

## Mise à jour d'une application / Actiona

### Le besoin :

Automatiser la mise à jour ou la consultation dans une application en utilisant un fichier de données en entrée.

Depuis 2 ans, pour répondre à ce type de besoin j'utilise Actiona.

Depuis la version 3.7 (Mai 2013), je n'ai constaté aucun plantage dû à Actiona. Les scripts traitent des milliers de données et tournent pendant plusieurs heures par jour.

Je propose une démonstration. Si c'est concluant, le but sera de former une personne.

### Objectifs de la démonstration :

- Montrer les possibilités d'actiona pour ce type de besoin.
- Faire un tour d'horizon de ce qui est mis en œuvre.

### Contrainte :

Pour préparer cette démonstration, je n'ai pas accès aux applications des personnes intéressées.  
J'ai donc créé une mini application pour montrer les possibilités d'actiona pour répondre à ce type de besoin.

### 4 fiches :

- Présentation de l'application de démonstration.
- Adaptations à faire pour une première utilisation sur un PC.
- Tour d'horizon des ressources utilisées (répertoires, fichiers, ..)
- Le script de démonstration.

*Libre choix de l'emplacement du répertoire **Scripts\_actiona**.  
Il y a des adaptations à faire pour une première utilisation (voir 2ème fiche).*

## Présentation de l'application de démonstration

mise\_a\_jour\_appli(demo)\_1.htm

mise\_a\_jour\_appli(demo)\_2.htm

Exemple de mise à jour dans une application  
En utilisant Actiona

Code SIREN de l'entreprise

envoi

Forme juridique de l'entreprise

Nombre d'employés

SA  
EURL  
SCI  
SARL  
GFA

Envoi

Retour

Le script lit un fichier en entrée ('entrep.csv') avec :  
siren ; forme juridique ; nombre d'employés.

Pour se rapprocher d'une vraie application :

- Il y a des données volontairement invalides.

Via du DHTML dans les deux pages HTML :

- Simulation du chargement de chaque page (temps de chargement différent)
- Message analogue à celui que pourrait retourner une vraie application.

Ici

**Cette entreprise a déjà fait l'objet d'une mise à jour en 2014\***

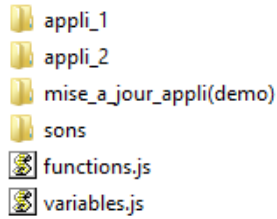
Le message est généré aléatoirement avec une proba de 1/10.  
Le but étant de montrer qu'avec Actiona, on peut gérer les cas particuliers.

- Bilan de passage. Fichier : bilan\_demo.csv.

```
siren ; RET_1 ; RET_2
552081317;OK;Mise à jour effectuée
123456789;W;Siren non valide
346000000;W;Siren déjà pris en compte
349098765;W;fjur = SASU / pas connue dans specific.js
```

## Adaptations à faire pour une 1ere utilisation sur un PC

### Arborescence géné



Adaptations à faire dans le répertoire **mise\_a\_jour\_appli(demo)**

Script Actiona : mise\_a\_jour\_appli(demo).ascr

#### 1) Répertoires

Dans le script Actiona, dans action :

**Code**      Initialisation : notions inter scripts

```
//=====
rep_actiona = "G:\\scripts_actiona";
//=====
```

Dans le script Actiona, dans action :

**Code**      Initialisation : notions spécifiques à ce script

```
appli = "mise_a_jour_appli(demo)";
```

Dans 'specific.js' :

```
// en réel, ces 2 répertoires pourraient être ailleurs.
rep_fichier_IN= rep_actiona + "\\ " +appli;
rep_fichier_OUT= rep_actiona + "\\ " +appli;
```

#### 2) Résolution d'écran

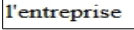








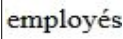

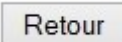
Les tests ont été faits avec : 1366 X 768.

Si <>, aller dans 'variables.js'

```
//=====
ecran_dimensions= obj_System.screenGeometry()
actiona_x= 0;
actiona_y= 0;
actiona_width=1366;
actiona_height=768;
//=====
```

### 3) Les images à trouver, (voir répertoire img)

Images éventuellement à refaire (notamment si résolution d'écran <>)

Page	Nom du fichier	Image	Précision
1	P11_libelle_siren.jpg		
1	P13_envoi.jpg		
2	P21_select_forme_juridique.jpg		
2	P22_entreprise_deja_connue.jpg		Apparaît aléatoirement page 2.
2	p24_fjur_GFA.jpg		
2	P24_fjur_SCI.jpg		
2	P24_fjur_SARL.jpg		
2	P24_fjur_SA.jpg		
2	P24_fjur_EURL.jpg		
2	P25_nb_employes.jpg		
2	P26_envoi.jpg		
2	P27_retour.jpg		

### 4) Prise en compte des décalages entre l'image à trouver ET la zone de saisie.

Voir dans 'specific.js' :

```
// page 1
p11_libelle_siren = rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p11_libelle_siren.jpg\"
p11_decalage_x = 145;
p11_decalage_y = 0;
```

Idem avec p25\_nb\_employes.

On peut maintenant lancer le script

Penser à ouvrir auparavant la page html : mise\_a\_jour\_appli(demo)<sub>1</sub>.htm

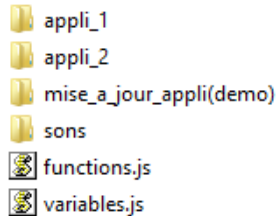
## Tour d'horizon des ressources utilisées (répertoires, fichiers, ..)

La mise en œuvre présentée ici serait un peu lourde pour un seul script.

En effet, on prépare ici un environnement avec des scripts pour N applications.

- Utilisation de ressources communes à tous ces scripts.
- Normalisation pour l'ensemble des scripts.

### 1) Sous le répertoire `scripts_actiona`



Un sous répertoire par application.

Le sous répertoire 'sons' contient les sons émis lorsque le traitement avance de + 5 %.  
Peut être utilisé par tous les scripts.

Le fichier 'functions.js' contient les fonctions pouvant être appelées par tous les scripts.

Exemple :

```
function affichage_percent(ctr_fichier_in,nb_arts_in_total,date_debut)
{
    //=====
    // affichage du travail fait / à faire
    //=====
    date_actuelle= new Date();
    percent = Math.round(ctr_fichier_in/ nb_arts_in_total *100)
    // Console.print(percent);
    if (percent != ante_percent && percent % 5 == 0 )
    {
```

Le fichier 'variables.js' contient les variables communes à tous les scripts

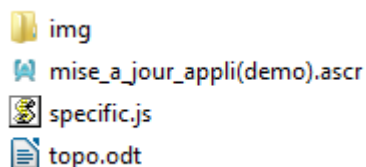
Exemple :

```
//=====
ecran_dimensions= obj_System.screenGeometry()
actiona_x= 0;
actiona_y= 0;
actiona_width=1366;
actiona_height=768;
//=====
```

### 2) Sous chaque répertoire 'appli'

Tous ces répertoires sont organisés de la même façon.

exemple : 'mise\_a\_jour\_appli(demo)'



En plus du script (voir fiche suivante) et de la documentation :

**Le répertoire 'img' contient toutes les images utilisées dans ce script**

Exemples :



**Le fichier 'specific.js' contient ce qui est spécifique à ce script.**

Exemples :

```
// en réel, ces 2 répertoires pourraient être ailleurs.
rep_fichier_IN= rep_actiona + "\\" +appli;
rep_fichier_OUT= rep_actiona + "\\" +appli;
```

```
//=====
// Images à reconnaître
//   -> figurent ici dans l'ordre d'utilisation (pratique lors de la mise à jour)
//   -> p53_exemple serait la 3eme image contenue dans la 5eme page.
//=====
// page 1
p11_libelle_siren = rep_actiona + "\\" + appli + "\\" + "img" + "\\" + "p11_libelle_siren.jpg"
p11_decalage_x = 145;
p11_decalage_y = 0;
```

Le détail du fichier 'specific.js' relatif au script de démo sera vu dans la fiche suivante.

Il peut y avoir plusieurs scripts pour la même application. Dans ce cas, les mettre sous le même répertoire afin de mutualiser les ressources communes (dans img et specific.js)

## Script de Démo

### 1) Début du script

Code	Test version actiona
Code	Initialisation : notions inter scripts
Code	Initialisation : notions spécifiques à ce script
Code	Résolution ecran correcte ?
Code	La page html est-elle ouverte ?

ne opération

Zoom sur 'initialisation : notions spécifiques à ce script'

Toutes les variables spécifiques à un script sont dans ce fichier :

- Synthèse.
- Facilite la maintenance.
- Il est possible de visualiser en même temps le script et ce fichier.

```
// en réel, ces 2 répertoires pourraient être ailleurs
rep_fichier_IN= rep_actiona + "\\\" +appli;
rep_fichier_OUT= rep_actiona + "\\\" +appli;
```

```
//=====
// IN
//=====
nom_fichier_in= "entrep.csv"
max_demandes_pec=10000 // Assurer le coup

// s'arrêtera s'il y a 'nb_max_anos'
nb_max_anos=10
//=====
```

```
//=====
// récupérer les données
//=====
var fich_in = new File();
fich_in_chemin_complet= rep_fichier_IN + "\\\" + nom_fichier_in

fich_in.open(fich_in_chemin_complet,File.ReadOnly| File.Text);
au_km = fich_in.readText();
fin_ligne =String.fromCharCode(10);
t_ligne =au_km.split(fin_ligne);
nb_arts_in= t_ligne.length;
```

### Recherche dans un SELECT html

```
// page 2
p21_SELECT_forme_juridique = rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p21_SELECT_forme_juridique.jpg\";
p22_entreprise_deja_connue = rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p22_entreprise_deja_connue.jpg\";

//=> les <option> du <select forme juridique>
var t_fjur = new Array(); // les images des différentes formes juridiques
var t_fjur_1 = new Array(); // la confiance mini pour la reconnaissance de chaque image

t_fjur[\"SA\"] =rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p24_fjur_SA.jpg\";
t_fjur[\"EURL\"] =rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p24_fjur_EURL.jpg\";
t_fjur[\"SCI\"] =rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p24_fjur_SCI.jpg\";
t_fjur[\"SARL\"] =rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p24_fjur_SARL.jpg\";
t_fjur[\"GFA\"] =rep_actiona + "\\\" + appli + "\\\" + \"img\" + "\\\" + \"p24_fjur_GFA.jpg\";
```

Si nouvelle Option dans le SELECT html, il suffit de l'ajouter ici. Le reste du script n'est pas à modifier.

La confiance mini par image :

Trop élevée, risque de ne pas être trouvée.

Trop faible, risque de trouver . . . n'importe quoi.

A l'usage,  
Le plus délicat dans un script

```
t_fjur_1['SA'] =93;  
t_fjur_1['EURL']=90;  
t_fjur_1['SCI']=92;  
t_fjur_1['SARL'] =90;  
t_fjur_1['GFA']=90;
```

```
// la propriété 'length' ne fonctionne pas avec les tableaux associatifs d'où ce qui suit :  
var nb_fjurs = 0;  
for (var key in t_fjur){if (t_fjur.hasOwnProperty(key)) nb_fjurs++;}  
// Console.print(nb_fjurs)
```

## 2) Suite du script

Zoom sur quelques actions.

### Code Début article à traiter + traitement de fin de fichier

```
1 //=====
2 // test fin
3 //=====
4 indic= 0;
5 if (ctr_fichier_in ==max_demandes_pec) {indic= 1};
6 if (ctr_anos ==nb_max_anos) {indic= 1};
7 if (ctr_fichier_in == nb_arts_in) {indic= 1};
8
9 if (indic ==1)
10 {
11 Console.print("----")
12 Console.print(" Nb art. dans fichier : " + nb_arts_in );
13 Console.printWarning(" Nb anos rencontrées " + ctr_anos
14 fich_in.close();
15 rc= fin_trait_normal(ctr_fichier_in,nb_arts_in,date_debut)
16 }
```

```
1 //=====
2 // Lire table t_ligne
3 //=====
4 pos_1=t_ligne[ctr_fichier_in].indexOf(":",0)
5 pos_2=t_ligne[ctr_fichier_in].indexOf(":",pos_1+1)
6 siren= t_ligne[ctr_fichier_in].substr(0,pos_1)
7 fjur= t_ligne[ctr_fichier_in].substr(pos_1+1,pos_2-pos_1-1)
8 nb_employes= t_ligne[ctr_fichier_in].substr(pos_2+1)
9 // Console.print( "fjur >" +fjur + "<" )
```



## Code Verification de la validité formelle des données

```
switch(true)
{
  case !verif_siren_K_valide(siren) :
    ano=1
    RET_1= "W"
    RET_2= "Siren non valide"
    break;

  case t_fjur[fjur] === undefined :
    // la forme juridique n'est pas connue dans specific.js
    ano=1
    RET_1= "W"
    RET_2= "fjur = " + fjur + " / pas connue dans specific.js"
    break;

  case isNaN(nb_employes) :
    ano=1
    RET_1= "W"
    RET_2= "nb_employes : " + nb_employes + " / Nombre ??"
    break;
} // fin switch
```

## Code Ecriture dans fichier 'bilan'

```
1 fichier_bilan.open(fichier_bilan_chemin_complet,File.WriteOnly | File.Text | File.Append);
2 tt = siren + "," + RET_1 + "," + RET_2 + "\n";
3 fichier_bilan.writeText(tt);
4 fichier_bilan.close();
5
```

## Trouver image Recherche t\_fjur['fjur']

Source :	Capture d'écran
Titre de fenêtre :	nom_page
Position relative à la fenêtre :	<input type="checkbox"/>
Image où chercher :	
Image à chercher :	t_fjur[fjur]
Si trouvé :	Ne rien faire
Si non trouvé :	Ne rien faire

Standard		Avancé		Commun		Exceptions	
Paramètres d'entrée							
Méthode :	Correlation Coefficient						
Confiance minimale :	t_fjur_1[fjur]						
Nombre maximum de résultats :	10						
Sous-échantillonnage :	1						
Expansion de la recherche :	1						
Délai entre deux recherches lors d'une attente :	100 ms						
Paramètres de sortie							
Confiance :	confiance						

Nombre maximum de résultats = 10 alors que 'logiquement' il ne doit y en avoir qu'un seul .  
Cela permet de voir si l'image à trouver est pertinente. Si nb > 1, bug et donc refaire cette image à trouver.

Détecté dans : **Code** **Forme juridique trouvée ??**

**Visible en fin de traitement :**



A screenshot of a console window with a light blue header bar. It displays a series of log entries, each preceded by a blue speech bubble icon. The entries track progress over time, showing the percentage of work completed and the remaining time. A separator line of four dashes is used to group related information. An orange warning icon precedes a message about the number of years encountered. The final line indicates the end time of the process.

```
07:38:57 : Réalisé :75 %Temps restant 0h3'8"  
07:39:41 : Réalisé :80 %Temps restant 0h2'32"  
07:40:21 : Réalisé :85 %Temps restant 0h1'54"  
07:41:04 : Réalisé :90 %Temps restant 0h1'16"  
07:41:46 : Réalisé :95 %Temps restant 0h0'38"  
07:42:22 : Réalisé :100 %Temps restant 0h0'0"  
----  
Nb articles dans fichier : 121  
(Nb max articles pris en compte : 10000)  
⚠ Nb anos rencontrées 5  
Voir dans bilan_ano.csv  
Heure fin trait : 07:42:23
```

*Peut être visible lors de l'exécution du script.*  
**Dans Actiona : Outils / Options / Actions**

