

## Présentation Serveur Apache et pour RePeGlio

I.	Présentation générale du serveur Apache .....	2
1.	Fonction du serveur Apache.....	2
2.	Fonction du programme RPG relativement au serveur Apache. ....	2
3.	Pré-requis pour le serveur APACHE et CGIDEV2 .....	2
4.	Objectifs de la brochure .....	3
II.	Division en instances du serveur Apache .....	4
1.	Le port .....	4
2.	Le fichier de configuration.....	4
3.	Ouverture d'une instance d'Apache .....	4
III.	Le fichier de configuration.....	5
1.	Instructions de base du fichier de configuration.....	5
2.	Autorisations dans le fichier de configuration .....	5
IV.	RePeGlio: création automatique d'une instance Apache .....	6
1.	Objectif .....	6
2.	Menu APAWRK00 et création instance .....	6
3.	Inspection de l'instance créée.....	7
4.	Particularités de l'instance pour les pages dynamiques .....	7
5.	Script AliasMatch et AliasMatch.....	7
6.	Profil utilisateur de type *vldl .....	8
V.	Partie système Web gérée par RePeGlio.....	9
1.	Les limitations système du serveur Apache .....	9
2.	Profil utilisateur de type IBM i ou OS/400 géré par Apache. ....	9
3.	Profil utilisateur de type *vldl .....	9
4.	Ajouter un profil *vldl RePeGlio .....	10
5.	Gestion des bibliothèques avec Apache.....	11
6.	Minimum requis : le profil et la liste des bibliothèques.....	11
7.	Principe d'exécution d'un programme RPG ou *pgm via Apache .....	11
8.	Suppression du profil *vldl pour accès direct sans profil:.....	12

## I. Présentation générale du serveur Apache

### 1. Fonction du serveur Apache

Le serveur Apache a pour fonction d'envoyer une page statique Web au client qui en a fait la demande. La page statique web est identifiée par un URL côté serveur. Le client est identifié par une adresse IP unique.

On appelle page statique, comparé aux écrans de gestion, une page qui ne contient aucune donnée variable mais des constantes, des images ou vidéos qui sont des fichiers stream statiques de toute nature (suite d'octets en ASCII).

On appelle page dynamique, une page qui contient des données variables issues généralement de fichiers.

Un site internet est typiquement composé de pages statiques. Un relevé mensuel de banque en ligne est composé essentiellement de données dynamiques issues de fichiers.

Le serveur Apache ne sait qu'envoyer des pages statiques à celui qui en a fait la demande via un URL comme <http://www.repeglio.com> (nous détaillerons plus loin la syntaxe de l'URL)

### 2. Fonction du programme RPG relativement au serveur Apache.

Pour reprendre l'exemple de la liste des mouvements de compte bancaire en ligne, Apache ne sait pas remplir la page puisque sa vocation est l'envoi de la page statique déjà remplie.

Dans le fichier de configuration d'Apache, nous aurons typiquement une instruction de type :

```
ScriptAliasMatch /serverp/(.*) /qsys.lib/JEAND.lib/$1.pgm
```

ScriptAliasMatch/serverp/ signifie qu'à chaque fois qu'Apache sera en présence d'un URL contenant /serverp/, il va appeler au préalable le programme \*pgm \$1 indiqué dans l'URL à la suite. Ici, ce programme RPG se trouve dans la bibliothèque JEAND.lib C'est ainsi qu'Apache sait lorsqu'il convient d'appeler un programme RPG au préalable pour remplir la page dynamique.

Le programme RPG appelé aura pour fonction de remplir la page html à l'aide de variables de remplacement et de sections. Ce procédé de remplissage est tout à fait similaire au DSPF. Par exemple pour la liste des comptes, seule la première ligne html est décrite, comme la ligne du sous-fichier. Il y aura dans chaque cellule des variables de remplacement qui sont renseignées par le programme RPG à l'aide d'instructions CGIDEV2. Ces instructions CGIDEV2 sont des APIs développées par IBM et disponibles gratuitement.

Une fois la page html remplie, le programme RPG donne l'ordre à APACHE de l'envoyer comme une page statique.

### 3. Pré-requis pour le serveur APACHE et CGIDEV2

Apache est installé par défaut sur tous les IBMi. Pour voir si le serveur Apache est présent sur votre IBM i, lancer la commande `WRKACTJOB QHTTPSrv`

Pour tester l'ouverture de l'instance d'Apache par défaut lancer la commande :

```
STRTCPSVR SERVER(*http) HTTPSvr(APACHEDFT), APACHEDFT étant l'instance standard par défaut.
```

A l'aide d'un navigateur du réseau local, saisir votre adresse ip locale comme par exemple :

<http://192.168.1.2> où 192.168.1.2 est votre adresse IP locale (comme pour Client Access).

Vous accèderez à la page statique par défaut de l'instance APACHEDFT ouverte sur le port 80 par défaut. Cette page statique par défaut s'appelle toujours index.html

Puis `WRKACTJOB QHTTPSrv` pour voir le travail ouvert de l'instance APACHEDFT

Et enfin :

```
ENDTCPSVR SERVER(*http) HTTPSvr(APACHEDFT) pour fermer APACHEDFT
```

#### 4. Objectifs de la brochure

La brochure a pour objectif :

- ) de créer automatiquement une instance d'Apache pour RePeGlio
- ) d'expliquer le fonctionnement du serveur Apache afin que le développeur puisse être autonome et progresser dans son utilisation.

Notons que cette brochure précède:

- ) la brochure des transactionnels Web
- ) Smart Excel Report Web

A l'issue de cette brochure, le stagiaire connaîtra les principes de base du serveur Apache et pourra gérer une instance d'Apache pour RePeGlio.

Note : la technologie que nous utilisons est parfaitement conforme aux standards du marché. Le principe de la séparation entre la page HTML et le programme a été validé par Google.

## II. Division en instances du serveur Apache

### 1. Le port

Le serveur Apache est divisé en instances. Chaque instance est identifiée par le port qui doit être unique lors du fonctionnement de l'instance. Le port est similaire à une porte. Le port :80 est le port par défaut et n'a pas besoin à ce titre d'être spécifié dans l'URL.

### 2. Le fichier de configuration

Chaque instance d'Apache possède son propre fichier de configuration. Ce fichier de configuration indique le port, le chemin pour aller à la racine, les sécurités, le programme RPG à appeler éventuellement etc...

Nous allons voir concrètement comment une instance d'Apache est ouverte sur IBM i.

### 3. Ouverture d'une instance d'Apache

L'emplacement de chaque instance d'Apache est répertorié sur IBM i dans un membre d'un fichier système, ce membre ayant le même nom que l'instance d'Apache. Concrètement, lorsque la commande `STRTCPSVR SERVER(*http) HTTPSVR(APACHEDFT)` est lancée pour ouvrir l'instance APACHEDFT qui est l'instance par défaut, le système va rechercher le membre :

`QUSRSYS/QATMHINSTC(APACHEDFT)`

Il est intéressant d'examiner ce membre avec la commande

`WRKMBRPDM FILE(QUSRSYS/QATMHINSTC) MBR(APACHEDFT)`

Ce membre ne contient qu'une seule ligne qui indique où est située l'instance dans l'IFS ainsi que le fichier de configuration:

`-apache -d /www/apachedft -f conf/httpd.conf -AutoStartN`

Ainsi, la commande saura que l'instance APACHEDFT (membre) est située dans l'IFS à l'adresse `/www/apachedft`

Et que le fichier de configuration se trouve à l'adresse :

`/www/apachedft/conf/httpd.conf`

AutoStartN signifie que l'instance d'Apache ne sera pas automatiquement ouverte à l'IPL. Pour ouvrir une instance automatiquement à l'IPL indiquer AutoStartY en modifiant la ligne avec UPDDTA.

Ainsi ce membre `QUSRSYS/QATMHINSTC(nominstance)` est le sommet de la pyramide. A l'IPL, chaque membre du fichier est lu pour savoir s'il faut ouvrir automatiquement une instance et pour trouver le fichier de configuration. Le membre est également utilisé avec la commande de démarrage manuel :

`STRTCPSVR SERVER(*http) HTTPSVR(NomInstance).`

Afin de trouver dans l'IFS et mémoriser les instructions du fichier de configuration.

### III. Le fichier de configuration

Pour rechercher le fichier de configuration de l'instance par défaut APACHEDFT, partez du membre comme le ferait la commande STRTCPSVR :

```
WRKMBRPDM FILE(QUSRSYS/QATMHINSTC) MBR(APACHEDFT)
```

Puis à l'aide des informations du membre lancer la commande :

```
WRKLNK OBJ('/www/apachedft/conf/httpd.conf')
```

#### 1. Instructions de base du fichier de configuration

# indique un commentaire sur la ligne

Listen \* :80 signifie que le port 80 est utilisé. Ce port est celui d'internet et est réputé ouvert par défaut pour accéder au Web à et depuis l'extérieur. Retenir que le port doit être unique lorsque l'instance est en fonctionnement (port=porte).

DocumentRoot /www/apachedft/htdocs indique la racine de l'instance. Ainsi, lorsque l'instance est en fonctionnement STRTCPSVR, le port 80 est ouvert et la racine est /www/apachedft/htdocs. Autrement dit, si une page html est placée à la racine de l'instance, elle peut être recherchée en indiquant avec l'url <http://adresseIP/nom.htm>. Par défaut, Apache sait où rechercher la page à la racine car il a mémorisé les instructions de l'instance identifiée par le port dans le fichier de configuration à l'ouverture. Notons que la page par défaut à la racine sera <http://adresseIP> soit index.html. Rappel, le port 80 est le port par défaut et index.html la page par défaut.

#### 2. Autorisations dans le fichier de configuration

<Directory>.....</Directory> est un markup langage avec un début et une fin. Les instructions entre les deux balises sont relatives principalement aux autorisations. Il faut savoir qu'Apache exclut tous les droits à tous par défaut : « deny from all », puis donne les autorisations une à une ici :

```
<Directory /www/apachedft/htdocs>
```

```
order allow,deny
```

```
allow from all
```

```
</Directory>
```

www/apachedft/htdocs où se trouvent les pages statiques html sera autorisé pour tout le monde « allow from all ».

Il est donc fréquent d'avoir plusieurs dizaines de <Directory>.....</Directory> autant que de directories à autoriser pour le bon fonctionnement de l'instance.

Retenir que si vous créez un sous-lien à la racine, vous devrez l'autoriser dans le fichier de configuration pour l'instance avec

```
<Directory chemin>.....allow from all.....</Directory>
```

#### Note relative au fichier de configuration :

Dans la pratique il faut se rappeler que le fichier de configuration est mémorisé une seule fois par Apache à l'ouverture. Si vous modifiez le fichier de configuration il faudra donc fermer l'instance ENDTCPVSR puis l'ouvrir à nouveau STRTCPSVR afin que la modification soit prise en compte.

## IV. RePeGlio: création automatique d'une instance Apache

### 1. Objectif

L'objectif est de créer automatiquement une instance d'Apache pour utiliser le produit RePeGlio.

**Note importante:** l'instance Apache sera créée pour des pages Web dynamiques (et statiques).

Pré-requis: il est conseillé d'avoir lu les trois chapitres précédents.

### 2. Menu APAWRK00 et création instance

- Ouvrir une session avec le profil QSECOFR
- Lancer la commande ADDLIBLE ARC\_OBJ
- Puis CALL APAWRK00

```

GERER INSTANCE APACHE SOUS QSECOFR

Gérer instance Apache au quotidien
1. Démarrer STRTCPSVR
2. Terminer ENDTCPSVR
3. wrkactjob QHTTSPVR
4. Clean .doc .xls dans pictures
5. WRKLNK 'www/instance'

Autres commandes utiles
21. Autorisation pictures
22. Mise à jour fichier de config.
23. Mise à jour QUSRSYS/QATMHINSTC

CREER/ANNULER Instance Apache
31. CREER une instance
32. ANNULER une instance sauf *lib

34. Restaurer images d'origine

Attention: Lire la documentation avant utilisation options 31, 32 et 33
Entrer Option: 31
  
```

Option 31 et 32 permettent de créer/annuler des instances.

Option 31 Créer

Puis entrer le nom de l'instance  
Profil de développement \*DFT

Puis F9=Valider .... Attendre jusqu'à la fin de la procédure de création.

```

GERER INSTANCE APACHE SOUS QSECOFR

Créer une instance

Nom instance. . . . . : RPGINST
Nom profil developpement : *DFT ou *DFT

Note 1: Une bibliothèque de même nom contenant
les programmes RPG CGIDEV2 sera créée
Note 2: Il peut y avoir un profil de dev. par
défaut pour tous les profils de dev.

F9=Valider      F12=Abandon

Attention: Lire la documentation avant utilisation option
Entrer Option: 31
  
```

### 3. Inspection de l'instance créée

```
Autres commandes utiles
21. Autorisation pictures
22. Mise à jour fichier de config.
23. Mise à jour QUSRSYS/QATMHINSTC
```

23 inspections du membre de même nom que l'instance dans le fichier système IBM QATMHINSTC

```
-apache -d /www/RPGINST -f conf/httpd.conf -AutoStartY
```

Le fichier de configuration instance se trouve dans /www/RPGINST/conf/httpd.conf  
AutoStartY signifie que l'instance va démarrer automatiquement à chaque IPL. Mettre -AutoStartN sinon.

### 4. Particularités de l'instance pour les pages dynamiques

```
Autres commandes utiles
21. Autorisation pictures
22. Mise à jour fichier de config.
23. Mise à jour QUSRSYS/QATMHINSTC
```

22 Mise à jour du fichier de configuration

Listen 7500 Par défaut le port est 7500. Rappel : pour accéder à l'extérieur il faut le port 80.  
DocumentRoot /www/RPGINST/htdocs/Pictures est la racine  
Nous allons voir maintenant les instructions importantes Alias

### 5. Script AliasMatch et AliasMatch

Principe : **pour ajouter une directory qui ne se trouve pas à la racine de l'instance** il est obligatoire de passer par les instructions Alias. En effet, par défaut, l'instance est limitée à sa racine qui s'exprime dans le html par ../  
Si par exemple nous voulons ajouter une directory utilisateur pour y placer des .pdf et des .xls générés automatiquement par RePeGlio, vous pouvez mettre cette directory à la racine ou si cette directory se trouve dans l'IFS en dehors de la racine procéder comme suit : Ajouter l'instruction AliasMatch :

```
AliasMatch /madir/ /repeglio/
```

A chaque fois qu'il y aura dans l'url <http://198.162.1.2/madir/mondoc.pdf>

Le système va remplacer /madir/ par /repeglio/ (match) en dehors de la racine de l'instance, à partir de la racine de l'IFS.

A retenir :

AliasMatch est utilisé pour un document

ScriptAliasMatch est utilisé **pour appeler un programme**, ici un programme RPG de type \*pgm

L'instruction :

```
ScriptAliasMatch /serverp/(.*) /qsys.lib/RPGINST.lib/$1.pgm
```

Signifie qu'à chaque fois qu'il y aura /serverp/ dans l'URL, Apache va appeler le programme RPG (\$1.pgm) dans la bibliothèque RPKINST (/qsys.lib/RPGINST.lib/) note : /QSYS .lib/ se trouve bien entendu en dehors de la racine de l'instance, mais à partir de la racine du système IBM i, d'où Alias. Script signifiant programme à exécuter et Match les remplacements.

Par exemple l'url suivante :

<http://192.168.1.2:7500/serverp/manoirhqp>

appellera le programme manoirhqp dans la bibliothèque RPKINST car /serverp/ est défini avec ScriptAliasMatch au niveau du fichier de configuration. Rappel: la fonction du programme RPG appelé est de remplir la partie variable du programme afin de délivrer au serveur Apache une page statique HTML prête à être lue par le navigateur du poste client.

### 6. Profil utilisateur de type \*vldl

```
<LocationMatch /serverp/(.*)$>  
AuthName "Application"  
AuthType Basic  
PasswdFile RPGINST/MYVLDL  
Require valid-user  
</LocationMatch>
```

A chaque ouverture de Session pour un utilisateur avant le programme appelé /serverp/ Le système va contrôler le profil et le mot de passe, objet \*vldl MYVLDL qui se trouve dans la bibliothèque RPGINST de l'instance.



## V. Partie système Web gérée par RePeGlio

### 1. Les limitations système du serveur Apache

Il faut absolument retenir que le serveur Apache a été conçu pour envoyer des pages statiques à une adresse IP et à recevoir les demandes via des URL. Par défaut, Apache envoie une seule page à la fois. Cette page est par défaut complètement déconnectée de la page d'avant et de la page d'après. Il faut retenir que les pages d'avant peuvent être rappelées car elles sont mémorisées en local dans la mémoire du navigateur. Cette informatique est dite stateless, sans état persistant, ou encore événementielle dans la mesure où l'utilisateur est seul maître de sa navigation. Apache fonctionne parfaitement bien pour la gestion des pages statiques. Pour la gestion des pages dynamiques, le développeur doit prendre en charge tout ce qu'Apache ne fait pas. Nous avons développé un certain nombre d'outils standards dont vous avez les sources dans la bibliothèque ARC\_SRC.

### 2. Profil utilisateur de type IBM i ou OS/400 géré par Apache.

Il faut retenir qu'Apache, via un navigateur, de connaît au niveau des sécurités que deux profils de type IBM i :

- a) QTMHHTTP qui gère uniquement les sécurités au niveau du contenu comme par exemple le droit d'afficher une image ou de modifier le contenu d'un document.
- b) QTPHHTP1 qui gère uniquement le contenant comme par exemple le droit d'annuler ou de renommer un document.

Les droits d'une directory ou d'un document sont gérés avec la commande CHGAUT. Notons que les droits autres que ceux accordés à QTMHHTTP et QTMHHTP1 sont ceux gérés par un utilisateur IBM i via Client Access.

Il faut retenir que renommer un objet avec Client Access nécessite des droits accordés au profil utilisateur de type IBM i. Gérer le contenu ou le contenant d'un objet via un navigateur et Apache côté serveur nécessite les droits accordés aux objets avec les deux profils IBM i QTMHHTTP et QTMHHTP1.

### 3. Profil utilisateur de type \*vldl

Il existe bien des profils utilisateurs et des mots de passe case sensitive (qui prennent en compte les majuscules et minuscules). Cependant il faut retenir absolument :

- a) Les profils internet ne gèrent pas les autorisations. Seuls QTMHHTTP et QTMHHTP1 gèrent les autorisations.
- b) Les profils internet ne permettent que d'entrer dans une session, rien de plus, rien de moins.

Si dans le fichier de configuration vous avez :

```
<LocationMatch /serverp/(.*)$>
AuthName "Application"
AuthType Basic
PasswdFile RPGINST/MYVLDL
Require valid-user
</LocationMatch>
```

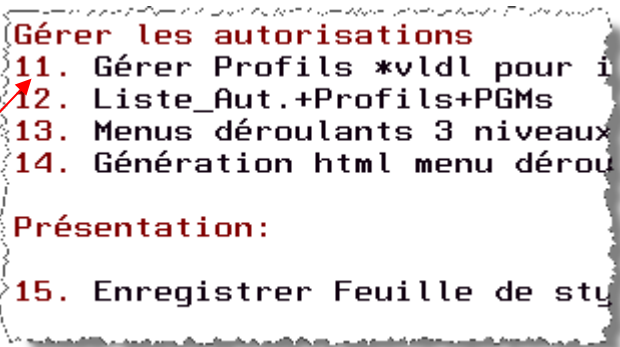
Cela signifie qu'à chaque fois qu'il y aura la directory /serverp/ au niveau de l'url, un profil utilisateur et un mot de passe seront requis « Require valid-user ». Ce profil et mot de passe cryptés sont stockés dans un objet \*vldl qui se trouve dans la bibliothèque RPGINST/MYVLDL.

L'objet liste de validation est créé automatiquement par RePeGlio avec la commande IBM i CRTVLDL. La liste de validation permet de gérer les profils internet.

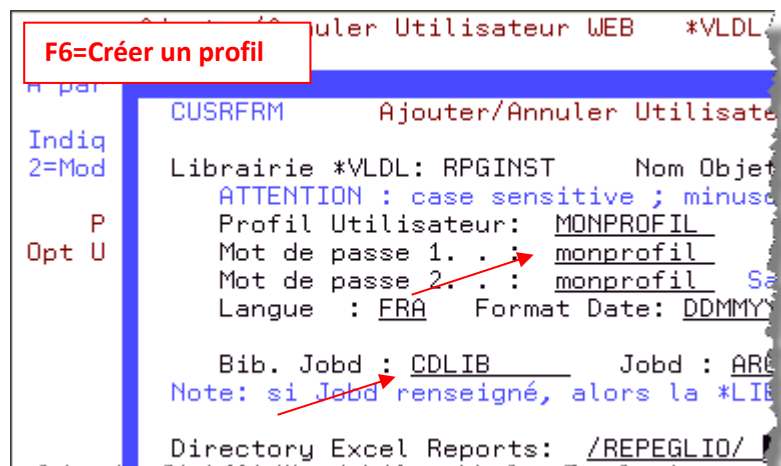
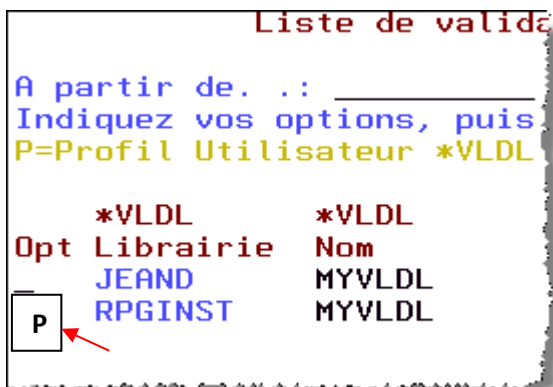
**Question fréquente** : Est-il possible d'utiliser les profils/mots de passe IBM i à la place des listes de validation ? Réponse courte oui. En fait, cette pratique est généralement déconseillée car le profil internet ne gère en aucun cas les autorisations au niveau des objets. Le profil internet permet seulement d'entrer via Apache dans une session Web et rien de plus. Donner des profils de type IBM i pour Apache est donc une pratique risquée car cela permettrait à l'utilisateur d'utiliser ce profil via Client Access.

Cependant, il est possible d'attribuer les mêmes profils et mots de passe internet volontairement et en toute connaissance de cause aux personnes qui utilisent le Web et Client Access, mais en faisant cependant attention à la casse.

4. Ajouter un profil \*vldl RePeGlio



Menu général RePeGlio ; /Web Paramètres/ puis Option 11 Gérer Profils \*vldl



**IMPORTANT** : Attribuer une Jobd de type IBM i au profil \*vldl. A l'intérieur de la Jobd, **mettre la liste des bibliothèque** des fichiers base de données avec CHGJOBd comme suit :

```
CHGJOBd JOBd(CDLIB/ARC_TEST)
        INLLIBL(ARC_USR ARC_OBJ QGPL ARC_DEMOFR JEUESSAI)
```

## 5. Gestion des bibliothèques avec Apache

**A retenir:** Apache ne connaît **qu'une seule** bibliothèque utilisateur qui est la bibliothèque de l'instance, celle incluse dans le fichier de configuration

ScriptAliasMatch /serverp/(.\*) /qsys.lib/RPGINST.lib/\$1.pgm

- a) **Liste des bibliothèques dans la Jobd du profil \*vldl** : Dans les programme RPG Web, à l'ouverture du fichier, dans le sous-programme SROPEN, le programme de service RLIBLJ recherche la première bibliothèque où se trouve un fichier PF ou LF dans la liste de la Jobd du profil \*vldl. Si la Jobd n'est pas renseignée au niveau du profil \*vldl, le programme RLIBLJ recherche l'autre alternative de la liste des bibliothèques par défaut pour l'instance.
- b) **Liste des bibliothèques par défaut pour l'instance** : depuis le menu déroulant /SEU\_WEB/ du menu général RePeGlio, option « WEB : Paramètres », puis option 1 « \*LIBL data base », il est possible de spécifier une liste de bibliothèque par défaut pour l'instance d'Apache si la Jobd n'est pas présente au niveau du profil \*vldl. Nous conseillons l'utilisation de la Jobd qui est beaucoup plus souple et plus fine.

Il faut retenir que dans la version 1 de RePeGlio nous ne disposons que de la liste par défaut au niveau de l'instance. Cette liste c'est révélée trop générale dans la pratique et nous avons donc donné la possibilité d'ajouter une Jobd avec une liste de bibliothèques au niveau plus fin du profil \*vldl.

## 6. Minimum requis : le profil et la liste des bibliothèques

Il faut retenir que le minimum requis pour appeler un programme Web est le profil et la liste des bibliothèques.

L'erreur la plus fréquente au début est le fichier non trouvé à l'ouverture faute d'une liste de bibliothèques pertinente.

## 7. Principe d'exécution d'un programme RPG ou \*pgm via Apache

Il faut retenir qu'Apache ouvre un travail au premier appel d'un programme RPG (voir ScriptAliasMatch du fichier de configuration).

Le travail ouvert par Apache reste ouvert et les autres programmes qui seront appelés par Apache le seront dans le même travail en principe.

- a) Conséquence 1 : QTEMP est partagé par tous, d'où la nécessité d'un profil \*vldl unique par utilisateur qui servira de clé majeure au fichier de travail éventuel.
- b) Conséquence 2 : gestion optimisée des temps de réponses dans la mesure où les fichiers sont ouverts dans le même travail avec un partage en mémoire de ces fichiers.

**8. Suppression du profil \*vldl pour accès direct sans profil:**

Il est possible de supprimer le profil \*vldl afin d’afficher directement la page Web sans avoir à saisir de profil. Dans ce cas, il convient de procéder de la façon suivante pour :

- Modifier le fichier de configuration
- Attribuer une liste de bibliothèques pour l’instance (et non la \*libl de la \*jobd du profil \*vldl)

Inconvénient :

- La gestion des menus déroulants et des autorisations ne sont pas gérés.

**Suivre pas à pas les étapes suivantes :**

➤ *Modifier le nom du fichier de configuration dans QUSRSYS/QATMHINSTC(instance) :*

Ouvrir une session avec le fichier QSECOFR. Sur la ligne de commandes :

**ADDLIBLE ARC\_OBJ**

**CALL APAWRK00**

Option 23 du menu : Mise à jour QUSRSYS/QATMHINSTC

Sélectionner votre instance qui est le membre du fichier +ROLLDOWN

Vous devez arriver avec UPDDTA sur une ligne de type :

-apache -d /www/JEAND -f conf/httpd.conf –AutoStartY

L’objectif est de modifier le nom du fichier de configuration lors du démarrage du serveur \*http

Changer le nom httpd.conf par httpdB.conf afin d’avoir la ligne :

-apache -d /www/JEAND -f conf/httpdB.conf –AutoStartY

Mettre à jour le membre qui correspond à l’instance.

➤ *Créer une copie du fichier de configuration pour l’instance avec le nouveau nom :*

Toujours sous QSECOFR, copier le fichier de configuration :

Option 5 du menu WRKLNK, sélectionner instance puis /www/JEAND/conf

(où JEAND est le nom de l’instance)

Option 3=copier en face du fichier httpd.conf **ET F4 invite**

Commande CPY :

Copier un objet (CPY)

Indiquez vos choix, puis appuyez sur ENTREE.

```
Objet . . . . . OBJ          > '/www/JEAND/conf/httpd.conf'
Répertoire de destination . . . TODIR      > '.'
Objet de destination . . . . . TOOBJ       '/www/JEAND/conf/httpdB.conf'
```

Vous devez avoir les deux lignes après copie:

Opt	Lien objet	Type	Attribut	Texte
	httpd.conf	STMF		
	httpdB.conf	STMF		

➤ *Suppression de l'objet \*vldl dans le fichier de configuration:*

Option 2 modifier en face de httpdB.conf et supprimer avec l'option 4 les lignes suivantes en fin du fichier de configuration:

```
.....
4 <LocationMatch /serverp/(.*)$>
4 AuthName "Application"
4 AuthType Basic
4 PasswdFile JEAND/MYVLDL
4 Require valid-user
4 </LocationMatch>
```

F3=Save/Exit

➤ *Arrêter et démarrer l'instance d'Apache :*

Option 2 ENDTCPVSR de l'instance \*http

Attendre la fin puis :

Option 1 STRTCPVSR de l'instance \*http

Maintenant, l'instance est démarrée avec les ordres du fichier de configuration httpB.conf et n'a donc plus de profil \*vldl donc d'invite utilisateur avec profil + mot de passe

➤ *Ajout de la liste de bibliothèques pour l'instance d'Apache :*

Il faut donc créer une liste de bibliothèques pour l'instance d'Apache.

Revenir au menu de développement RePeGlio.

Menu déroulant /SEU\_WEB/ option Web : Paramètres

Option 1 : \*LIBL data base

Option L devant l'instance

Ajouter les bibliothèques dans l'ordre des numéros (\*LIBL) avec par exemple :

Opt	Rang	Libr.	Libr.
		Nom	
	00020	ARC_DEMOFR	
	00100	ARC_OBJ	
	Etc...		

Si vos fichiers se trouvent dans ARC\_DEMOFR, la page s'ouvrira directement avec les enregistrements des fichiers de cette bibliothèque sans gestion du profil \*vldl.

Note : l'url du programme peut donc être appelée directement depuis une page web quelconque d'un autre serveur pour être intégrée à une application Web existante.