



IBM Power Systems - IBM i

Modernisation, développement d'applications et DB2 sous IBM i
Technologies, outils et nouveautés 2013-2014

13 et 14 mai 2014 – IBM Client Center Paris, Bois-Colombes

S22 – PHP sous IBM i : accédez à vos données et applications avec le PHP Toolkit

Mercredi 14 mai – 13h30-15h00

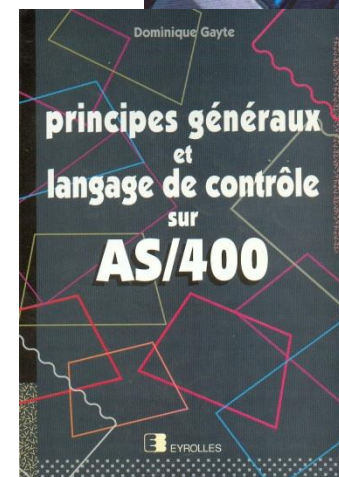
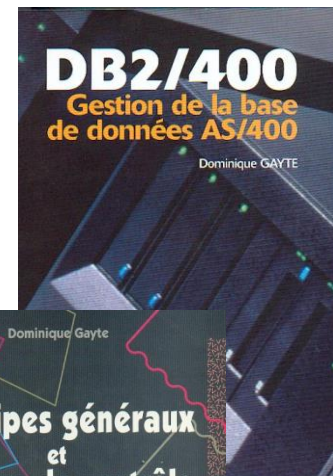
DUMAS Gautier

NoToS

- Expertise autour de l'IBM i
 - Plus de 20 ans d'expérience sur AS/400
 - Regard moderne
- PHP sur IBM i avec Zend
 - Depuis 6 ans
- Développement de progiciels



- Fourni des solutions et des services autour de PHP et de Zend



Zend Server IBM i

- Partenariat fort avec IBM depuis 2005

- Stack PHP sur IBM i aujourd'hui en version 6 (6.2 6.3)

- Mode de licence
 - Gratuit (Free Edition)
 - Entreprise

- Détails des différences :
 - Disponible sur <http://www.zend.com/fr/products/server/editions-ibm-i>

Installation Zend Server

- Disponible en téléchargement sur :

<http://www.zend.com/fr/products/server/downloads>



- Environnement de développement :

- Zend Studio

- <http://www.zend.com/fr/products/studio/downloads>

Qu'est ce que le Toolkit ?

- Ensemble de classes PHP pour appeler les objets natifs IBM i
- Les programmes
 - RPG, COBOL, CL ...
- Autres :
 - Commandes IBM i
 - Spools, Job logs, Active jobs, Valeurs système
 - Data Area, User Spaces
 - ...
- Accès naturel depuis le code PHP

Historique du ToolKit

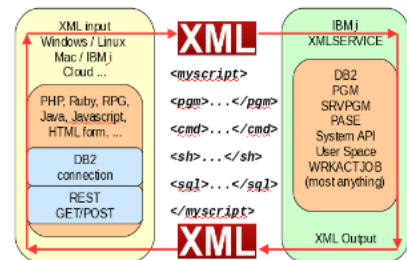
- Connecteur i5 EasyCom
- Zend propose le Toolkit en partenariat avec IBM
- Compatibility Wrapper (CW)
- Maintenant, le Toolkit est utilisable depuis une plateforme Windows ou Linux pour accéder aux objets de l'IBM i

Installation et mise à jour

- A partir de la V5R4
- <http://www.youngiprofessionals.com/wiki/XMLSERVICE>

View Edit History Print

XMLSERVICE



XMLSERVICE is a single library of Open Source RPG code that enables XML scripting calls of System i resources using most any language available on your platform.

XMLSERVICE RPG server library does not require other language products to run on your IBM i, however language teams often provide a client toolkit to greatly simplify XML calls to XMLSERVICE. Language vendors often provide compiled XMLSERVICE distributed with product installations (Zend Server, PowerRuby).

- PTF pour IBM i
 - HTTP
 - Java (PASE)
 - DB2 (dernier groupe PTF) ou au minimum :
 - SI39831/SI39917 pour la V7R1
 - SI39829 pour la V6R1
 - SI39610 pour la V5R4

Définir un CCSID

- **CCSID : Coded Character Set Identifier**
 - Détermine les conversions entre EBCDIC et ASCII
 - Valeur spéciale 65535 :
 - Signifie : pas de conversion
 - N'est pas une bonne solution mais c'est la valeur par défaut
 - Tout le monde doit définir son CCSID
 - CCSID pour la France : 297

- Pas seulement pour le Toolkit
 - Beaucoup d'opérations DB2 exige un CCSID

Vérifier son CCSID

- **DSPSYSVAL QCCSID**
 - Ne doit pas être 65535
 - Utiliser la commande CHGSYSVAL SYSVAL(QCCSID) VALUE(297)
 - A faire sous contrôle

- Ou dans le fichier de configuration Apache :
/www/zendsvr/conf/httpd.conf
 - Configuration pour la France
 - DefaultFsCCSID 297
 - CGIJobCCSID 297

- **Tout le monde doit définir un CCSID**
 - Sinon le ToolKit (et pas seulement) ne fonctionne pas !

Vérification de l'installation

- Après installation du Zend Server et du ToolKit

http://ip_ibm_i:10088/Samples/Toolkit/

New PHP Toolkit Demo Scripts

The new Toolkit technology is based on an open source XML transport developed by IBM and Zend, which allows you to extend PHP code to access IBM i resources. The Toolkit XML service protocol is designed to work with "state full" RPG/CL programs (many open database files and other system resources), so a developer won't have to rewrite any programs just to get on the web. Clicking on the script name will run the code and clicking on the display code will display the PHP code. The scripts location is `/www/zendsvr/htdocs/NewToolkit`.

- [Run Display Library \(DSPLIBL\) command](#) [Display Code](#)
- [Program Call with two parameters](#) [Display Code](#)
- [Program Call with Data structure parameters](#) [Display Code](#)
- [Service Program call that returns function value](#) [Display Code](#)

Additional code samples

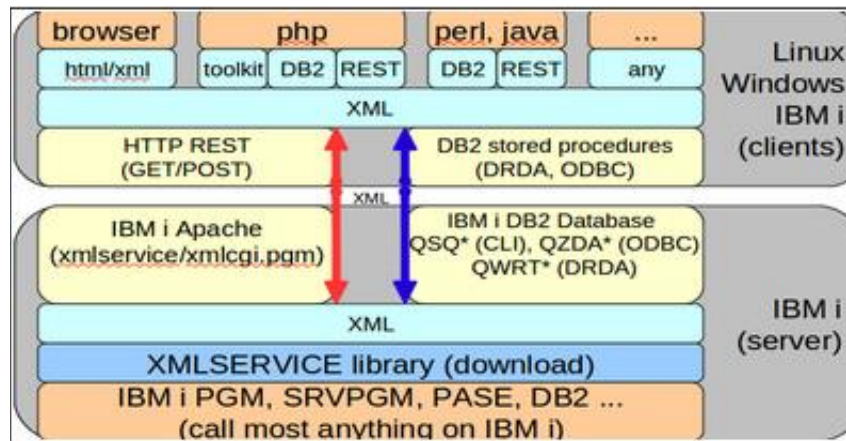
- Work with Spooled File entries [Display Code](#)
- Work with Data Queue (keyed) [Display Code](#)
- Work with User Space [Display Code](#)
- Work with Objects [Display Code](#)
- Work with Jobs [Display Code](#)

© 2011 Zend Technologies Ltd. All rights reserved.

XML Service

- Premier projet Open Source sur IBM i
- Composé de RPG, CL et procédures stockées

Source : Tony Cairns, IBM Rochester



Gratuit et Open Source

- Licence BSD Berkeley Software Distribution
 - http://fr.wikipedia.org/wiki/Licence_BSD
 - Licence libre pour la distribution de logiciels
- Code source disponible pour tous
 - Les contributions sont bienvenues
- Support IBM
- Pas d'enregistrement ou de licence
- Pas de message intempestif
 - (exemple : QSYSOPR)

Portable, accessible par d'autres langages

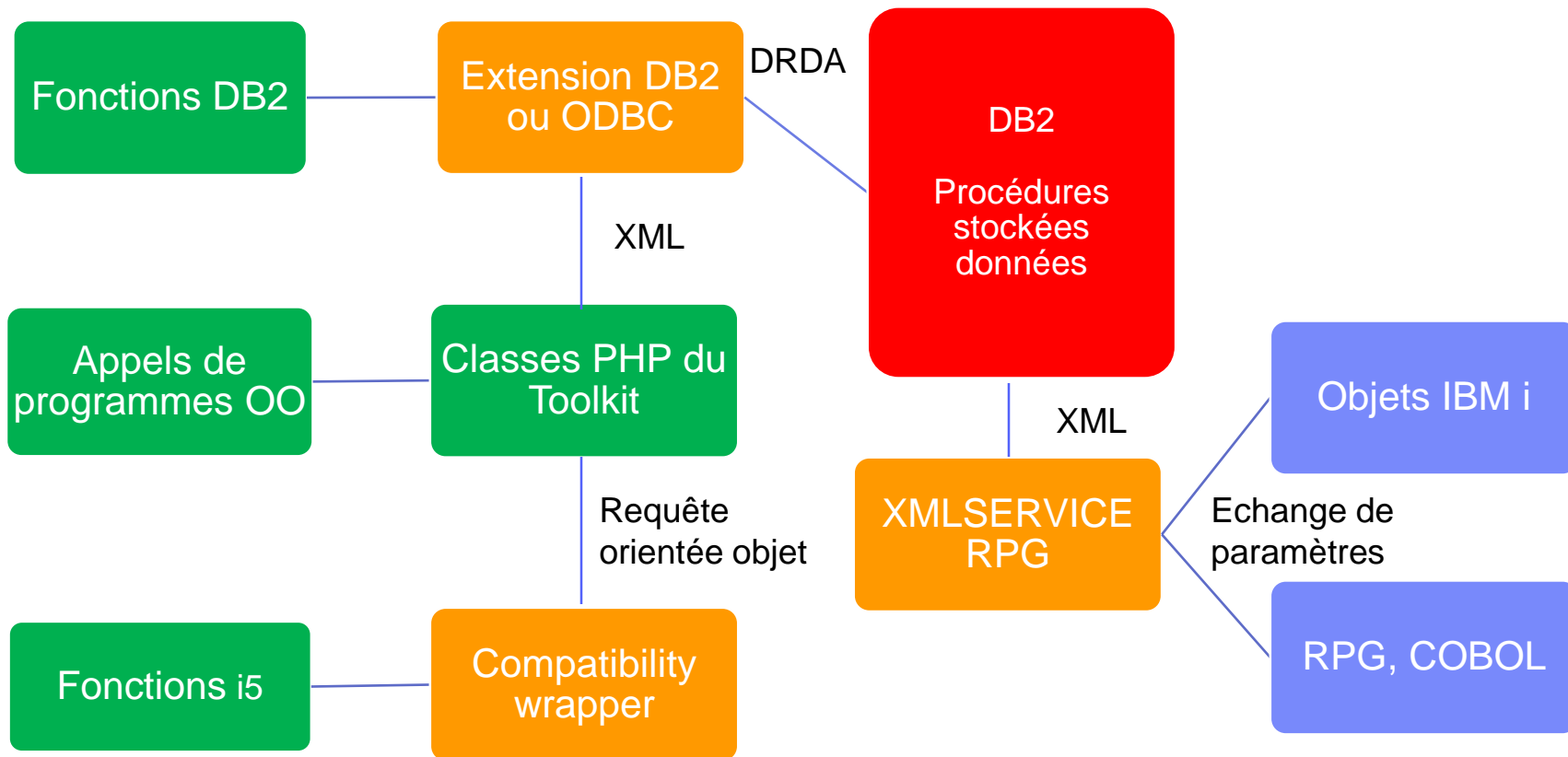
- Architecture 2-Tiers
 - Windows
 - Linux
 - Mac

- Connexion ODBC ou DB2
 - Avec DB2 Connect d'IBM (payant)
 - ODBC : gratuit avec IBM i Access

- Même code pour exécuter les programmes

- Configuration du paramètre \$extension de get_instance

Fonctionnement global



Focus sur le compatibility wrapper

- Intercepte les fonctions i5 existantes
- Ecrit entièrement en PHP
- Langage Orienté Objet (fonctions i5 en code procédural)
- Utilisation de XML Service

Mode opératoire de changement de l'existant

- Inclure la classe dans les scripts
 - `Require_once('CW/cw.php');`
- Désactiver l'extension Easycom
- Redémarrer Apache
- Vérifier le nom d'hôte de la connexion :
 - Ancien ToolKit : Le nom d'hôte
 - Nouveau ToolKit : Le nom de la base de données (*LOCAL)

- NB : Nom de la base de données dans le WKRRDBDIRE

Pour bien commencer avec le Toolkit

- C'est une classe qui va souvent servir dans les projets IBM i
- Intérêt de le mettre dans `include_path` pour pouvoir l'inclure facilement sur tous les scripts qui vont l'utiliser

/usr/local/zendsvr/share/ToolkitAPI

- Ce qui permet de mettre dans tous les scripts :

Include_once 'ToolkitService.php';

La connexion

- Utilisation de la méthode `getInstance(...)`

```
21 $extension='ibm_db2';
22 try {
23     $ToolkitServiceObj = ToolkitService::getInstance($db, $user, $pass, $extension);
24 }
25
26 catch (Exception $e)
27 {
28     echo $e->getMessage(), "\n";
29     exit();
30 }
31
32
33 $ToolkitServiceObj->setToolkitServiceParams(array('InternalKey'=>"/tmp/$user"));
34 $code = $_POST ['code'];
35 $desc = ' ';
```

Connexion Stateless ou Stateful

- Facile pour commencer : stateless
 - Connexion plus rapide, exécution dans les JOB DB2 (QSQSRVR)
 - N'assure pas de travailler dans le même JOB
 - Mot clé 'stateless'

```
$conn=>setToolkitServiceParams(array('stateless'=>>true));
```

Connexion Stateful

- Configuration avancée : stateful/private
 - Un nouveau JOB XMLTOOLKIT est démarré et réservé
 - Conserve un contexte QTEMP, LIBL, open file
 - Meilleures performances
 - Mot clé 'InternalKey' comme identifiant unique du JOB

```
$conn=>setToolkitServiceParams(array('InternalKey=>'/tmp/gdu123'));
```

Options de connexions : plugSize

- Configurations du moins performant au plus performant

- `$ToolkitServiceObj->setToolkitServiceParams(array('InternalKey'=>'/tmp/users'.rand(1,$maxpool), 'plugSize' => '15M'));`

- `$ToolkitServiceObj->setToolkitServiceParams(array('InternalKey'=>'/tmp/users'.rand(1,$maxpool), 'plugSize'=>'512K'));`

- `$ToolkitServiceObj->setToolkitServiceParams(array('InternalKey'=>'/tmp/users'.rand(1,$maxpool), 'plugSize'=>'4K'));`

Option plug size

- Une connexion DB2 qui assure un transport de données XML sans problème
- Un dimensionnement dû à l'utilisation des paramètres des procédures stockées

Appel de programmes : paramètre CHAR

AddParameterChar(\$io, \$size, \$comment, \$varName, \$value)

\$io : IN, OUT, BOTH

\$size : longueur du paramètre

\$comment : Commentaire sur le paramètre

\$varName : Nom du paramètre

\$value : Valeur à passer

```
37 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10, 'CODE', 'CODE', $code);
38 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10, 'DESC', 'DESC', $desc);
39
40 $result = $ToolkitServiceObj->PgmCall("COMMONPGM", "ZENDSVR", $param, null, null);
41
42 if($result){
43     showTable( $result['io_param']);
44 }
45 else
46     echo "Execution failed.";
```

L'appel de programmes : les paramètres

- Les différents paramètres possibles :
 - AddParameterInt32(\$io, \$comment, \$varName, \$value, \$dimension)
 - AddParameterInt64(\$io, \$comment, \$varName, \$value, \$dimension)
 - AddParameterUInt32(\$io, \$comment, \$varName, \$value, \$dimension)
 - AddParameterUInt64(\$io, \$comment, \$varName, \$value, \$dimension)
 - AddParameterFloat (\$io, \$comment, \$varName, \$value, \$dimension)

L'appel de programme : les paramètres

- Les différents paramètres possibles :
 - AddParameterReal(\$io, \$comment, \$varName, \$value, \$dimension)
 - AddParameterPackdec(\$io, \$length, \$scale, \$comment, \$varName, \$value, \$dimension)
 - AddParameterZoned(\$io, \$length, \$scale, \$comment, \$varName, \$value, \$dimension)
 - AddParameterBin(\$io, \$size, \$comment, \$varName, \$value, \$dimension)
 - AddDataStruct(\$array, \$parameters, \$name='struct_name', \$dim, \$by, \$isArray, \$labelLen)

L'appel de programme

- Méthode `pgmCall()`
 - `$Program` : Nom du programme
 - `$Library` : Nom de la bibliothèque
 - `$Parameters` : Tableau de paramètres
 - `$ReturnValue` : Valeur de retour
 - `$options` : Tableau d'options

Les commandes

- Intérêt : récupération des paramètres
- **\$cnx->CLCommand('my command');**
 - ➔ Retourne un booléen
- **\$cnx->CLInteractiveCommand('DSP...command');**
 - ➔ Retourne l'écran de sortie de la commande
- **\$cnx->CLCommandWithOutput('RTV... command');**
 - ➔ Retourne des paramètres retours de la commande

Exemple (PHP)

```
23 $cmd = "addlib ZENDSVR";
24 $ToolkitServiceObj->CLCommand($cmd);
25
26
27 echo "<pre>";
28 $Rows = $ToolkitServiceObj->CLInteractiveCommand("DSPLIBL");
29 if(!$Rows)
30     echo $ToolkitServiceObj->getLastError();
31 else
32     var_dump($Rows);
33 echo "</pre>";
34
35 $cmd_rtv = 'RTVJOBA JOB(?) USER(?) NBR(?) CURUSER(?) DFTCCSID(?N)';
36 //$cmd_rtv = 'RTVSYVAL SYSVAL(QCCSID) RTNVAR(?)';
37 $output = $ToolkitServiceObj->CLCommandWithOutput($cmd_rtv);
38 print_r($output);
39
40 ?>
```

Exemple (affichage)

```
5770SS1 V7R1M0 100423                               Liste des bibliothèques                               9/05/14 17:32:00                               Page 1
```

Biblio	Type	Unité	Texte descriptif
QSYS	SYS	ASP	Bibliothèque système
QSYS2	SYS		Bibliothèque système pour CPI
QHLPSYS	SYS		
QUSRSYS	SYS		System Library for Users
PHPSPool	CUR		Objets pour php.Spool
ZENDSVR	USR		
QGPL	USR		General Purpose Library
QTEMP	USR		

* * * * * F I N D E L I S T A G E * * * * *

Array ([JOB] => XTOOLKIT [USER] => GDUMAS [NBR] => 668083 [CURUSER] => GDUMAS [DFTCCSID] => 297)

iToolkitService

- Surcouche PHP pour l'accès à des objets spécifiques à l'IBM i :
 - SpooledFile : lecture de liste de spool par utilisateur, lecture d'un spool
 - DataQueue : création, suppression, écriture et lecture de dtaq
 - Userspace : création, suppression, écriture et lecture de user spaces
 - ObjectLists : lecture des objets d'une bibliothèque
 - JobLogs : lecture de liste de joblog par user, lecture d'une joblog
 - SystemValues : obtention de la liste des valeurs système ou d'une valeur système

Les spools

- Méthode: GetSPLList(\$usrprf)
 - Commande : WRKSPLF
 - En retour on obtient une liste descriptive des spools et des noms de jobs les ayant générés

- La lecture d'un spool se fait grâce aux informations lues par GetSPLList()
 - Méthode : GetSPLF()
 - Retour : contenu du spool sous forme d'une chaîne de caractères

Les DataQueues

- Création, Suppression, Ecriture des DTAQ
 - Création: `CreateDataQ()`
 - Qualification d'une DTAQ non créée: `SetDataQName()`
 - Suppression: `DeleteDQ()`
 - Lecture: `ReceieveDataQueue ()`: accès par comparaison de clés
 - Ecriture: `SendDataQueue()`
 - Vidage: `ClearDQ()`: par accès sur clé

Les Users Spaces

- Même fonctionnement que pour les DTAQ
 - Création: `CreateUserSpace()`

 - Qualification d'un UserSpace non créé: `setUSName()`

 - Suppression: `DeleteUserSpace()`

 - Obtention des attributs: `RetrieveUserSpaceAttr()`

 - Obtention de la taille: `RetrieveUserSpaceSize()`

 - Ecriture: `WriteUserSpace()`, paramètres: `start`, `len`, `value`

 - Lecture: `ReadUserSpace()`, paramètres: `from`, `len`

Les objets

- Commande WORKOBJ
 - Obtention de la liste des objets: getObjectList()

Les jobs

- L'accès à la liste des jobs et de leurs attributs s'effectue à travers l'objet JobLogs
 - Obtention de la liste des jobs d'un utilisateur: JobList()
 - Découpage de cette liste en tableau: createJobListArray()
 - Obtention des infos de log d'un job: JobLog()
 - Obtention des infos sur un job: GetJobInfo()

Les valeurs systèmes

- L'accès aux valeurs système peut se faire via l'objet SystemValues
 - Obtention de la liste complète, valeur comprise: SystemValuesList()
Commande WRKSYSVAL

 - Obtention d'une valeur: GetSystemValue(Nom)
Commande DSPSYSVAL SYSVAL(QCCSID)

S22 – PHP sous IBM i : accédez à vos données et applications avec le PHP Toolkit



Merci de votre attention



Gautier DUMAS - gdumas@notos.fr
04 30 96 97 31
www.notos.fr

