? Post with screenshot in the 'FoxInCloud' section of <http://west-wind.com/wwThreads/> or <http://www.universalthread.com/>



FoxInCloud has generated a JS `.click()` event handler

Because the `.click()` event of this button is implemented in the original VFP form, FAS has generated a similar event handler in form's JavaScript named `index_scx.js`

The screenshot shows a web browser window at `localhost/tutotest/index.tuto`. The page content includes a fox illustration, the title "FoxInCloud Live Tutorial", and several orange buttons: "User Events", ".KeyPress()", "Grid RowColChange", "RowSourceType", "Modal Response", "Using data", "Event Log", and "paged Grid". The URL `www.foxincloud.com` is visible at the bottom of the page content.

The browser's developer console is open, showing the JavaScript code for `index_scx.js`. The code consists of 12 lines, each binding a click event to the `FoxInCloud.DOMEvent.bind(FoxInCloud)` function. Line 11 is highlighted in blue:

```
4 $('index_scx-tutocmdmodal').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
5 $('index_scx-tutocmdrowcolchange').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
6 $('index_scx-tutolanrelease').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
7 $('index_scx-tutolanrelease').disabled=true;
8 $('index_scx-tutocmdlogevent').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
9 $('index_scx-tutocmddata').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
10 $('index_scx-tutocmddata').disabled=true;
11 $('index_scx-tutorowsourcetype').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
12 $('index_scx-tutocmbaagedarid').observe('click', FoxInCloud.DOMEvent.bind(FoxInCloud));
```



Clicking this button triggers an AJAX request to FAS

All user interaction events are processed by standard `FoxInCloud.js.DOMEvent` method, resulting in sending an AJAX request to the FAS server; URL details the event occurred

The screenshot displays a web browser window at `localhost/tutotest/index.tuto`. The page content includes a cartoon fox on the left and a central area with the text "FoxInCloud Live Tutorial" and "www.foxincloud.com". Several buttons are visible: "User Events", ".KeyPress()", "Grid RowColChange", "RowSourceType", "Modal Response", "Using data", "Event Log", and "paged Grid". The browser's developer console is open, showing a `RequestSend` event in `FoxInCloud.js`. The console log shows the following code snippet:

```
621 new Ajax.Request(this.wcURL(url),
622 { asynchronous: !sync
623 , parameters: this.parameters
624 , onCreate: this.AjaxCreate.bind(this)
625 , onException: this.AjaxException.bind(this)
626 , onFailure: this.AjaxFailure.bind(this)
627 , onSuccess: this.AjaxSuccess.bind(this)
628 , onComplete: this.AjaxComplete.bind(this)
629 }
```

The console also shows the request details in the "Espions" pane:

```
Nouvelle expression espion...
this.wcURL(url) "DOMEvent.tuto?&Event=click&
ObjAddr=index_scx-
tutorowsourcetype&PageInit=&
UserID=3VT1AYS89&nReq=2"
this Object { version=2,
cEvent="click", dEvent=Date,
```



FAS executes button's .click() method

.DOMEvent ()
goes through several FAS
methods to
end up
executing the
application
button's
.Click ()
method.

.wForm ()
opens
RowSourceTy
pe.scx as a
child form

```
Visual FoxPro Debugger [break] - index.scxindex_scx.tutorowsourcetype.click
File Edit Debug Tools Window Help
Call Stack Trace
Object: tutorowsourcetype Procedure: Click
1 LPARAMETERS nButton, nShift, nXCoord, nYCoord
2 set step on
3 IF (Type('m.THISFORM.wlHTMLgen') == 'L' AND m.THISFORM.wlHTMLgen)
4 RETURN .T. && {fr} Traiter l'événement sur le serveur (bloc ELSE ci-après)
5 ELSE
6 =THISFORM.wForm("RowSourceType.scx")
7 && {fr} aspire 'URowSourceType.scx' dans le projet de l'application Web
8 && {en} makes sure 'RowSourceType.scx' is included into the web app project
9 EXTERNAL FORM RowSourceType
10 ENDIF
16 index_scx.tutorowsourcetype.click
15 index_scx.tutorowsourcetype.click
14 awapphost.useraction_
13 awapphost.useraction
12 awajax.useraction
11 awajax.formrequest
10 tutoprocess.ajaxformrequest
9 tutoprocess.domevent
8 wwprocess.process
7 tutoprocess.process
6 tutoservertest.routerequest
5 tutoservertest.process
4 tutoservertest.processhit
3 tutoservertest.tmfiletimer.timer
2 awserverstart
1 tutotest.prg
```



wForm() asks Application Host for a form reference

Similar to what we saw in the previous episode, `.wForm()` asks application host for a reference to the form; if none exist, AppHost launches the form, saves its state and generate HTML/CSS/JS

```
Visual FoxPro Debugger [break] - aw.vocindex_scx.wformnshow
File Edit Debug Tools Window Help
Call Stack
18 index_scx.wformnshow
17 index_scx.wform
16 index_scx.tutorowsourcetype.click
15 index_scx.tutorowsourcetype.click
14 awapphost.useraction_
13 awapphost.useraction
12 awajax.useraction
11 awajax.formrequest
10 tutoprocess.ajaxformrequest
9 tutoprocess.domevent
8 wwprocess.process
7 tutoprocess.process
6 tutoservertest.routerequest
5 tutoservertest.process
4 tutoservertest.processhit
3 tutoservertest.tmrfiletimer.timer
2 awserverstart
1 tutotest.prg
Trace
Object: index_scx Procedure: wformnshow
55 )))
56 IF m.luResult
57
58     lcParms = wcFormParms(Pcount() - 2)
59     * =====
60     loForm = .oAppHost.oForm(@m.tuForm, m.this, m.llModal &lcP
61     * =====
62
63     luResult = ICASE(
64         .oAppHost.lFormAnonymousDenied;
65         , .BackHome() AND .F.;
66         , NOT .lSuccess(Vartype(m.loForm) == 'O', .oAppHost.cR
67         , .F.;
68         , m.loForm;
69     )
70 ENDIF
71 ENDWITH
72 ENDIF
73
74 ELSE && LAN MODE
75
76 *!* (V 1.25) Textile ici = nValue doit être visible quand this l'upload est
```

Name	Value	Type
m.tuform	"RowSourceType.scx"	C



.wForm() adds child form dialog into AJAX response

Knowing that form's HTML/CSS/JS is available, `.wForm()` asks the FAS AJAX response object to add the form's HTML/CSS/JS to the AJAX response.

The screenshot shows the Visual FoxPro Debugger interface. The Call Stack window on the left lists the following frames:

- 18 index_scx.wformshow
- 17 index_scx.wform
- 16 index_scx.tutorowsourcetype.click
- 15 index_scx.tutorowsourcetype.click
- 14 awapphost.useraction_
- 13 awapphost.useraction
- 12 awajax.useraction
- 11 awajax.formrequest
- 10 tutoprocess.ajaxformrequest
- 9 tutoprocess.domevent
- 8 wwprocess.process
- 7 tutoprocess.process
- 6 tutoservtest.routerequest
- 5 tutoservtest.process
- 4 tutoservtest.processhit
- 3 tutoservtest.tmrfiletimer.timer
- 2 awserverstart
- 1 tutotest.prg

The Trace window on the right shows the execution of the `wformshow` procedure. The current line of code is:

```
RETURN m.AJAX.DialogAdd(m.toForm, m.tuFormCallBack, m.loControl)
```

The Watch window at the bottom shows the following variables:

Name	Value	Type
mousepointer	0	N
movable	.T.	L
name	"rowsourcetype_scx"	C
objects	(Collection)	A
oledragmode	0	N



Form HTML/CSS/JS is encoded into the XML response

FAS response to AJAX requests is encoded as an XML stream.

FoxInCloud.js parses this XML response into JavaScript instructions to update HTML page.

Visual FoxPro Debugger [break] - awserver.prg:ajax.xmldialogadd

Call Stack

- 20 ajax.xmldialogadd
- 19 ajax.dialogadd
- 18 index_scx.wformshow
- 17 index_scx.wform
- 16 index_scx.tutorowsourcetype.click
- 15 index_scx.tutorowsourcetype.clicki
- 14 awapphost.useraction_
- 13 awapphost.useraction
- 12 ajax.useraction
- 11 ajax.formrequest
- 10 tutoprocess.ajaxformrequest
- 9 tutoprocess.domevent
- 8 wwprocess.process
- 7 tutoprocess.process
- 6 tutoservtest.routerrequest
- 5 tutoservtest.process
- 4 tutoservtest.processhit
- 3 tutoservtest.tmrfiletimer.timer
- 2 awserverstart
- 1 tutotest.prg

Trace

```
10165      + '!'
10166
10167      * Ajouter le HTML à la réponse XML, dans l'élément 'HTMLDialog' &&
10168      TEXT TO this.cXML ADDITIVE TEXTMERGE NOSHOW FLAGS 1
10169      <<CRLF + Space(0)>><HTMLDialog>
10170      <DivID><<m.toForm.Name>></DivID>
10171      <DivTitle><<Escaped_XML(m.lcTitle)>></DivTitle>
10172      <DivHTML>
10173      <<Escaped(m.lcHTML)>>
10174      </DivHTML>
10175      <DivWidth><<Int(m.toForm.Width)>>px</DivWidth>
10176      <DivHeight><<Int(m.toForm.Height)>>px</DivHeight>
10177      <DivModal><<LitteralJS(m.toForm.wlModal)>></DivModal>
10178      <ObjAddr><<Iif(Empty(m.tcMethod), Space(0), Strtran(JustStem(m.tcMethod),
10179      <MethAddr><<Iif(Empty(m.tcMethod), Space(0), JustExt(m.tcMethod))>></MethA
10180      <Theme><<m.toForm.wcWindowTheme>></Theme>
10181      <Center><<LitteralJS(m.toForm.AutoComplete)>></Center>
10182      <Buttons><<m.lcButtons>></Buttons>
10183      </HTMLDialog>
10184      ENDTXT
10185
10186      IF VFP.Startmode = 0 && IDE
```

Debug Output

```
m.this.cXML =
<HTMLDialog>
  <DivID>rowsourcetype_scx</DivID>
  <DivTitle>RowSourceType processing by FoxInCloud -
FoxInCloud Tutorial</DivTitle>
  <DivHTML>
    &#60;!-- Généré le 01/11/13 16:24:54 par FoxInCloud version
2.00 beta2 --&#62;
```



Browser receives XML response

Web dev. Tools such as Firebug for Firefox provide a very handy debugging environment; here we can see the full XML response received from FAS server, original request, and much more.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?'><IntuiCat>
<_script>(function(){ var o = $('link[type="text/css"][href="/tutotest/awDefault_"]')[0]; /* "" for
IE */ if (o &#38;&#38; !o.href.endsWith("/tutotest/awDefault_3X00ZMP5H.css")){ var p = o.previousSiblings
() [0]; o.remove(); o = Object.extend(document.createElement('link'), {href:"/tutotest/awDefault_3X00ZMP5H
.css", type:'text/css', rel:'stylesheet'}) ; p.insert(after:o); } })();</_script>
<HTMLDialog>
  <DivID>rowsourcetype_scx</DivID>
  <DivTitle>RowSourceType processing by FoxInCloud - FoxInCloud Tutorial</DivTitle>
  <DivHTML>
    &#60;!-- Généré le 01/11/13 16:37:21 par FoxInCloud version 2.00 beta2 --&#62;
    &#60;div id=&#34;rowsourcetype_scx&#34; class=&#34;form awfrm ficfrm&#34; tabindex=&#34;9&#34;&#62;

    &#60;div id=&#34;rowsourcetype_scx-tutopgf&#34; class=&#34;pageframe awpgf ficpgf&#34; style=&#34;display
:block;&#34; tabindex=&#34;10&#34;&#62;
    &#60;div id=&#34;rowsourcetype_scx-tutopgf-tutopage1&#34; class=&#34;page awpag ficpag&#34; style=&#34;#34
;display:block;&#34;&#62;
    &#60;div id=&#34;rowsourcetype_scx-tutopgf-tutopage1-tutoinfofic&#34; id=&#34;rowsourcetype_scx-tutopgf-tutopage1-tutoinfofic
&#34; class=&#34;editbox awebx ficbxs&#34; style=&#34;display:block;&#34; title=&#34;Tastrate Products&#34; tabindex=&#34;12&#34; size
=&#62;&#60;/div&#62;
    &#60;label id=&#34;rowsourcetype_scx-tutopgf-tutopage1-myblttype2&#34; class=&#34;label awlbl ficlbl
&#34; style=&#34;display:block;&#34;&#62;
    &#60;select id=&#34;rowsourcetype_scx-tutopgf-tutopage1-mylistboxchildtype2&#34; class=&#34;listbox awlst
ficlst&#34; style=&#34;display:block;&#34; title=&#34;Tastrate Suppliers&#34; tabindex=&#34;13&#34; size
=&#34;2&#34; autocomplete=&#34;off&#34;&#62;
    &#60;/select&#62;
    &#60;select id=&#34;rowsourcetype_scx-tutopgf-tutopage1-mylistboxparent&#34; class=&#34;listbox awlst
ficlst&#34; style=&#34;display:block;&#34; title=&#34;Tastrate Suppliers&#34; tabindex=&#34;13&#34; size
=&#34;2&#34; autocomplete=&#34;off&#34;&#62;
      &#60;options&#62;Aux joyeux ecclésiastiques&#60;/options&#62;
      &#60;options&#62;Bigfoot Breweries&#60;/options&#62;
      &#60;options&#62;Cooperativa de Quesos &#39;Las Cabras&#39;&#60;/options&#62;
      &#60;options&#62;Escargots Nouveaux&#60;/options&#62;
      &#60;options&#62;Exotic Liquids&#60;/options&#62;
      &#60;options&#62;Formaggi Fortini s.r.l.&#60;/options&#62;
      &#60;options&#62;Porêts de&#39;érables&#60;/options&#62;
      &#60;options&#62;G&#39;day, Mates&#60;/options&#62;
      &#60;options&#62;Gai pâturage&#60;/options&#62;
      &#60;options&#62;Grandma Kellys&#39; Homestead&#60;/options&#62;
      &#60;options&#62;Heli Süßwaren GmbH &#38;&#38; Co. KG&#60;/options&#62;
  </DivHTML>
</HTMLDialog>
</IntuiCat>
```

Any question? Post with screenshot in the 'FoxInCloud' section of <http://west-wind.com/wwThreads/> or <http://www.universalthread.com/>



FoxInCloud.js executes orders read in XML response

FAS builds an XML response to AJAX requests (mainly user events).

XML response embeds JS instructions for the HTML display to reflect VFP form's state after user action, just like in desktop.

```
916 // Si aucune erreur serveur
917 var loError = this.oDOMElChild(loIntuiCat, 'ServerError'); // cf. awAJAX.XMLHttpRequest()
918 llResult = ! Boolean(loError);
919 if (llResult){
920
921     // Si la version du serveur est fournie, mémoriser et vérifier
922     var serverVersion = this.oDOMElChild(loIntuiCat, 'ServerVersion');
923     if (serverVersion) this.serverVersion = parseFloat(serverVersion.firstChild.nodeValue);
924
925     // Si le résultat de l'événement est fourni, mémoriser
926     this.uResult = this.oDOMElChild(loIntuiCat, 'Result');
927     this.uResult = this.uResult
928     ? this.uValue
929       ( this.oDOMElChild(this.uResult, 'v').firstChild.nodeValue /* valeur littérale cf.
930         , this.oDOMElChild(this.uResult, 't').firstChild.nodeValue /* type cf. awAJAX.XMLRe
931       )
932     : null;
933
934     // Si des scripts sont à exécuter AVANT l'affichage du dialogue
935     script = this.oDOMElChild(loIntuiCat, '_script');
936     if (script){
937         script = script.firstChild.nodeValue;
938         if(script) eval(script); /* script.evalScripts.defer(); */
939     }
940
941     // Afficher les dialogues éventuels
942     this.oDOMElChildren(loIntuiCat, 'HTMLDialog').each(function(loDialog){ // cf. awAJAX.X
943         var llDialog = this.DialogShow(loDialog);
944         if (!llDialog){
945             this.Error((this.langFR ? "Échec de l'affichage du dialogue modal ." : "Modal dialog
946             + ' ' + this.oDOMElChild(loDialog, 'DivID').firstChild.nodeValue);
947         }
948         llResult = llResult && llDialog;
949     }.bind(this));
950     if (llResult){
951
952         // Si des scripts sont à exécuter AVANT les modifications du DOM
953         script = this.oDOMElChild(loIntuiCat, 'script');
954         if (script){
955             script = script.firstChild.nodeValue;
956             if(script) eval(script); /* script.evalScripts.defer(); */
957         }
958     }
959 }
```



Child form shows up in the browser

By comparing the state of each form before and after user action

[`.Click()` in this case], FAS could send all the necessary orders for `FoxInCloud.js` to update the client HTML page accordingly

