

# **Administration eDataCenter**

Copyright© 1994-2008, Apsynet S.A. Tous droits réservés.

Apsynet, Computer Inventory Manager, Cim Intranet Suite et CIM DataCenter sont des marques déposées de la société Apsynet; elles sont protégées par la loi française sur les droits d'auteur et par les clauses de divers traités internationaux.

Le logiciel et les manuels ne peuvent pas être copiés ou distribués, quel que soit le support, ni divulgués à des tierces parties ou utilisés d'une manière autre que ce qui est prévu dans le texte de la licence d'utilisation du logiciel compris avec le logiciel.

Cim DataCenter est une marque déposée de la société APSYNET.

Computer Inventory Manager est une marque déposée de la société APSYNET.

PC-DOS, OS/2 et LAN Server sont des marques déposées de la société International Business Machines Corporation.

MS-DOS et LAN Manager sont des marques déposées de la société Microsoft Corporation.

Windows, Windows 95, Windows 98, Windows Millenium, Windows NT , Windows 2000, Windows XP, Windows 2003 sont des marques commerciales de la société Microsoft Corporation.

Macintosh est une marque commerciale de la société Apple Computer Inc.

Novell et NetWare sont des marques déposées de la société Novell Inc.

VINES est une marque déposée de la société Banyan Systems Inc.

Excel est une marque commerciale de la société Microsoft Corporation.

Word est une marque commerciale de la société Microsoft Corporation.

Les autres marques énumérées dans ce guide sont des marques commerciales ou des marques déposées de leurs sociétés respectives.



## Sommaire

---

<b>Architecture</b>	<b>5</b>
<b>Administration du site</b>	<b>6</b>
« Utilisateur anonyme » : profil utilisateur par défaut	6
<b>URL d'interface</b>	<b>7</b>
Interface vers une machine	7
Interface vers un document	7
Interface vers un agenda	8
Méthode pour obtenir un agenda	8
<b>Information du Poste Client</b>	<b>9</b>
Accéder à la page d'information client	9
<b>Mot de passe oublié</b>	<b>10</b>
Les Droits sur les profils utilisateurs.	10
L'unification des utilisateurs et mail d'envoi	10
<b>Modification d'un portail léger</b>	<b>12</b>
Introduction	12
Associer des pages par défaut pour les utilisateurs non connectés	12
Modification du contenu en mode connecté / déconnecté	13
Liens vers l'application	13
Ajouter des valeurs par défaut dans une création de fiche	14
Liste des points d'entrée de l'application :	16
Bloc de Requête	17
Ajout d'un graphique	17
Ajout d'un bloc de news à un mini portail	18
Pré-requis	18
Démarche d'ajout :	19
<b>Mise en place de portails supplémentaires</b>	<b>21</b>
<b>Informations complémentaires</b>	<b>22</b>
<b>Annexes</b>	<b>23</b>
Table d'association des caractères HTML	23
Edition du fichier UAInfo.xml	23
Présentation	23
Structure d'un User Agent	24
Intérêt du fichier	25
Structure du fichier de dictionnaire	26
Exemples	27

---

# Architecture

L'architecture d'eDataCenter repose sur des pages HTML qui sont dynamiquement générées en fonction des demandes des utilisateurs (par l'extension ISAPI de eDataCenter).

Côté serveur Web, l'extension ISAPI soumettent des requêtes à la base de données, mettent en forme les résultats et envoient vers un navigateur Internet des pages au format HTML.

Côté client, l'utilisateur doit disposer d'un navigateur Internet (cf. pré requis dans la documentation d'installation) et pouvoir accéder à la page d'accueil d'eDataCenter.

# Administration du site

Cette section présente quelques mesures pour sécuriser votre site eDataCenter, les possibilités de paramétrage du fonctionnement et de gestion de l'extension ISAPI de eDataCenter.

---

## « Utilisateur anonyme » : profil utilisateur par défaut

eDataCenter est accessible de façon anonyme, c'est à dire sans que l'utilisateur ne s'authentifie : dans ce cas, le profil utilisateur utilisé est celui appelé « Utilisateur anonyme », qui est créé lors de l'installation de eDataCenter.

Il apparaît comme profil utilisateur de l'application et vous pouvez lui affecter différents droits (création/édition/lecture seule/aucun accès) sur chaque modèle de document.

# URL d'interface

---

## Interface vers une machine

Dans le cadre du développement d'une interface de consultation de la liste des machines, il est possible d'obtenir le détail d'une machine à partir de données spécifiques, grâce à une URL spécifique du site eDatacenter.

Voici les différentes méthodes pour afficher le détail d'une machine.

- Afficher une machine dont on connaît le config ID :  
`/Qrylsapi.dll?GetDetailMachine&config=xxxx`
- Afficher une machine en fonction de valeurs spécifiques : «  
`Qrylsapi.dll?GetDetailMachine&table=xxxx&column=yyyy&value=zzzzz` »

La variable config peut être paramétrée dans le fichier web.ini.

`SEARCH_TABLE = MACHINE`

`SEARCH_COL = COMPUTER_NAME`. Si ces paramètres sont absents la requête s'effectuera sur la table MACHINE et la colonne MAC\_CONFIG\_ID

 Exemple :

Obtenir la machine ayant pour MAC Address 0008C7292D7B

`Qrylsapi.dll?GetDetailMachine&table=MACHINE_NETWORK&column=NETWK_ADDR&value=0008C7292D7B`

Obtenir la machine ayant pour comme CHAMP MAT. 24 le numéro 128 12 47890

`Qrylsapi.dll?GetDetailMachine&table=HARDWARE_USER&column=USRFD_VALUE24&value=128%2012%2047890`

Obtenir la machine ayant pour N° de série AX-44444-01

`Qrylsapi.dll?GetDetailMachine&table=MACHINE&column=MAC_SN&value=AX-44444-01`

---

## Interface vers un document

Voici les différentes méthodes pour afficher un document.

- Afficher un document dont on connaît le document ID :  
`/Qrylsapi.dll?OpenSheet&DocId=xxxx`
- Afficher un document en fonction de valeurs spécifiques :  
«`Qrylsapi.dll?GetAPSDocument&ModelName=xxxx&FldName=yyyy&FldValues=zzzzz`»
  - ModelName : nom système du modèle
  - FldName : nom système du champ
  - FldValues : valeur du champ

🔗 Exemple :

Obtenir une fiche d'annuaire de M. Dupond

Qrylsapi.dll?GetAPSDocument&ModelName=Directory&FldName=Name&FldValues=D  
upond

---

## Interface vers un agenda

### Méthode pour obtenir un agenda

Qrylsapi.dll?MfcISAPICommand=OpenSheet&DocId=XXX&calendar=YYY


<https://support.apsynet.fr/apsyerp/Qrylsapi.dll?MfcISAPICommand=OpenSheet&DocId=5236244&calendar=group>

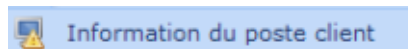
- DocId : identifiant de la fiche
- Calendar : Type d'affichage souhaité :
  - Group (affiche l'agenda de groupe)
  - Day (affiche l'agenda journalier)
  - Week (affiche l'agenda hebdomadaire)
  - Month (affiche l'agenda mensuel)
  - Year (affiche l'agenda annuel)



# Information du Poste Client

## Accéder à la page d'information client

Cette page permet de contrôler si le navigateur du client est compatible avec les pré-requis. Pour cela, il suffit d'utiliser le lien du menu : 



et une page de ce type apparaît :



### Information du poste client

Nom du Navigateur	Microsoft Internet Explorer
version	7.0
langue	fr
Cookie	Oui
IP vue du serveur	192.168.20.59
Utilisateur vu du serveur	
Nom du domaine	

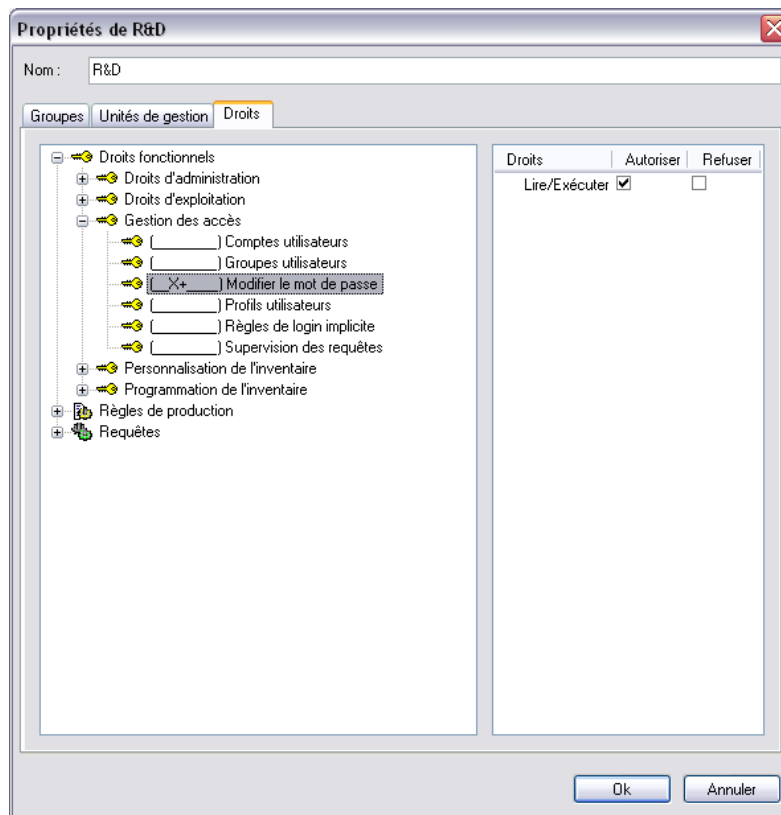
Exemple de configuration client.

# Mot de passe oublié

Un utilisateur ayant oublié son mot de passe peut demander à le régénérer. Pour cela il faut configurer un certain nombre d'éléments.

## Les Droits sur les profils utilisateurs.

Dans le Designer sélectionnez le profil pouvant demander à saisir un nouveau mot de passe.

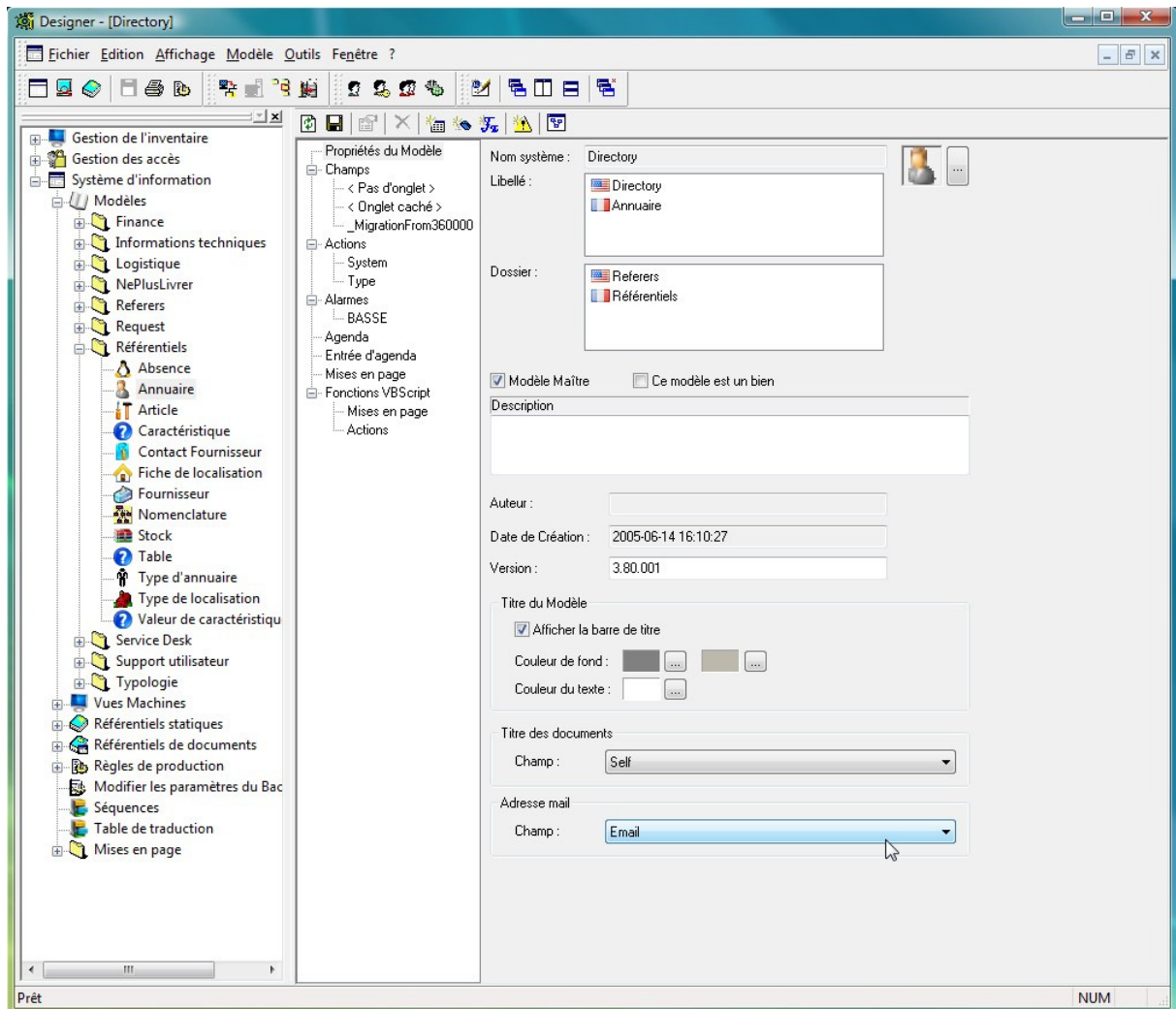


## L'unification des utilisateurs et mail d'envoi

Afin de retrouver le mail de la personne demandant un nouveau mot de passe, il faut un lien entre le login et la fiche utilisateur. Vérifier donc que le lien existe dans l'unification des utilisateurs.



Contrôlez que la fiche utilisateur possède un champ Email associé.

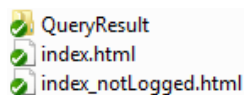


# Modification d'un portail léger

---

## Introduction

Un exemple de portail léger est contenu dans un répertoire à la racine du site web (votre site eDatacenter) et contient les fichiers suivants :



✍ Pour le moment un seul portail est livré, il s'agit du portail « mini » qui se situe dans le répertoire dashboard de eDataCenter.

Ce dossier contient les pages que l'on souhaite afficher.

Une page peut être de différents types :

- Soit il s'agit d'une page publique accessible pour n'importe quel utilisateur.
- Soit elle n'est visible que pour les utilisateurs connectés. (dans ce cas il faut définir une page par défaut comme nous le verrons dans le chapitre suivant)

Le dossier « QueryResult » contient les feuilles de styles qui serviront à faire le rendu du tableau de résultat.

---

## Associer des pages par défaut pour les utilisateurs non connectés

Il n'est pas nécessaire de déclarer une page Web dans le fichier ApsUrlMon.ini pour qu'elle soit visible, il suffit que cette page se situe sur le répertoire du portail. Attention si cette page n'est pas déclarée dans ce fichier de configuration, elle sera visible par les utilisateurs anonymes.

Pour obliger les utilisateurs à se connecter afin d'accéder à certaines pages, il faut tout d'abord définir une page qui contiendra le formulaire de connexion suivant :

```
<form action="qryisapi.dll" method="post">
  <input type="hidden" value="DisplayDashBoard" name="MfcISAPICommand" />
  <input type="hidden" value="{currentDir}" name="directory" />
  <input type="hidden" value="{file}" name="file" />
  <input type="hidden" value="{defaultFile}" name="defaultFile" />
  Utilisateur <input type="text" name="login" style="width: 70px;height:14px;" />
  Mot de passe<input type="password" name="pwd" style="width: 70px;height:14px;" />
  Mémoriser mes informations sur cet ordinateur
  <input type="checkbox" name="saveInCookie" />
```

```
<input type="image" src="../../img/connect.png" style="position: relative; top: 3px" />
</form>
```

Dans notre exemple une telle page existe et porte le nom « index\_notLogged.html »

Il faut ensuite éditer le fichier APSUrlMon.ini situé dans le répertoire racine du site internet de la manière suivante :

Dans la rubrique [defaultnotconnected] ajouter la ligne suivante :

```
Nomduportail/Monfichierprotege.html='Nomduportail\Monfichierpardefaut.html'
```

Par exemple si on souhaite rediriger le fichier index.html vers le fichier index\_notLogged.html dans le portail 'dashboard', il faudra ajouter la ligne suivante :

```
dashboard/index.html='dashboard\index_notLogged.html'
```

✎ Attention la première partie est une url, donc les séparateurs sont des « / » tandis que la seconde partie est un chemin windows dont les séparateurs seront des « \ »

✎ Pour tout nouveau portail il est fortement conseillé d'associer l'accès au répertoire à un fichier par défaut de la manière suivante : 'Nomduportail/=Nomduportail\Monfichierpardefaut.html'

---

## Modification du contenu en mode connecté / déconnecté

Il s'agit de fichiers HTML qui peuvent être édité par n'importe quel éditeur HTML.

Donc tous les outils HTML sont disponibles pour mettre en page ces fichiers.

Nous allons voir comment intégrer des éléments de l'application DataCenter.

### Liens vers l'application

Pour intégrer des éléments, il existe 2 méthodes, soit par le biais d'un lien soit par le biais d'un formulaire.

Attention dans le cadre d'un lien vers l'application, la référence Qrylsapi.dll devra nécessairement être précédée de « ../ ».

En effet les chemins des feuilles de styles et des fichiers de script des formulaires dépendent du répertoire relatif, si on saisit une url commençant par Qrylsapi.dll, le répertoire relatif sera « \$dossierInstallation/\$nomDuPortail/ » et alors aucun fichier de script ou de style ne sera chargé. Le lien ne donnera pas lieu à une erreur de « page non trouvée » mais la page qui sera alors affichée aura un comportement défaillant.

### Méthode lien

Cela donnera le rendu suivant :

<a href="#">Créer une nouvelle ligne de frais</a> <a href="#">Créer une nouvelle note de frais</a>
---

Pour créer un lien il faut le code HTML suivant<sup>1</sup> :

```
<a href=" ../Qrylsapi.dll?MfcISAPICommand=YYYY&ModId=XXXX"> Un lien</a><br />
```

- ✂ YYYY = Identification du point d'entrée (voir page suivante)
- XXX = Argument supplémentaire

Dans cet exemple il s'agit d'une création d'une nouvelle fiche donc YYYY et XXX correspondent respectivement à « NewSheet » et à l'identifiant du modèle de la fiche.

## Méthode formulaire

Cela donnera le rendu suivant :



Créer une nouvelle fiche

Modèle :

Pour un créer un formulaire, il suffit du code suivant :

```
<form method="get" action=" ../Qrylsapi.dll" >  
<h3>Créer une nouvelle fiche </h3>  
<input type="hidden" name="MfcISAPICommand" value="YYYY" />  
<label for='ModId'>Modèle :</label>  
<select name="ModId">  
<option value="XXXX">Note de frais</option>  
<option value="XXXX">Ligne de frais</option>  
</select>  
<input type="submit" value="Créer" />  
</form>
```

## Ajouter des valeurs par défaut dans une création de fiche

L'application peut désormais intégrer des valeurs par défaut lors d'une création de fiche.

Par exemple, si l'on souhaite proposer sur le portail un choix entre déposer un RTT, un congé ou un Arrêt maladie pour une création d'une nouvelle absence.



Créer un congé

Type de congé :

Date :

Il est possible de remplir n'importe quel champ de cette manière. D'un champ texte à un champ de référentiel dynamique. Nous verrons dans cette partie comment intégrer ces différents champs.

## Cas des champs de type Entier / String / Date

Il suffit de créer dans le formulaire un champ input avec le code suivant :

---

```
<input type="text" name="YYYYYY" />
```

YYYY correspond au nom de la colonne.



Ce nom de colonne est fourni dans le rapport de conception.

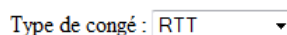
Dans notre exemple précédent cela donnerait pour le champ date la valeur suivante :

```
<input type="text" name="F_STARTDATE" />
```

✎ Si le format du champ ne correspond pas le champ ne sera *pas* défini par la valeur choisie. Par exemple si on saisit « texte d'exemple » dans le champ date précédent, la valeur ne sera pas défini lors de l'affichage de la fiche.

## Cas des champs de type Référentiel Statique

Dans notre exemple précédent, le choix du type d'absences est un référentiel statique.



Le code suivant suffit :

```
<select name="F_NATURE">
  <option value="RTT">RTT</option>
  <option value="CP congé payé">Congé Payé</option>
</select>
```

Il n'y aucune différence avec le cas précédent excepté les comportements suivants :

Si on saisit une valeur non présente dans le référentiel statique, si l'utilisateur dispose du droit de modifier les référentiels statiques associés au modèle sélectionné alors cette nouvelle valeur sera sélectionnée, dans le cas contraire la valeur sera une chaîne vide.



## Cas des champs de type Référentiel Dynamique

Le référentiel dynamique fonctionne de la même manière sauf qu'il faut fournir le DocId dans la valeur. Par exemple dans le cas précédent, le code suivant donnera le rendu précédent :

```
<input type="radio" name="F_NOMENCLATURE" value="324004" title="Graveur" />
Graveur (Matériel Informatique)<br />
<input type="radio" name="F_NOMENCLATURE" value="319625" title="Mobilier" />
Mobilier<br />
```

## Résumé de l'intégration de valeur par défaut par la méthode formulaire

Ainsi l'exemple suivant :



Se traduirait par le code HTML suivant :

```

<form method="post" action=" ../Qrylsapi.dll" >
  <h3>Créer un congé</h3>
  <input type="hidden" name="MfcISAPICommand" value="NewSheet" />
  <input type="hidden" name="ModId" value="1020042797" />
  <label for='F_NATURE'>Type de congé : </label>
  <select name="F_NATURE">
    <option value="RTT">RTT</option>
    <option value="CP congé payé">Congé Payé</option>
  </select>
  <br />
  <label for='F_STARTDATE'> Date : </label>
  <input type="text" name="F_STARTDATE" />
  <br />
  <input type="submit" value="Créer" />
</form>

```

### Créer des Url avec des champs pré-remplis

Il est possible de réaliser des URL vers des fiches pré-remplis directement, l'url devra avoir la forme suivante :

```

../Qrylsapi.dll?MfcISAPICommand=NewSheet&ModId=XXXX&YYYY1=ZZZZ1&YYYY2=ZZZZ2

```

Ou XXXX correspond à l'id du modèle,

Et pour chaque champ que l'on souhaite définir à l'avance YYYYn correspond au nom de la colonne et ZZZZn correspond à la valeur que l'on souhaite définir.

Par exemple, si on souhaite créer une nouvelle fiche d'article de type mobilier, il faudra fournir le lien suivant :

```

../Qrylsapi.dll?MfcISAPICommand=NewSheet&ModId=269625473&F_NOMENCLATURE=319625

```

ZZZZn ne doit pas contenir les caractères suivants 'espace' '&' '=' '(' ')' '%' qui devront être remplacés. (Voir l'Annexe 1 pour la table d'association). Par exemple, dans le cas des congés si on souhaite faire un lien vers une fiche d'absence ou le type d'absence est renseigné à Congé payé, il faut utiliser l'URL suivante :

```

../Qrylsapi.dll?MfcISAPICommand=NewSheet&ModId=1020042797&F_NATURE=CP%20cong  %20pay  

```

La taille d'une URL est limitée à 2046 caractères.

### Liste des points d'entrée de l'application :

Description du point :	Identification :	Arguments supplémentaires requis:
Création d'une nouvelle fiche	NewSheet	ModId : L'identifiant du modèle de fiche
Ouverture d'une fiche	OpenSheet	DocId : Référence de la fiche
Ouverture d'une requête	Execute	QryID : Identifiant de la requête
Ouverture d'un graphique	ExecuteQryGraph	QryID : Identifiant de la requête
Exécution d'une règle de production	ExecProd	PR_ID : Identifiant de la règle de production Docidlist : docid docid ...



---

## Bloc de Requête

Les blocs de requêtes sont des fichiers XML générés par le serveur qui sont rendus par une feuille de style.

Pour modifier le rendu d'un bloc requête il suffit de modifier la feuille de style associée.

Une feuille de style est fournie avec l'application il s'agit de « QueryResult/Query.xsl ».

Code	Cumul des contrats	Cumul des coûts
RD07	0	0
loc2008	0	0
mkt2008	0	0
info2008	0	0
RD2008	0	0
bur2008	0	0
SOC2008	0	0
webex08	0	0

Cette feuille de style fournira le rendu suivant :

Pour pouvoir intégrer Ce bloc de requête dans l'application, il suffit d'intégrer le code HTML suivant :

```
<object type="text/xml"
data="Qrylsapi.dll?MfclSAPICommand=ExecQueryLight&QryID=XXXXXXXXXX&xsl=
QueryResult/Query.xsl"></object>
```

✎ XXXX est l'identifiant de la requête.

On remarque ici que l'on rajoute la référence de la feuille de style par le biais de l'argument xsl.

Faire attention : l'url de la feuille de style est relative à l'url à partir de laquelle vous appelez Qrylsapi.dll.

Par exemple si vous saisissez une adresse du type « ../Qrylsapi.dll ?.....xsl=\$xsl\_dir/\$xslfile », la feuille devra être située sur « \$repInstallation/\$xsl\_dir/\$xslfile ».

Alors que si l'url saisie est « Qrylsapi.dll ?.....xsl=\$xsl\_dir/\$xslfile », la feuille de style devra être localisée sur « \$repInstallation/\$repPortail/\$xsl\_dir/\$xslfile ».

✎ Si le chemin vers la feuille de style est mauvais, Internet Explorer n'affichera pas la page du portail.

---

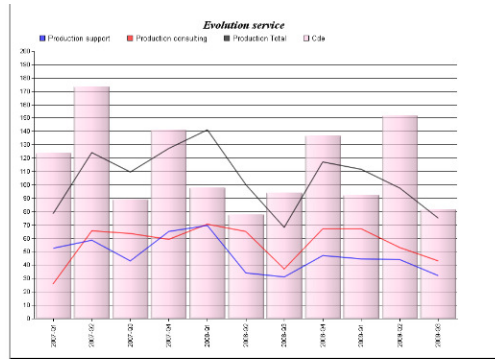
## Ajout d'un graphique

Un graphique est une image qui se récupère par le biais de l'url suivante :

[Qrylsapi.dll?MfclSAPICommand=ExecuteQryGraph&QryID=XXXXXX](#)

Où XXXX est l'identifiant de la requête.

Ainsi le graphique suivant :



Peut s'obtenir par le code HTML suivant :

```

```

Où XXX est l'identifiant de la requête, aaaa et bbbb la largeur et la hauteur de l'image calculée

Par ailleurs il est possible de combiner une image vers un lien vers l'application. Dans cet exemple précis, on peut vouloir souhaiter faire un lien vers la requête si on clique sur l'image. Cela donnera le code suivant (voir les détails des points d'entrée vers l'application dans la documentation associée) :

```
<a href=" ../Qrylsapi.dll?MfcISAPICommand=Execute&QryID=XXXX" >

</a>
```

Si on veut rendre les barres d'un graphique cliquables individuellement on peut remplacer le code ci-dessus par celui-ci :

```
<iframe height="200"
src="Qrylsapi.dll?MfcISAPICommand=ExecuteQryGraphFrame&QryID=XXX&width=2
69&height=200" title="Planning" alt="Planning" style="width: 269px;" />
```

Dans ce cas ce n'est plus la requete source qui est ouverte mais bien l'élément qui correspond à la barre, par exemple une opération dans un graphique qui liste des entrées dans un agenda.

## Ajout d'un bloc de news à un mini portail

### Pré-requis

Il faut l'identifiant du bloc de news qui s'obtient de la manière suivante sur le portail lourd (uniquement en Web) :

**Informations produit** 100

Bienvenue sur le portail d'assistance client Apsynet, si vous ne disposez pas des informations de votre compte de connexion, contactez le support Apsynet à support@apsynet.fr

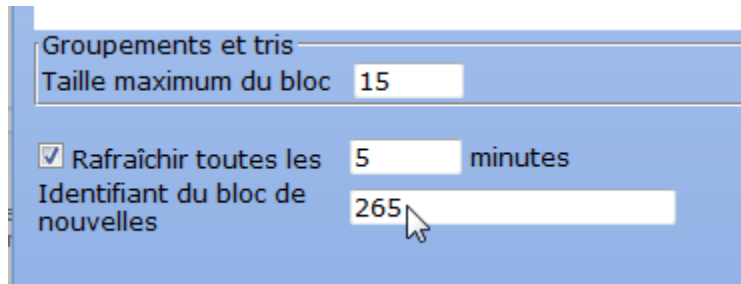
La version 3.4 est disponible  
Pour recevoir le CD ROM, faites une demande à : contacts@apsynet.fr

Editer la re

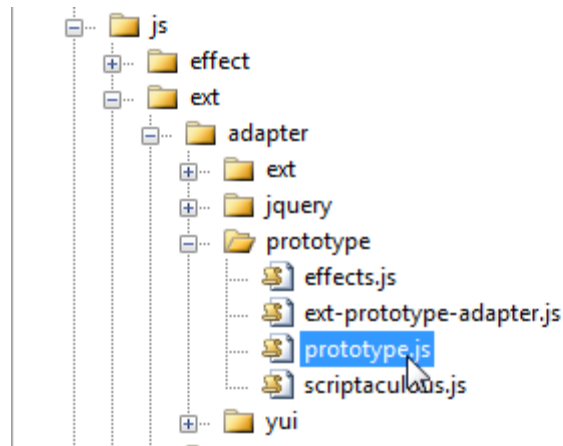
Partager...

Supprimer

Propriétés



Il faut recopier le fichier Javascript suivant à la racine du répertoire de votre mini-portal :



Et l'inclure dans votre page HTML par le biais de la ligne suivante (positionné dans la balise Head du site) :

```
<script type="text/javascript" src="prototype.js"></script>
```

Enfin il faut Ajouter les lignes suivantes dans la balise Head de votre page Web :

```
<script type="text/javascript">
    function printNews(area, id) {
        new Ajax.Request('QryIsapi.dll', {
            method: 'post', parameters: {MfcISAPICommand:
'LoadCdcPageDailyNewsEx', blockId: id},
            onSuccess: function(result) {$(area).innerHTML =
result.responseText.replace(/QryIsapi\.dll/gi,
'../QryIsapi.dll');}
        });
    }
</script>
```

## Démarche d'ajout :

### Création de la zone d'affichage :

Il faut dans un premier créer une zone dans lequel le contenu sera affiché dans votre page Web :

```
<span id="news1"></span>
```

### Récupération du contenu du bloc de nouvelles :

Ensuite il faut ajouter le bloc suivant à la fin de votre page Web, (juste avant la balise </body>)

```
<script type="text/javascript">
  printNews('news1', 449);
  printNews('news2', 264);
</script>
```

Le premier argument est l'identifiant de la zone dans laquelle vous voulez ajouter le contenu du bloc de news.

Le second argument est l'identifiant du bloc de news (cf. pré-requis pour le récupérer).

Il est possible d'ajouter plusieurs blocs de news dans une même page.

# Mise en place de portails supplémentaires

Pour ajouter un portail supplémentaire, les étapes suivantes sont nécessaires :

- 1) Copier le contenu du répertoire de dashboard dans un nouveau répertoire à la racine du site
- 2) Modifier les fichiers html à la racine de ce site pour selon les besoins du nouveau portail.
- 3) Editer le fichier APSUrIMon.ini qui est situé dans le répertoire racine du site internet de la manière suivante (*ne pas mettre en majuscules*) :

Dans la rubrique [enableddashboard], ajouter la ligne  
Nomdurepertoire=1

# Informations complémentaires

La connexion est valable pour l'ensemble de l'application ainsi il est possible de créer un portail complexe avec plusieurs pages. Il suffit de créer un portail pour chaque nouvelle page.

Il est possible qu'il existe plusieurs répertoires versions de l'application sur un même site, il faut néanmoins l'indiquer dans le paramétrage du filtre ISAPI.

En effet le filtre est commun à tous les sites. Si plusieurs eDatacenter sont installés, le filtre ne fonctionnera alors que sur le répertoire où il est installé.

Pour pallier ce problème, il existe un moyen de ne pas utiliser les chemins relatifs à l'emplacement du filtre mais des chemins absolus sur le serveur.

Voici la méthodologie pour permettre d'utiliser ce filtre avec des chemins absolus :


Dans la rubrique `GeneralSettings` (après la balise `[general]`), ajouter la ligne suivante :

```
pathalias_absolute=1
```

Dans la rubrique `PathAlias` (après la balise `[pathalias]`), Ajouter la ligne suivante :

```
Adresse_physique_du_site\Nom_du_portail\qryisapi.dll='Adresse_physique_du_site\qryisapi.dll'
```

Effectuer cette manipulation pour chaque portail et pour chaque site.

 Ce type de configuration ne peut pas être utilisé de manière générique.

# Annexes

## Table d'association des caractères HTML

Character	Code Points (Hex)	Code	Why encode?
Space	20	%20	Significant sequences of spaces may be lost in some uses (especially multiple spaces)
Quotation marks 'Less Than' symbol ("<") 'Greater Than' symbol (">")	22 3C 3E	%22 %3C %3E	These characters are often used to delimit URLs in plain text.
'Pound' character ("#")	23	%23	This is used in URLs to indicate where a fragment identifier (bookmarks/anchors in HTML) begins.
Percent character ("%")	25	%25	This is used to URL encode/escape other characters, so it should itself also be encoded.
Misc. characters: Left Curly Brace ("{") Right Curly Brace ("}") Vertical Bar/Pipe (" ") Backslash ("\") Caret ("^") Tilde ("~") Left Square Bracket ("[") Right Square Bracket ("]") Grave Accent ("`")	7B 7D 7C 5C 5E 7E 5B 5D 60	%7B %7D %7C %5C %5E %7E %5B %5D %60	Some systems can possibly modify these characters.

## Edition du fichier UAInfo.xml

### Présentation

La plupart des navigateurs diffuse des informations concernant la plate-forme de l'utilisateur.

Ces informations sont envoyées sur la forme d'une chaîne normalisée qui est la chaîne User Agent.

Cela permet :

- De connaître le navigateur de l'utilisateur,
- D'identifier le type de plate-forme de l'utilisateur,
- De connaître les plugins installés sur le navigateur qu'il est en train d'utiliser.

Dans le produit Datacenter il nous permet d'identifier si l'utilisateur utilise un mobile ou non.

Cependant les informations sur l'utilisateur dépendent directement du navigateur et elles ne sont pas exhaustives.

Par exemple Opéra mobile ne renvoie pas la résolution du navigateur.

Toute chaîne peut être récupérée sur le site [www.apsynet.fr/test\\_ua.php](http://www.apsynet.fr/test_ua.php)

Il suffit d'aller sur cette page avec le navigateur que l'on souhaite tester.

## Structure d'un User Agent

La chaîne d'User Agent est une liste d'agents séparés par des espaces qui sont de la forme suivante :

AGENT\_NAME[VERSION] [TAG\_LIST]

### AGENT\_NAME

Il s'agit du nom de l'agent, il ne contient pas les caractères suivants : ESPACE, '/'

Il s'agit d'une spécification d'informations sur le navigateur, parfois il s'agit du navigateur mais ce n'est pas toujours le cas.

Exemple :

Mozilla est un Agent et NON un navigateur, en effet la chaîne d'User Agent de Internet Explorer est

```
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0; SLCC1; .NET CLR 2.0.50727; .NET CLR 3.5.30729; InfoPath.2; .NET CLR 3.0.30729; .NET4.0C; .NET4.0E)
```

Tandis que celle de Firefox est :

```
Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.2.4) Gecko/20100413 Firefox/3.6.4 (.NET CLR 3.5.30729; .NET4.0E)
```

On remarque que l'information du navigateur (surlignée en jaune) est dans le premier cas un TAG (voir plus bas) et dans un autre cas il s'agit d'un autre Agent.

### VERSION

Facultatif, toujours sous la forme d'un / suivi d'une série de nombre éventuellement séparés par des points.

### TAG\_LIST

La liste est facultative, et se présente sous la forme suivante :

( [ TAG; ] \* [ TAG ] )

Il s'agit d'une liste de Tags séparés par des ';'.

Un TAG est une chaîne de caractère ne contenant pas les caractères ';', ')', '('.

Il peut être accessoirement accompagné d'une série d'informations entre parenthèse.

En général, ces informations ne sont pas pertinentes (sauf dans le cas du framework .NET par exemple) et elles sont inutiles pour eDatacenter, elles sont donc ignorées.

L'information du Tag n'est pertinente que relativement à l'agent qui l'utilise et à la version de ce dernier.



## Intérêt du fichier

Le fichier UAInfo.xml, permet d'associer une liste de propriétés à un TAG.

Toutes les propriétés ne seront pas obligatoires pour un TAG, par ailleurs il est possible qu'il en ait aucune, dans ce cas il sera ignoré par le système.

Pour rappel, chaque Tag est associé à un Agent (et éventuellement une version de cet Agent).

Pour l'instant la seule propriété existante est la suivante :

### Ismobile

- Identifie si l'utilisateur utilise un appareil mobile pour se connecter à l'application.
- Peut être récupérée dans les scripts ainsi :  
`CdcCurrentUser.properties.IsMobile` dans le code VB.

## Structure du fichier de dictionnaire

Il s'agit d'un arbre arborescent à trois niveau : les AGENTS, les VERSIONS, les TAGS (voire définitions plus haut) et enfin les informations.

Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<agentlist>
  <agent value="Opera" type="single" >
    <version type="single" value="10.1">
      <tag value="Windows Mobile" type="single" >
        <ismobile>true</ismobile>
      </tag>
    </version>
  </agent>
  <agent type="default">
    <version type="default">
      <tag type="default">
        <ismobile>>false</ismobile>
      </tag>
    </version>
  </agent>
</agentlist>
```

Chaque balise peut être de deux types : « default » et « single »

- Une balise de type « single » possède nécessairement une valeur, elle permet de sélectionner le type d'agent.
- Si on ne trouve aucun agent on essaie de se reporter sur la valeur par défaut qui est la balise de type « default ».

Si on ne trouve pas de valeur par défaut d'une version par exemple on se reporte sur un niveau précédent. Par exemple, dans ce cas ci, il ira chercher l'information dans l'agent par défaut.

Dans cet exemple, sur les agents qui sont de types opéra avec une version égale à 10.1 on ajoutera l'attribut « ismobile » à l'utilisateur si cet agent possède le tag « Windows Mobile ».

Néanmoins on voudrait que pour toutes les versions d'Opéra cela fonctionne ainsi.

Donc il suffit de changer la ligne suivante :

```
<version type="single" value="10.1">
Par la ligne :
<version type="default">
```

✎ En cas de conflits, la position la plus avancée sur la chaîne de l'User Agent est prioritaire. (L'information qui arrive en premier dans le sens de lecture classique est privilégiée)

✎ Il est conseillé d'utiliser les versions par défaut pour des raisons de maintenance.

✎ Ce fichier est sensible à la casse donc il faut faire attention aux majuscules.

## Exemples

### Ajout d'un Tag

On voudrait considérer comme utilisateur mobile un utilisateur d'Opéra sur Iphone.

Il possède l'User Agent suivant :

```
Opera/9.80 (iPhone; Opera Mini/5.0.019802/18.738; U; fr) Presto/2.4.15
```

Le Tag pertinent (surligné) se situe sur l'agent Opera.

Donc on veut ajouter un Tag pour la version d'opéra par défaut qui est Iphone et indiquer qu'il est mobile.

```
<agentlist>
  <agent value="Opera" type="single" >
    <version type="default">
      <tag value="iPhone" type="single" >
        <ismobile>true</ismobile>
      </tag>
      <tag value="Windows Mobile" type="single" >
        <ismobile>true</ismobile>
      </tag>
    </version>
  </agent>
  <agent type="default">
    <version type="default">
      <tag type="default">
        <ismobile>>false</ismobile>
      </tag>
    </version>
  </agent>
</agentlist>
```