

GMO Monitoring JVM

Type diffusion
Révision 33

Auteur
Date de création

Roa
14/02/2011

Suite à une demande collégiale de la part des différents acteurs exploitant l'application GCE (clients, partenaires, AVV ...), une solution toute nouvelle fait son apparition afin de prendre en compte 2 points principaux d'administration :

- Possibilité de demander des rechargements de certaines tables de paramétrage (PPE ...)
- Monitoring périodique de la consommation mémoire (+ nombre de sessions métiers) et récupération des informations environnementales de l'AS

Important : il s'agit uniquement d'avoir une vision transversale et applicative de l'ensemble des instances de JVM GCE, en aucun cas cela ne doit venir en remplacement des outils fournis par les serveurs d'applications eux-mêmes, voire des outils tiers.

La version de ce document est basée sur un socle (TF) 8.0 qui sera distribué avec GCE161

L'information contenue dans ce document est fournie sans garantie d'aucune sorte, explicite ou implicite. Elle est fournie à titre éducatif et devra être relue et validée en environnement de test avant utilisation en production. L'utilisateur doit s'assurer du caractère approprié du contenu de ce document à son usage particulier, et assume le risque de son utilisation.

Société	: GENERIX Group	Titre	: <i>Projet : socle ET 8.0</i>
Dernière modification	: <i>Administrateur</i>	Sujet	: <i>GMO Monitoring JVM</i>
Temps de modification	: 2731	Résumé	:
Nom du fichier	GMO Monitoring GCE161.docx		
Liste de diffusion			

Siège Social et Agence Nord Immeuble le Verdi 6 rue du Moulin de Lezennes BP 10215 59654 VILLENEUVE D'ASCQ Cedex Tél. : +33 (0)3 20 41 48 00 Fax. : +33 (0)3 20 41 48 09 Service Technique : Tél. : +33 (0)3 20 41 48 07 Fax. : +33 (0)3 20 41 48 08	Agence Région Parisienne 69/71 rue Beaubourg 75003 PARIS Tél. : +33 (0)1 47 49 36 66 Fax. : +33 (0)1 47 51 22 17
---	--

Table des matières

1	Moyens mise en œuvre	3
1.1	Rechargement des tables de paramétrage	3
1.2	Monitoring	5
1.2.1	Exemple de requête multi-jvm	6
1.2.2	Calcul de AS_ID	6
1.3	APIs de gestion	7
2	Mise en place	8
2.1	Partie applicative	8
2.2	Utilisation via WS	9
3	Divers	10
3.1	Lien entre l'utilisateur et sa JVM	10
3.2	Servlet de service	10

1 Moyens mise en œuvre

La base de données constitue le seul moyen d'avoir une vision partagée d'un environnement de production (en tout cas d'un point de vue applicatif). De nouvelles tables ont donc été ajoutées afin de mettre à jour ces informations environnementales.

1.1 Rechargement des tables de paramétrage

Script DDL¹

```
create table ut_prg
/* Process Reloading Guarantee */
(codsoc integer,
 utimod varchar2(8),
 datmod varchar2(8),
 flag_tf integer,
 flag_bs integer,
 jvm_id varchar2(128),
 as_id varchar2(128),
 ip varchar2(50),
 server_name varchar2(256),
 reload_timer integer
) ;

alter table ut_prg add constraint ut_prg_idx1 primary key (jvm_id, as_id) ;
```

Description de la table

Tableau 1 : UT_PRG	
Champ	Description
codsoc	Non significatif : toujours à 0
datmod	La date de création de l'enregistrement
utimod	L'utilisateur ayant créé l'enregistrement (normalement GUEST)
flag_tf	Flag permettant de demander un rechargement lié au socle technique
flag_bs	Flag permettant de demander un rechargement lié à la session métier
jvm_id	UUID de la JVM : change a chaque démarrage de la JVM, cela permet de conserver une traçabilité
as_id	Identifiant de l'instance de l'AS
ip	L'adresse IP du serveur d'application (IPv4)
server_name	Le nom canonique du serveur d'application
reload_timer	NON UTILISE POUR LE MOMENT

Rechargement du socle technique : flag_tf

- Bit 0 : rechargement des fichiers de configuration (nop²)
- Bit 1 : rechargement de gce.properties
- Bit 2 : rechargement de LOG4J
- Bit 3 : vidage du cache de feuilles de style (nop)
- Bit 4 : forçage de la mise à jour de la table de monitoring

Rechargement de la session métier : flag_bs

- Bit 0 : paramétrage fonctionnel (PPE, PEV, UT_FCG)

¹ DDL : Data Definition Language (création d'objets dans la base, à ne pas confondre avec DML)

² Nop : non opérationnel dans l'immédiat

- Bit 1 : paramétrage des stocks (OSK, ASK, CSK)
- Bit 2 : nouvelles habilitations (UT_BV_HAB)
- Bit 3 : définition des BV (UT_BV_DEF)
- Bit 4 : définition des modèles de BV (UT_BV_MODELE)
- Bit 5 : définition des liens entre BV (UT_BV_LINK)

Exemple d'utilisation :

```
update ut_prg set flag_bs=3, flag_tf=16;  
commit;
```

On demande un rechargement du paramétrage fonctionnel (PPE ... + stocks) et de gce.properties sur l'ensemble des JVM pointant sur la base de données.

Important : afin de ne pas générer de contention inutile sur cette table, il convient de faire la mise à jour et de la valider (commit) directement dans la foulée.

Cette table est lue systématiquement lors de chaque URL, à l'instar de la table UT_UUID. Cela n'a aucun impact sur les performances, le temps mesuré sur un simple PC portable est inférieur à la ms.

1.2 Monitoring

Script DDL

```

create table ut_dma
/* Dynamic Monitoring Application */
(codsoc integer, utimod varchar2(8), datmod varchar2(8),
jvm_release varchar2(128),
jvm_id varchar2(128),
as_id varchar2(128),
jvm_local varchar2(16),
user_id varchar2(128),
os_name varchar2(256),
os_version varchar2(256),
ip varchar2(50),
server_name varchar2(256),
datheu_dem varchar2(19),
datheu_maj varchar2(19),
log_dir varchar2(256),
mem_xmx integer,
mem_mps integer,
mem_used integer,
gce_properties varchar2(256),
sm_count integer,
egx_release varchar2(32),
gce_release varchar2(32)
) ;

alter table ut_dma add constraint ut_dma_idx1 primary key (jvm_id, as_id);

```

Description de la table

Tableau 2	
Champ	Description
codsoc	Non significatif : toujours à 0
datmod	La date de création de l'enregistrement
utimod	L'utilisateur ayant créé l'enregistrement (forcément GUEST puisque le traitement est « socle »)
jvm_release	La version de la JVM utilisée au runtime (ex : 1.6.0_23-b05)
jvm_id	UUID de la JVM : change a chaque démarrage de la JVM, cela permet de conserver une traçabilité
as_id	Identifiant de l'instance de l'AS ³
jvm_local	La localisation de la JVM : doit être absolument à en_US pour GCE
user_id	L'utilisateur (au sens OS ayant démarré la JVM)
os_name	L'OS utilisé (ex : Windows 7)
os_version	La version de l'OS (ex : 6.1 correspond à la version DOS sous Windows)
ip	L'adresse IP du serveur d'application (IPv4)
server_name	Le nom canonique du serveur d'application
datheu_dem	Date de démarrage ⁴
datheu_maj	Date de dernière mise à jour de la table par la JVM
log_dir	Le répertoire de destination des logs GCE (et timers)
mem_xmx	La mémoire maximum utilisable par la JVM (correspond au paramètre -Xmx de la JVM)
mem_mps	La mémoire maximum définie pour le chargement des classes Java (XX:MaxPermSize)
mem_used	Mémoire utilisée au moment du monitoring (purement indicatif)
gce_properties	Le gce.properties utilisé ou null s'il s'agit de la valeur par défaut (gce.properties.xml)

³ L'identifiant de l'instance de la JVM : est dépendant du serveur d'application utilisé

⁴ Format yyyyMMdd-HHmms.SSS : descend donc à la milliseconde (ms)

sm_count	Le nombre de sessions métiers en instance dans la JVM (nop)
egx_release	La version eGX Technology (différente de <i>gce_release</i> à compter de la GCE160)
gce_release	La version GCE

La table de monitoring est mise à jour périodiquement par l'ensemble des JVM pointant sur la base de données si le paramétrage l'autorise (*gce.properties*). Ceci est réalisé via un Thread autonome qui maintient un *RootApplicationModule* (donc attention à la limite fixée dans le fichier de configuration de plus haut niveau car on en consomme donc un systématiquement). A noter que la connexion JDBC n'est pas maintenue sur ce thread afin de ne consommer la connexion « qu'au besoin ».

A ce sujet, il convient dorénavant de mettre en place des DataSources poolées, ceci quel que soit le serveur d'application utilisé. Dans ce cas, ne pas oublier de passer l'attribut disconnectFrequency de la configuration idoine à request et non plus session.

L'unité d'œuvre de la période de scrutation est la seconde, il ne semble pas très utile de descendre à une unité plus faible (genre ms ...). S'agissant d'un Thread de type monitoring, une période de 10s semble une bonne alternative.

1.2.1 Exemple de requête multi-jvm

```
select c.* from ut_dma c
where (c.as_id,c.datmaj) in (
  select as_id,datmaj
  from ut_dma b
  where b.datmaj=(select max(datmaj) datmaj
                  from ut_dma a
                  where a.as_id=b.as_id)
  group by as_id,datmaj)
order by c.datmaj desc;
```

1.2.2 Calcul de AS_ID

Ce champ permet de déterminer l'instance de JVM en fonction du serveur d'application. Quatre cas sont pris en compte :

- WebLogic 11g : paramètre *weblogic.Name*
- Oracle AS 10g : paramètre *oracle.ons.indexid*
- JBoss : nom du répertoire final définissant *user.dir*
- Indéterminé : dans ce cas, c'est le nom du serveur qui est utilisé

1.3 APIs de gestion

Un ApplicationModule a été créé au niveau du socle technique afin de proposer des services de gestions. L'ensemble des services a aussi été publié sous forme de WebServices afin de proposer un accès via des outils tiers.

Pour des raisons évidentes, cet ApplicationModule est « final », il ne peut donc pas être surchargé de manière spécifique.

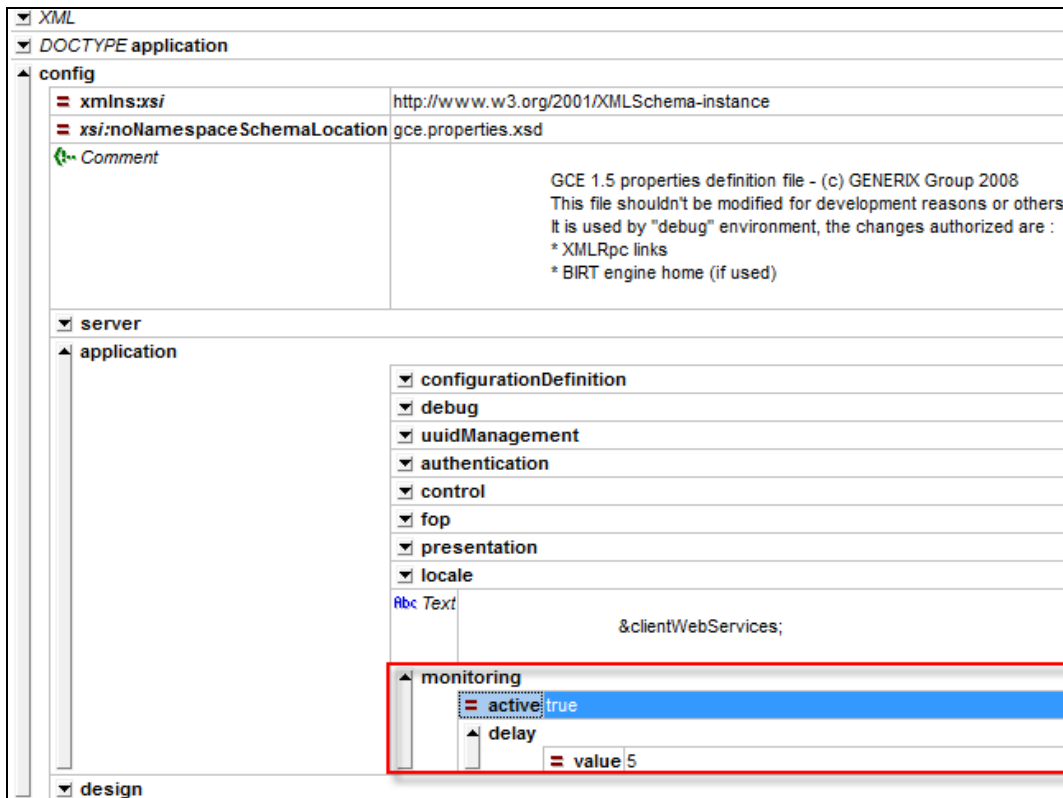
D'autre part, il n'est pas prévu de créer des DocumentRow afin de rendre ces services disponibles sur la couche de présentation.

2 Mise en place

Cette nouveauté est disponible depuis GCE160, elle a connu quelques évolutions.

2.1 Partie applicative

De manière exceptionnelle, le schéma de gce.properties a été modifié en GCE155. Il s'agit d'un élément optionnel donc sans impact sur un environnement déjà existant. Par contre il convient d'avoir le schéma à jour pour déclencher la fonctionnalité (et que le XML valide bien le schéma, sinon l'application ne démarrera pas).



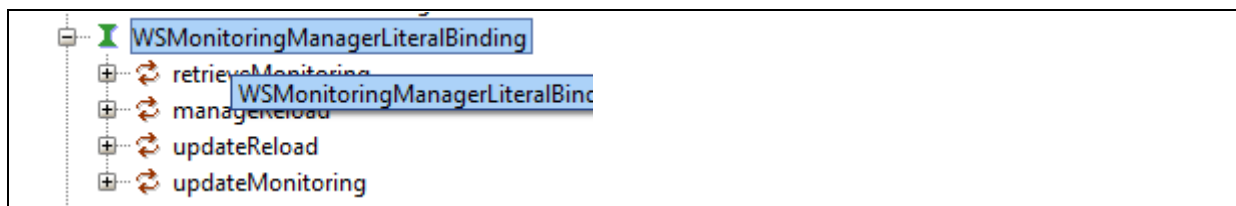
The screenshot shows an XML configuration editor with the following structure:

- XML
- DOCTYPE application
- config
 - xmlns:xsi: http://www.w3.org/2001/XMLSchema-instance
 - xsi:noNamespaceSchemaLocation: gce.properties.xsd
 - Comment: GCE 1.5 properties definition file - (c) GENERIX Group 2008. This file shouldn't be modified for development reasons or others. It is used by "debug" environment, the changes authorized are : * XMLRpc links * BIRT engine home (if used)
- server
 - application
 - configurationDefinition
 - debug
 - uuidManagement
 - authentication
 - control
 - fop
 - presentation
 - locale
 - Text: &clientWebServices;
 - monitoring
 - active: true
 - delay
 - value: 5
- design

Le délai spécifié est en secondes.

2.2 Utilisation via WS

Exemple depuis SOAPui :



Flux d'entrée pour le mode « retrieve » en littéral :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:GCE"
xmlns:com="http://www.generix.fr/technicalframework/businesscomponent/applicationmodule/common"
xmlns:ser="http://www.generix.fr/technicalframework/business/webservice/server">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:retrieveMonitoring>
      <!--Optional:-->
      <com:ctx entity="1" language="FRA" target="GART" user="DEMO"
password="DEMO"/>
      <com:MonitoringManagerretrieveMonitoringIn>
        <com:active>true</com:active>
      </com:MonitoringManagerretrieveMonitoringIn>
    </urn:retrieveMonitoring>
  </soapenv:Body>
</soapenv:Envelope>
```

Exemple d'URL d'accès au WSDL :

<http://localhost:8888/gce161/services/literal/monitoringManager?WSDL>

La valeur *active=true* est décorative mais elle est néanmoins indispensable au bon déclenchement des WebServices présentés ici.

Services disponibles :

- retrieveMonitoring : retourne un flux de description de la table UT_DMA
- manageReload : demande explicite de rechargement en se basant sur la table UT_PRG
- updateReload : mise à jour de la table UT_PRG
- updateMonitoring : mise à jour de la table de monitoring (UT_DMA) → champs DATMAJ & MEM_USED

A noter : le monitoring (suivant le paramétrage de gce.properties) est déclenché aussi bien sur l'IHM que sur les WS. Par contre les principes de rechargement du paramétrage (basé sur la table UT_PRG) n'est pas disponible à ce jour pour les WS.

3 Divers

3.1 Lien entre l'utilisateur et sa JVM

La table UT_UUID contient désormais le champ JVM_ID qui contient l'identifiant unique de la JVM. De ce fait, on peut savoir précisément sur quel environnement travaille un utilisateur en requêtant les tables UT_DMA et UT_UUID.

3.2 Servlet de service

Elle autorise une interrogation de certaines activités via une URL. Elle renvoie un flux XML qui peut être analysé et traité via des outils de monitoring tiers (nagios ...).

Plusieurs modes d'interrogation sont disponibles :

- ping : la JVM est-elle démarrée, l'application est-elle démarrée, répond-elle ?
- status : vision de la JVM courante en terme de sessions utilisateurs + vision inter-JVM qui pointent sur la même base de données.
- activity

Important : le monitoring doit être actif afin de bénéficier de l'ensemble des possibilités, il se pilote depuis gce.properties :

```
<monitoring active="true">  
  <delay value="10"/>  
</monitoring>
```

Par défaut, le monitoring est actif et il est désormais impossible à désactiver depuis la version 8.0 du socle technique.
