



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

**S27 – Accédez aux ressources IBM i depuis n'importe quelle
plateforme avec XML Service**

Mercredi 14 mai – 15h15-16h45

Jean-Marie SAAD – Gaia Mini Systèmes

Introduction

Problématique (1/2)

- Communication entre l'IBM i et les autres systèmes
 - Prévoir les conversions dans les deux sens
 - Spécificités des IBM i : data area, ...
 - Accéder à la partie métier
 - Ecrire du spécifique
 - Apprendre d'autres langages...

Problématique (2/2)

- Une solution : échanger en XML
 - Avoir un langage commun, simple
 - Des logiciels permettent la création (écriture) de documents XML et leur gestion
 - Des logiciels permettent la lecture de ces mêmes documents

- Pourquoi pas des services Web ?
 - Lecture/écriture en XML
 - Mais besoin d'utiliser des WSDL, des composants en C/C++, écriture de programmes de services...
 - Limitations des outils mis à disposition
 - Appels de programmes mais quid des autres objets ?

XML Service

- Echanger en XML
- Appel de tout objet sur l'IBM i
- Licence BSD
- Utilisable à partir de la V5R4 de l'OS
- Peut-être déjà présent sur votre IBM i (Zend ?)
- Bibliothèque RPG Open source
- Permet d'utiliser n'importe quel langage pour accéder aux ressources IBM i
 - Connexion en mode DB2
 - avec utilisation de programmes de service et de procédures stockées (*xmlstoredp.srvpgm*)
 - Connexion Apache REST
 - Méthodes GET/POST (*xmlcgi.pgm*)
- Attention !!! Encore en développement !!

XML Service

- Fruit de la collaboration d'IBM, de Zend et des membres du groupe Young i Professionals
 - Permettre de simplifier l'utilisation des éléments natifs de l'IBM i depuis l'extérieur
 - Nécessité pour Zend d'une nouveau toolkit, non propriétaire (Aura Eqpt)
 - Intégration dans les plateformes Zend i depuis 2012.
 - Fonctionne comme une procédure stockée générique
 - Lit et écrit du XML
 - Permet des structures de données complexes
 - Et c'est Open Source et gratuit (BSD) !!!

XML Service

- Tous les environnements pouvant envoyer et recevoir du XML peuvent utiliser XMLService
 - Essais concluants avec .NET
 - (Craig Pelkie – SystemiNetwork)

Installation

Avant tout gérer le CCSID

- Code Character Set Identifier
 - Permet d'indiquer la conversion entre EBCDIC et ASCII
 - La valeur 65535 (défaut) : ne jamais convertir
 - A bannir !!! Mais on a (presque) tous cette valeur par défaut.

 - CCSID :
 - 37 (USA)
 - 297 (France)
 - 500 (Canada/Belgique/Suisse)
 - 1208 (UTF-8)
 - 1147 (France Euro)

Prérequis

■ Dépendances

- PTF pour PASE, HTTP, DB2
 - http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&Start=1&Count=30&Collapse=1#1
- Si CCSID 65535 (!!!)
 - dans le fichier httpd.conf de votre serveur Apache indiquez
 - conf/httpd.conf
 - DefaultFsCCSID 1208
 - CGIJobCCSID 1208
- V5R4 appel en échec : OPM/CLP ou appel de RPG avec %parms
<pgm mode='opm' ...>

Utilisation

- Aller sur le site « Young i Professionals »
 - <http://youngiprofessionals.com/wiki/XMLSERVICE>
- Récupérez l'archive XMLSERVICE la plus récente et la dézipper (fichier .savf)

XMLSERVICE Archive (server)

Versions:

2013-11-21 - [xmlservice-rpg-1.8.3.zip](#)

2013-11-13 - [xmlservice-rpg-1.8.2.zip](#)

2013-09-06 - [xmlservice-rpg-1.8.1.zip](#)

2013-08-23 - [xmlservice-rpg-1.8.0.zip](#)

2013-03-25 - [xmlservice-rpg-1.7.8.zip](#)

- Suivez les instructions du site.

Travail

- A l'utilisation, un travail est créé
 - Sous système QSYSWRK
 - Travail XTOOLKIT
 - Initialisé avec le profil qui a ouvert la première connexion pour un IPC donné
 - L'IPC est un chemin IFS défini comme clé pour le travail.

| Opt | S-syst/trav | cours | Type | % UC | Fonction | Etat |
|-----|-------------|---------|------|------|----------------|------|
| — | QZSOSMAPD | QUSER | BCH | 0,0 | | SELW |
| — | XTOOLKIT | HSAAD | BCH | 0,0 | PGM-XMLSERVICE | SEMW |
| — | XTOOLKIT | HSAAD | BCH | 0,0 | PGM-XMLSERVICE | SEMW |
| — | XTOOLKIT | QSECOFR | BCH | 0,0 | PGM-XMLSERVICE | SEMW |
| — | QUSRWRK | QSYS | SBS | 0,0 | | DEQW |

Connexion publique

- XMLSERVICE Stateless
 - Chaque profil réinitialise le lancement
 - Plusieurs travaux peuvent être lancés
 - La durée de vie est limitée à une seule requête XmlIn/XmlOut
 - Pas de liste de bibliothèques
 - CTL : *here

Connexion privée

■ XMLSERVICE Statefull

- Lancement d'un seul travail
- Un profil possède sont propre travail XTOOLKIT
- Un profil donné est routé vers les travaux XTOOLKIT
 - CTL : *sbmjob
 - *Si pas lancé, création d'un nouveau travail XTOOLKIT*
 - IPC : chemin IFS servant de clé
 - *Toutes les requêtes sont routées vers le travail XTOOLKIT*
 - *Si le dossier n'existe pas dans l'IFS, il est créé.*
- La durée de vie n'est pas limitée dans le temps, tant que le travail XTOOLKIT tourne

Connexion Web/REST

■ API XMLCGI.pgm

<http://myibmi/cgi-bin/xmlcgi.pgm?db2=x@uid=x@pwd=x@ipc=x@ctl=x@xmlin=x@xmlout=x>

– Indiquer les paramètres

- DB2 : base de données
- UID/PWD : identifiant et mot de passe pour la connexion
- IPC : emplacement clé
- CTL : valeur de contrôle
- XMLIn : requête en entrée
- XMLOut : requête en sortie

Connexion DB2

- API : XMLSERVICE/XMLSTOREDP.SRVPGM
 - Utilisation de procédures cataloguées
 - Interface DB2 classique

```
iPLUG4K (IN IPC CHAR(1024),  
        IN CTL CHAR(1024),  
        IN XMLIN CHAR(4064),  
        OUT XMLOUT CHAR(4064))
```

```
iPLUG32K : idem sauf IN XMLIN CLOB(32000), OUT XMLOUT CLOB(32000)
```

```
iPLUG65K : idem sauf IN XMLIN CLOB(65K), OUT XMLOUT CLOB(65K)
```

```
iPLUG512K : idem sauf IN XMLIN CLOB(512K), OUT XMLOUT CLOB(512K)
```

```
iPLUG1M : idem sauf IN XMLIN CLOB(1M), OUT XMLOUT CLOB(1M)
```

```
iPLUG5M : idem sauf IN XMLIN CLOB(5M), OUT XMLOUT CLOB(5M)
```

```
iPLUG10M : idem sauf IN XMLIN CLOB(10M), OUT XMLOUT CLOB(10M)
```

```
iPLUG15M : idem sauf IN XMLIN CLOB(15M), OUT XMLOUT CLOB(15M)
```


Connexion DB2

- API : XMLSERVICE/XMLSTOREDP.SRVPGM

- Utilisation de procédures cataloguées

- gestion de curseurs requise

```
iPLUGR4K (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CHAR(4064))  
iPLUGR32K (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(32000))  
iPLUGR65K (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(65K))  
iPLUGR512K(IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(512K))  
iPLUGR1M (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(1M))  
iPLUGR5M (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(5M))  
iPLUGR10M (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(10M))  
iPLUGR15M (IN IPC CHAR(1024), IN CTL CHAR(1024), IN XMLIN CLOB(15M))
```

Utilisation Web

Serveur Apache

- Ajout dans le fichier **conf/httpd.conf** de votre instance Apache

```
ScriptAlias /cgi-bin/ /QSYS.LIB/XMLSERVICE.LIB/  
<Directory /QSYS.LIB/XMLSERVICE.LIB/>  
    AllowOverride None  
    order allow,deny  
    allow from all  
    SetHandler cgi-script  
    Options +ExecCGI  
</Directory>
```

Principe

- Appeler, via un formulaire, le programme xmlcgi.pgm
 - Définit l'entête

- Coder les scripts d'appel
 - Script de commandes en REXX ou autre

Entête

| Attribut | Description | Valeurs possibles |
|------------|--|---|
| db2 | Base de données | *LOCAL ou nom |
| uid et pwd | Profil et mot de passe | *NONE |
| ipc | Nom clé IPC pour router les travaux XMLSERVICE | |
| ctl | CTL admin control XMLSERVICE job | *immed *sbmjob *nostart *here... |
| xmlin | Requête XML en entrée | |
| xmlout | Requête XML en sortie | |

```
<form name="input" action="/cgi-bin/xmlcgi.pgm" method="post">
<input type="hidden" name="db2" value="*LOCAL">
<input type="hidden" name="uid" value="PROFIL">
<input type="hidden" name="pwd" value="PASSWORD">
<input type="hidden" name="ipc" value="/home/perso/xmlserv">
<input type="hidden" name="ctl" value="*sbmjob">
[...]
```

Entête : CTL

■ Valeurs pour CTL

| Valeur | Description |
|--|---|
| *immed | Arrêt du serveur immédiatement. Destruction de l'IPC |
| *sbmjob *sbmjob[(lib/jobd/ job/asp)] | Statefull |
| *here | Exécution en stateless dans un travail de procédure stockée |
| *nostart | Désactive la création |
| *session | Récupération de la clé de session IPC (chemin IFS...) |
| *license | Retourne la licence ce code |
| *ignore | Ignore les flags (appels en boucle, ...) |
| *clear | Mise à blanc du cache interne XMLSERVICE |

Script

- Attribut xmlin

- `<input type="hidden" name="xmlin" value="script" />`

```
<input type="hidden" name="xmlin"
value="<?xml version='1.0'?>
<script>
<cmd>CLROUTQ OUTQ (QGPL/TESTHP1022) </cmd>
</script>">
<input type="hidden" name="xmlout" value="32768">
[...]
```

- La commande devant être précédée du prologue XML :

- `<?xml version = '1.0' ?>`

Script : commande simple

- **Commande :**

```
<cmd [exec='cmd|system|rexx' error='on|off|fast' * hex='on']>IBM i  
command</cmd>
```

```
<script>
```

```
<cmd>CLROUTQ OUTQ(QUSRSYS/QEZJOBLOG</cmd>
```

```
<cmd>DSPLIB LIB(SQLFORM) OUTPUT(*PRINT) </cmd>
```

```
<cmd> ADDLIBLE LIB(QGPL) </cmd>
```

```
</script>
```


Script : retour de valeurs

■ Commandes RTVxxx

```
<script>
```

```
<cmd exec='rexx'>RTVJOB(USRLIBL(?) SYSLIBL(?) JOB(?) USER(?) NBR(?)  
CURUSER(?))</cmd>
```

```
</script>
```

```
script>  
-<cmd exec="rexx">  
-<success>  
+++ success RTVJOB(USRLIBL(?) SYSLIBL(?) JOB(?) USER(?) NBR(?) CURUSER(?))  
</success>  
-<row>  
-<data desc="USRLIBL">  
SQLFORM QGPL CPSPGM ARCAD_FRA ARCAD_PRD QTEMP  
</data>  
</row>  
-<row>  
<data desc="SYSLIBL">QSYS QSYS2 QHLPSYS QUSRSYS</data>  
</row>  
-<row>  
<data desc="JOB">XTOOLKIT</data>  
</row>  
-<row>  
<data desc="USER">HSAAD</data>  
</row>  
-<row>  
<data desc="NBR">402646</data>  
</row>  
-<row>  
<data desc="CURUSER">HSAAD</data>  
</row>  
</cmd>  
/script>
```

Script : retour de valeurs

■ Commandes RTVxxx

```
<script>
```

```
<cmd exec='rexx'>RTVJOBA JOB (?) NBR (?)</cmd>
```

```
</script>
```

– Ou pas de log

```
<script>
```

```
<cmd exec='system'>RTVJOBA JOB (?) NBR (?)</cmd>
```

```
</script>
```

Script : appel de programmes

■ Appel de programmes méthode 1

```
<script>  
  <cmd> CALL (SQLFORM/H001) PARM('QUSRSYS' 'PDF') </cmd>  
</script>
```

■ Appel de programmes méthode 2

```
<script>  
  <cmd> ADDLIB LIB (SQLFORM) </cmd>  
  <pgm name='H001'>  
    <parm><data type='10A' var='LIB'>QUSRSYS</data></parm>  
    <parm> <data type='10A' var='NOM'>PDF</data></parm>  
  </pgm>
```

Script : appel de programmes

■ Appel de programmes

```
<pgm name='MONPROG'>  
  <parm io='both'>  
    <data type='1A' var='NOM'>P1</data> </parm>  
    <parm io='both'>  
      <ds>  
<data type='2A' var='PRODUIT.PAYS'>33</data>  
<data type='4A' var='PRODUIT.CODE'>1977</data>  
<data type='7p4' var='PRODUIT.STOCK'>10.5000</data>  
</ds>  
      </parm>  
<return> <data type='100'>0</data></return>  
</pgm>
```

Script : PASE

■ Commandes PASE

```
<script><sh>pwd</sh></script>
```

```
<script rows='on'><sh>pwd</sh></script>
```

```
- <script>  
  <sh> /home/HSAAD </sh>  
</script>
```

```
- <script>  
  - <sh rows="on">  
    <row>/home/HSAAD</row>  
  </sh>  
</script>
```

Script : PASE

■ Commandes CLP via PASE

```
<script>
```

```
<sh rows='on'>/QOpenSys/usr/bin/system -i 'wrkactjob'</sh>
```

```
</script>
```

Si pas de rows='on'

```
<script>
<sh>
5761SS1 V6R1M0 080215 Gestion des travaux actifs 3/04/14 16:59:00 Page 1 Réinitialisation. ....: *NO Sous
.....: *NONE Séquence .....: *SBS Nom du travail .....: *ALL % UC .....: 0,0 Intervalle.
Numéro en cours Type Pool Pté UC Opér T-Rép E-S % UC Fonction Etat Unités d' QBATCH QSYS 397010 QSYS SBS
397022 QUSER PJ 2 20 0,0 0,0 PSRW 1 QLZPSERV QUSER 397026 QUSER PJ 2 20 0,0 0,0 PSRW 1 QNMAPINC
0,0 0,0 PSRW 1 QNPSERV QUSER 397025 QUSER PJ 2 20 0,0 0,0 PSRW 1 QZRCRVR QUSER 397023 QUSE
396936 QSYS SBS 2 0 0,2 0,0 DEQW 2 QSYSQCD QPGMR 397007 QPGMR BCH 2 10 0,0 0,0 PGM-QEZSCNEP
BCH 2 25 0,1 0 0,0 TIMW 1 QTFTP00101 QTCP 397079 QTCP BCH 2 25 0,0 0,0 DEQW 1 QTFTP00102 QTCP 397
ADMIN QLWISVR 397949 QLWISVR BCI 2 25 27,0 0 0,0 JVM-com.ibm.lw THDW 29 ADMIN QTMHHTP 397072
QTMHHTP BCI 2 25 12,0 0 0,0 PGM-QZSRLG SIGW 1 ADMIN QTMHHTP 397216 QTMHHTP BCI 2 25 8,4
JVM-com.ibm.lw THDW 37 ADMIN2 QLWISVR 397947 QLWISVR BCI 2 25 28,0 0 0,0 JVM-com.ibm.lw THDW 33
QTMHHTP 397239 QTMHHTP BCH 2 25 0,2 0 0,0 PGM-QZHBMAIN SIGW 1 APACHEDFT QTMHHTP 39725
QTMHHTP BCI 2 25 12,3 0 0,0 PGM-QZSRLG SIGW 1 APACHEDFT QTMHHTP 397276 QTMHHTP BCI 2 2
0 0,0 PGM-QZHBMAIN SIGW 1 CGIDEV2APA QTMHHTP 397256 QTMHHTP BCI 2 25 12,0 0 0,0 PGM-QZSRL
SIGW 60 CGIDEV2APA QTMHHTP 397287 QTMHHTP1 BCI 2 25 0,0 0 0,0 PGM-QZSRCGI TIMW 1 CGIDEV2A
QTMHHTP 397287 QTMHHTP1 BCI 2 25 0,0 0 0,0 PGM-QZSRCGI TIMW 1 CGIDEV2A
```

```
<script>
- <sh rows="on">
- <row>
5761SS1 V6R1M0 080215 Gestion des travaux actifs 3/04/14 16:56:52 Page 1
</row>
<row> Réinitialisation. ....: *NO</row>
<row> Sous-systèmes .....: *ALL</row>
<row> Limite pourcentage UC .....: *NONE</row>
<row> Limite temps de réponse .....: *NONE</row>
<row> Séquence .....: *SBS</row>
<row> Nom du travail .....: *ALL</row>
- <row>
% UC .....: 0,0 Intervalle .....: 00:00:00 Travaux actifs .....: 291
</row>
- <row>
Utilisateur -----Intervalle-----
</row>
- <row>
S-syst/travail Utilisateur Numéro en cours Type Pool Pté UC Opér T-Rép E-S % UC Fonction Etat Unités d'
</row>
- <row>
QBATCH QSYS 397010 QSYS SBS 2 0 0,0 0 0,0 DEQW 2
</row>
- <row>
QCMN QSYS 397011 QSYS SBS 2 0 0,0 0 0,0 DEQW 2
</row>
```

Script : SQL

■ Commande

- Propriétés de connexion : balise <options ... />
- Attribut options

```
<options options='noauto' autocommit='off' /></sql>
```

```
<sql><connect conn='myconn' options='noauto' /></sql>
```

```
<sql>
```

```
<query conn='myconn'>select * from lib/pays</query>
```

```
</sql>
```

```
<sql><fetch block='all' desc='off' /></sql>
```

```
<sql><free /></sql>
```

```
-<sql>
- <fetch block="all" desc="on" stmt="stmt1">
- <row>
  <data desc="IDENT">1</data>
  <data desc="PAYS">Albanie</data>
</row>
- <row>
  <data desc="IDENT">2</data>
  <data desc="PAYS">Algérie</data>
</row>
- <row>
  <data desc="IDENT">3</data>
  <data desc="PAYS">Allemagne</data>
</row>
```

desc = 'off' →

← desc = 'on'

```
<script>
- <sql>
- <options options="noauto" autocommit="off">
  <success>+++ success noauto</success>
</options>
</sql>
- <sql>
- <connect conn="myconn" options="noauto">
  <success>+++ success myconn</success>
</connect>
</sql>
- <sql>
- <query conn="myconn" stmt="stmt1">
  <success>+++ success select * from sqlform/pays</success>
</query>
</sql>
- <sql>
- <fetch block="all" desc="off" stmt="stmt1">
- <row>
  <data>1</data>
  <data>Albanie</data>
</row>
- <row>
  <data>2</data>
  <data>Algérie</data>
</row>
- <row>
  <data>3</data>
```

Utilisation en RPGLE/SQLRPGLE

Principaux éléments (1/2)

- Une bibliothèque à mettre en ligne
 - XMLSERVICE

- Deux programmes de services
 - XMLSERVICE/RPGKIT
 - XMLSERVICE/ZZMYAUTH

Principaux éléments (2/2)

- Trois procédures essentielles (communes à tous les types d'appels)
 - crtXML : création du flux
 - callXML : envoi du flux
 - dltXML : suppression du flux créé après appel

- Deux DS de paramétrage
 - xml_Auth_t : base de données et identification
 - xml_Rec_t : descriptif du flux

Principe

■ Schéma de base

- Création du modèle pour la session

```
crtXML (...) ;
```

- Appel du modèle

```
rc = callXML (...) ;
```

- Transmission de la demande (commande, ...)

```
rc = crtCmd (...) ;
```

```
rc = crtSH () ;
```

```
...
```

- Destruction du modèle

```
dltXML (...) ;
```

Data Structures

- xml_Auth_t

| Nom | Définition | commentaire |
|---------|------------|--|
| xml_db | 10A | Nom de la base de données *LOCAL 'nom' |
| xml_uid | 10A | Identifiant IBM i (profil) |
| xml_pwd | 10A | Mot de passe |

Data Structures

■ xml_Rec_t

| Nom | Définition | commentaire |
|--------------------------|---------------------------------|--|
| <code>xml_auth</code> | <code>likeds(xml_Auth_t)</code> | cf. DS <code>xml_Auth_t</code> |
| <code>xml_ipc</code> | 1024A | Dossier IFS pouvant contenir les clés de sécurité. |
| <code>xml_ctl</code> | 1024A | Contrôle (mots réservés) *immed , *sbmjob, *nostart, *here... |
| <code>xml_in_sz</code> | 10i0 | 4Ko → 15 Mo. Taille réservée en entrée |
| <code>xml_in_len</code> | 10i0 | Taille effective en entrée |
| <code>xml_out_sz</code> | 10i0 | 4Ko → 15 Mo. Taille réservée en sortie |
| <code>xml_out_len</code> | 10i0 | Taille effective en sortie |
| <code>xml_in_p</code> | pointeur | Pointeur d'adresse pour la donnée en entrée |
| <code>xml_out_p</code> | pointeur | Pointeur d'adresse pour la donnée en sortie |

xml_ctl

■ Valeurs pour xml_ctl

| Valeur | Description |
|--|---|
| *immed | Arrêt du serveur immédiatement. Destruction de l'IPC |
| *sbmjob *sbmjob[(lib/jobd/job/asp)] | Stateful |
| *here | Exécution en stateless dans un travail de procédure stockée |
| *nostart | Désactive la création |
| *session | Récupération de la clé de session IPC (chemin IFS...) |
| *license | Retourne la licence ce code |
| *ignore | Ignore les flags (appels en boucle, ...) |
| *clear | Mise à blanc du cache interne XMLSERVICE |

Procédure crtXML

■ Création du modèle d'appel

| | | | |
|----------|-------|--|--------------------|
| D crtXML | PR | | likeds(xml_Rec_t) |
| D auth | | | likeds(xml_Auth_t) |
| D ipc | 1024A | | value |
| D ctl | 1024A | | value |
| D sz_in | 10i 0 | | value |
| D sz_out | 10i 0 | | value |

- Création d'une DS de type **xml_rec_t** avec génération, entre autres, du prologue XML, et des sz_in/sz_out.

Procédure crtXML

■ Exemple

```
D auth          ds          likeds(xml_Auth_t)
D xml           ds          likeds(xml_Rec_t)

/free

  auth.xml_uid = 'XMLSERV' ;
  auth.xml_pwd = 'PWD2014' ;
  auth.xml_db  = '*LOCAL' ; // Nom ou *LOCAL
  control     = '*here';
  ipcRoute    = '/home/XMLSRV' ;

xml = crtXML(auth :ipcRoute :control :xmlIn :xmlOut);
```


Procédures callXML/dltXML

■ Appel du modèle

```
D callXML PR          10i 0
D xml                  likeds(xml_Rec_t)
```

– En retour (constantes dans RPGKIT_H.mbr)

```
XML_SUCCESS    0
XML_FAILURE   -1
```

Connexion à la base de données.

■ Suppression du modèle

```
D dltXML PR
D xml                  likeds(xml_Rec_t)
```

Procédure crtCmd

■ Exécution

```
D crtCmd      PR      10i 0
D xml         likeds(xml_Rec_t)
D uname      10a      value
D cmd        1024a    value
D err        10i 0    value options(*nopass)
D exc        10i 0    value options(*nopass)
```

– En retour (constantes dans RPGKIT_H.mbr)

- **XML_SUCCESS** 0
- **XML_FAILURE** -1
- **XML_UNAME_NOT_UNIQUE** -2 (choisir un nouveau nom)

Commandes CLP

■ Exécution

```
rc = crtCmd(xml:'chg1':'CHGLIBL LIBL(QTEMP XMLSERVICE)');  
rc = callXML(xml);
```

■ Exécution avec valeur de retour

//--- Création et appel de la commande

```
rc = crtCmd(xml:'rtv1':'RTVJOBA USRLIBL(?) SYSLIBL(?)');  
rc = callXML(xml);
```

//--- Valeurs de retour des mots-clés USRLIBL et SYSLIBL

```
rc = getCmdStr(xml:'USRLIBL':%addr(sULib):%size(sULib));  
rc = getCmdStr(xml:'SYSLIBL':%addr(sULib):%size(sULib));
```

Commandes PASE

- Exécution avec valeur de retour

```
rc=crtSH(xml : 'sh1' : 'system -i "WRKSYSVAL OUTPUT (*PRINT) "') ;  
rc = callXML(xml) ;  
sBUF = *BLANKS ;  
rc = getSHStr(xml : 'sh1' : %addr(sBUF) : %size(sBUF)) ;
```

Appels de programmes 1/3

■ Structure d'appel

```
rc = begPgm(xml:'PGM1':'PROG00':'XMLLIB');  
//--- Paramètres  
[...]  
rc = endPgm(xml);
```

■ Paramètres simples

```
rc = begParm(xml:'p1');  
    CHAR1a = 'a';  
rc = addVal(xml:'INCHARA':'a':1:0:%addr(CHAR1a));  
rc = endParm(xml);
```

Appels de programmes 2/3

■ Data structures

```
rc = begParm(xml:'p5');  
    rc = begDS(xml:'INDS1');  
        CHAR1a = 'g';  
        rc = addVal(xml:'DSCHARA':'a':1:0:%addr(CHAR1a));  
        CHAR1a = 'h';  
        rc = addVal(xml:'DSCHARB':'a':1:0:%addr(CHAR1a));  
        DEC7p4 = 1.1111;  
        rc = addVal(xml:'DSDEC1':'p':7:4:%addr(DEC7p4));  
        DEC12p2 = 22.22;  
        rc = addVal(xml:'DSDEC2':'p':12:2:%addr(DEC12p2));  
    rc = endDS(xml);  
rc = endParm(xml);
```

Appels de programmes 3/3

■ Retour des valeurs

```
/free
```

```
[...]
```

```
rc = getVal(xml:'DSCHARA':%addr(CHAR1a));
```

```
rc = getVal(xml:'INDEC1':%addr(DEC7p4));
```

```
rc = getVal(xml:'DSDEC2':%addr(DEC12p2));
```

```
rc = getValStr(xml:'INDEC1':%addr(sDEC7p4):%size(sDEC7p4));
```

```
rc = getValStr(xml:'DSDEC2':%addr(sDEC12p2):%size(sDEC12p2));
```

```
[...]
```

```
/end-free
```

Gestion des erreurs 1/2

■ Constantes

D XML_ERR_ON c const(1)

D XML_ERR_ON_C...

D c const('error="on"')

D XML_ERR_OFF c const(2)

D XML_ERR_OFF_C...

D c const('error="off"')

D XML_ERR_FAST...

D c const(3)

D XML_ERR_FAST_C...

D c const('error="fast"')

Gestion des erreurs 2/2

■ Monitoring

```
D sErrCPF S 7A
```

```
D sErrMsg S 65000A
```

```
    [...]
```

```
if isError(xml) = XML_SUCCESS;
```

```
    [...]
```

```
else ;
```

```
    getErrCPF(xml:%addr(sErrCPF):%size(sErrCPF));
```

```
//--- error = "ON"
```

```
    getErrMsg(xml:%addr(sErrMsg):%size(sErrMsg));
```

```
endif ;
```

SQL RPGLE

■ Création de la requête XML

```
myIPC = '/home/HSAADSQL';
myCtl = '*sbmjob';
myXmlIn =
'<?xml version="1.0" encoding="ISO-8859-1"?>'           + x'0D'
+ '<script>'                                             + x'0D'
+ '<pgm name="XS_01" lib="XMLSRC">'                     + x'0D'
+ '<parm><data type="1a" v="1">A</data></parm>'          + x'0D'
+ '<parm><data type="7p4" v="2">10.4590</data></parm>'    + x'0D'
+ '<parm><data type="12p2" v="3">3.14</data></parm>'      + x'0D'
+ ' <ds>'                                               + x'0D'
+ '  <data type="1a" v="4">A</data>'                     + x'0D'
+ '  <data type="1a" v="5">B</data>'                     + x'0D'
+ ' </ds>'                                              + x'0D'
+ '</parm>'                                             + x'0D'
+ '</pgm>'                                             + x'0D'
+ '</script>'                                           + x'00';
```

SQL RPGLE

■ Exécution

```
Exec Sql call XMLSERVICE/iPLUG4K(:myIPC, :myCtl, :myXmlIn, :myXmlOut);
```

■ Erreur

```
// error  
  
    posol = %addr(myXmlOut);  
    look1  = '<report>' + x'00';  
    look2  = '</report>' + x'00';  
    posol = strstr(posol:%addr(look1));  
    if posol <> *NULL;  
//--- Il y a une/des erreur(s)  
    endif;
```

Conclusion

- Interface unique permettant d'accéder aux ressources de l'IBM i
- Open Source et gratuit mais toujours en développement !!!
- Supporte des structures complexes : DS imbriquées, ...
- Utilisation de tous les objets courants IBM i
 - programmes
 - objets accessibles via les commandes CLP
 - ...
- Utilisable via des procédures stockées ou DB2
- Mais garde les limitations du XML sur IBM i (15 Mo pour le XML)

Références

- YIP

- <http://youngprofessionals.com/wiki/XMLSERVICE>

- Zend Toolkit Forum

- <http://forums.zend.com/viewforum.php?f=113>

- IBMSystems mag

- <http://www.ibmsystemsmag.com/ibmi/developer/rpg/EGL-and-XMLService--Hand,-Meet-Glove/>

- http://www.ibmsystemsmag.com/ibmi/developer/rpg/xmlservice_new_life/

Nous contacter

■ Par mail

- jmsaad@gaia.fr
- contact@gaia.fr

■ Nos sites

- www.gaia.fr
- www.know400.fr
- www.as400.fr