



LIVRE BLANC

MIGRATION WEBSHERE 6.1

<http://blog.xebia.fr>

<http://www.xebia.fr>

Copyright © Xebia 2007

Xebia IT Architects SAS
10/12 Avenue de l'Arche
92419 Courbevoie Cedex
Tél : 01 46 91 76 16
mail : info@xebia.fr



○ Migration WebSphere 6.1

Migrer sa plate-forme applicative d'une version d'un serveur d'applications à une autre est souvent vécu comme une démarche coûteuse, risquée, décorellée des enjeux opérationnels du métier, et laissant entrevoir un retour sur investissement pour le moins douteux ; ceci est d'autant plus vrai que cette migration est imposée de l'extérieur par le cycle de vie des produits de l'éditeur, et contrainte par une date butoir - celle de la fin du support de la version canonique - qui peut entrer en conflit avec d'autres échéances.

La migration vers Websphere 6.1 n'échappe bien sûr pas à cette règle – et ce d'autant plus que la version 7 de Websphere est espérée pour le premier trimestre 2008 ; pour autant, la richesse de la dernière mouture du serveur d'applications d'IBM permet d'envisager également la migration comme une opportunité de profiter de nouvelles fonctionnalités, permettant d'améliorer la disponibilité des plates-formes, de simplifier les infrastructures ou encore de déployer des applications plus performantes et plus riches. Cette migration peut aussi être l'occasion de réétudier le choix de la version de Websphere la plus adaptée aux besoins d'infrastructure : Websphere base, Network Deployment, eXtended Deployment, chaque version apporte ses fonctionnalités propres - et ses coûts.

La migration vers Websphere 6.1 doit donc être abordée selon deux axes : celui du risque, d'une part, dont la maîtrise passe par une analyse systématique d'impact portant sur l'ensemble de la chaîne de production logicielle – applications, environnements de développement, infrastructure, architecture, administration, supervision, déploiement, performances, etc. ; celui de l'opportunité, d'autre part, qui permettra de définir une cible conforme aux enjeux du Système d'Information des prochaines années et d'en réajuster le coût total de possession.

Cette migration doit également s'adosser à une méthodologie claire, autorisant une transition éclairée et maîtrisée, et dont les jalons macroscopiques sont les suivants : analyse d'impact, définition de la cible et du mode de transition, expérimentation puis industrialisation.

Table des matières

Introduction : Migrer ou Sisyphe à l'ouvrage.....	3
Nouveautés de Websphere 6.1.....	5
Langage et APIs.....	5
Web services.....	5
Sécurité.....	6
Architecture.....	7
Administration et maintenance.....	7
Analyse d'impact.....	8
Impacts sur l'architecture technique.....	8
Impacts sur l'exploitation et l'administration.....	18
Impacts sur les développements.....	22
Stratégie de migration.....	27
Stratégie de migration pour l'exploitation.....	27
Stratégie de migration de l'infrastructure.....	27
Stratégie de migration des applications	29
Planning de migration / séquençement	31
Phase de Prototypage.....	31
Phase de déploiement.....	32
Phase de rationalisation.....	33
Bibliographie.....	34
Conclusion.....	35
Annexe 1 - Matrices des versions d'APIs et bibliothèques Websphere.....	36
Annexe 2 - Fichiers et répertoires à prendre en compte lors des sauvegardes.....	38

Introduction : Migrer ou Sisyphe à l'ouvrage

Peut-être vous souvenez-vous du fameux mythe de Sisyphe, dans lequel l'infortuné est condamné ad vitam à pousser un lourd rocher jusqu'au sommet d'une montagne, avant de le laisser rouler jusqu'en bas pour mieux pouvoir recommencer.

La perspective de migrer sa plateforme JEE vers une nouvelle version d'un serveur d'applications plonge souvent ceux qui y sont contraints dans une humeur proche de celle de Sisyphe au pied de sa montagne : pour souvent avoir déjà vécu une migration du même ordre, ils savent que la tâche sera pénible ; et s'ils suivent un tant soit peu l'actualité JEE, ils se doutent qu'une fois le sommet atteint, l'horizon d'un nouveau cycle se sera radicalement rapproché ...

Une migration verticale – à opposer à une migration horizontale, vers un autre serveurs d'application – est en effet rarement motivée par des objectifs opérationnels positifs. Si l'on change d'éditeur pour gagner quelque chose – de la souplesse, des performances, de l'argent -, on « upgrade » en général plus trivialement pour ne pas perdre le support de sa version canonique, et on le fait sous la pression d'une date butoir imposée par l'éditeur.

Le tableau suivant fournit pour mémoire la date de fin de support des différentes versions de Websphere :

Version de Websphere	Fin du support
3.5	30 Novembre 2003
4.0	30 Avril 2005
5.0	30 Septembre 2006
5.1	26 Septembre 2008
6.1	???????

Sisyphe, cependant, poussait perpétuellement le même rocher vers le même sommet ; la migration vers Websphere 6.1 offre fort heureusement des perspectives nettement plus alléchantes ; la montée de version globale du système JEE - machine virtuelle, spécifications, API – reste naturellement, et malgré la compatibilité annoncée, une source de risque évidente pour la stabilité ou même le fonctionnement des applications. Mais elle fournit également au développement la possibilité d'adopter les bibliothèques et standards les plus récents, et de bénéficier des nombreuses innovations de la javasphère. Websphere 6.1 vient également avec un ensemble d'améliorations portant sur tous les aspects du déploiement d'une infrastructure JEE : architecture haute disponibilité, administration, supervision, déploiement, performances, etc.

C'est la raison pour laquelle notre premier chapitre s'attachera à faire un tour d'horizon aussi exhaustif que possible des nouveautés de la plate-forme, et en particulier de celles qui semblent les mieux à même de fournir des leviers de retour sur investissement à la migration.

Par ailleurs, les applications J2EE ont souvent par nature des caractéristiques très complexes :

- Ce sont des applications typiquement distribuées sur différentes couches, au moyen de différents protocoles : Client, Présentation, Fonctionnel (Services métier), Intégration et Ressources
- Elles utilisent une vaste palette technologique : JSPs, Servlets, EJBs, JDBC, JMS, JTS/XA, JCA, SOAP, CORBA, etc.
- Elles intègrent un grand nombre de bibliothèques et logiciels tiers : Bases de données, Annuaire, Logging, Persistance, Frameworks MVC, IoC et autres, MOMs, ORBs, Moniteurs transactionnels, etc.
- Elles couvrent des domaines fonctionnels très divers
- Elles nécessitent un niveau de qualité de service très élevé : Temps de réponse et Débit contractualisés, Haute Disponibilité, Augmentation de capacité

La maîtrise de cette complexité est souvent le fruit d'une maturation longue et minutieuse, et d'une mise au point progressive qu'une migration trop brutale, ou mal préparée, risque de mettre en péril.

Notre second chapitre s'attachera donc à décrire l'ensemble des impacts de la migration, en particulier sur tous les aspects touchant à l'exploitation de la plate-forme, et à fournir les clés permettant de minimiser dès les phases de préparation les risques inhérents à la migration.

Le dernier chapitre, enfin, abordera le thème de la méthode : quelle stratégie adopter pour la migration – Big Bang ou cohabitation - ?, comment planifier les différentes étapes et comment les articuler entre elles ?

Nous espérons qu'au terme de cette lecture, vous aborderez la côte sinon avec enthousiasme, du moins avec un sourire confiant, et le sentiment que là-haut, l'herbe est plus verte.

Nouveautés de Websphere 6.1

Langage et APIs

Le support de Java 5 est indéniablement l'une de nouveautés les plus marquantes de Websphere 6.1, celle qui souvent justifie pleinement la migration aux yeux des développeurs. La version 5 de Java apporte en effet la plus grande évolution du langage Java depuis sa création ; sans exhaustive :

- L'introduction dans la syntaxe Java des génériques et des types énumérés apportent une clarification et une meilleure auto-documentation du code.
- Les annotations permettent d'alléger les fichiers de configuration en décrivant dans le code les informations de configuration 'statiques' (mapping objet-relationnel, démarcation des transactions, association commande-page pour la couche MVC ...). Les frameworks de dernière génération, Struts2, Spring, Hibernate et JAXB/JAX-WS en premier lieu, utilisent largement les annotations.
- Côté nouvelles API, citons la librairie util.concurrent, qui adresse le besoin grandissant de développement multi-threadé, et l'instrumentation du Classloader qui permet la généralisation de la Programmation Orientée Aspect.

En complément de ces évolutions du langage, la JVM 5 d'IBM a également été l'objet d'une modernisation substantielle. Ses performances, en particulier, ont été significativement améliorées par l'introduction d'une gestion « générationnelle » de la mémoire (Generational Concurrent Garbage Collector) – une approche introduite par Sun dans sa propre JVM, HotSpot, dès la version 1.3. Le GC générationnel a un impact spécialement sensible sur les performances des applications fortement consommatrices d'objets dotés d'une très courte durée de vie – ce qui est typiquement le cas des applications Web. La gestion des threads ainsi que la compilation « Just-in-time » ont elles aussi été optimisées.

Websphere 6.1 est certifié J2EE 1.4 (on se reportera à l'annexe 1 pour une matrice complète des niveaux de spécification correspondant). Certaines innovations méritent une attention particulière, en raison des axes de migration évolutive qu'elles offrent :

- Invocations asynchrones et callbacks avec JCA 1.5
- Unification des APIs point-to-point et publish-subscribe avec JMS 1.1
- WorkManager (JSR 237) and Timers (JSR 236) pour les applications asynchrones et planifiées (scheduled)

Web services

Pour les applications qui ont choisi l'implémentation SOAP de Websphere plutôt qu'une librairie tierce (telle que Axis), Websphere 6.1 apportera des améliorations longtemps attendues.

La nouveauté majeure est l'introduction du runtime client qui permet de déployer des stubs SOAP Websphere hors du runtime Websphere, même sur des JVM non IBM¹. Cette nouveauté simplifie le processus de build en permettant de générer simultanément les implémentations clientes et serveurs avec la même librairie SOAP alors qu'il était au préalable nécessaire d'utiliser une librairie tierce (le plus souvent Axis) pour la couche cliente.

¹Voir Infocenter : [Running an unmanaged Web services JAX-RPC client](#)

Par ailleurs, Websphere 6.1 introduit le support de nouvelles fonctionnalités WS-* dont le succès auprès des autres acteurs des Web Services n'est pas encore avéré :

- Web Services Notification (WS-N) standardise les interactions entre les web services grâce à des mécanismes de notification et d'événements.
- Web Services Interoperability Basic Security Profile (WS-I BSP) standardise les mécanismes de sécurisation des web services pour faciliter l'interopérabilité
- Web Services Business Activity (WS-BA) standardise le 'rollback' des transactions dans le cadre des commits multi-phase

Il faut toutefois rappeler deux bémols à l'implémentation Websphere de SOAP qui n'ont pas été corrigés avec la V6.1:

- La documentation de la librairie est toujours succincte. L'infocenter est peu détaillé et les forums Websphere abordent rarement la couche SOAP. Il faut donc s'organiser pour consulter fréquemment le support Websphere
- La version de la librairie SOAP de Websphere est toujours liée à la version de Websphere. Il faudra donc migrer vers Websphere 7 (!) pour profiter des nouvelles fonctionnalités de cette librairie. Pour mémoire, les nouveautés introduites dans la v 6.1 n'ont pas été backportées sur Websphere 5.1 pour les clients qui n'ont pas encore migré.

⚠ Ce problème d'adhérence de la version d'une API à la version de Websphere est particulièrement délicat pour les technologies Web Services qui évoluent très vite. La capacité à monter de version de Websphere doit donc être un paramètre important lors du choix d'une implémentation SOAP (websphere-soap vs. Axis vs Xfire)

Sécurité

Pour les applications qui utilisent authentifications et autorisations J2EE, Websphere gérait jusqu'ici les utilisateurs dans une seule source de données (système d'exploitation, serveur LDAP ou « Custom User Registry »). La version 6.1 offre maintenant la possibilité de gérer en lecture et en écriture les identités stockées dans plusieurs sources de données (LDAP, JDBC ou fichier) grâce au Federated User Repository.

Cette amélioration permettra d'éviter le développement de complexes Custom User Registry réputées coûteuses en énergie aussi bien pour les équipes de développement que celles de déploiement.

Pour aller plus loin sur cet aspect :

- [Expand your user registry options with a federated repository in WebSphere Application Server V6.1](#) by IBM Software Services for Websphere in IBM DeveloperWorks - WebSphere Developer Technical Journal
- [WebSphere Application Server V6.1: What's new in security?](#) by IBM Software Services for Websphere in IBM DeveloperWorks

Architecture

Websphere embarque une nouvelle version de son Embedded Messaging Engine, un bus JMS qui suffit pour les communications entre serveurs d'une cellule Websphere. La substitution de Websphere MQ par ce moteur JMS intégré peut s'avérer une source d'économies substantielles – tant en coûts de licence qu'en charge d'exploitation.

Websphere 6 offre en standard la haute disponibilité pour la plupart des services critiques (moteur de messages JMS, gestionnaire des transactions, répartisseur de charge ...) avec un simple système de fichiers distribués comme pré-requis d'infrastructure.

Le dernier « single point of failure » est le Deployment Manager mais son rôle est désormais limité à la configuration des serveurs et n'est donc plus critiques pour le fonctionnement des applications déployées. Si la haute disponibilité de ce composant est requise, une architecture de cluster actif-passif avec un répertoire partagé pour la configuration est recommandée.

Par ailleurs, la fragilité connue de « single point of failure » du serveur LDAP a été résolue avec les nouveaux mécanismes de reprise sur panne LDAP.

Administration et maintenance

La grande nouveauté de Websphere 6.1 en matière d'administration est l'introduction d'un véritable environnement d'administration intégré : Websphere Application Server Toolkit (AST). AST est à l'administration Websphere ce que les IDEs sont au développement Java. On y retrouve la possibilité d'administrer à distance les serveurs Websphere, un éditeur et un débogueur de scripts jython-wsadmin, des éditeurs graphiques pour les fichiers de configuration spécifiques Websphere (ibm-web-bnd.xml, ibm-web-ext.xml ...) et les fonctionnalités standards Eclipse (versionnage des fichiers ...).

L'administration Websphere a aussi été simplifiée grâce à l'introduction d'API de scripting de haut niveau orientées tâches (commandes AdminTask de l'API wsadmin).

L'administration des serveurs web associés à des serveurs Websphere a été améliorée par l'intégration des commandes d'administration (stop, start, reconfigure et statut) dans la console d'administration Websphere.

Enfin, l'installation des serveurs Websphere a été simplifiée avec l'introduction de l'Installation Factory pour créer des packages d'installation sur mesure (composées des binaires WAS et potentiellement de fix packs, de paramètres de configuration et d'EARS). Cet outil bénéficiera principalement aux grandes topologies Websphere ND qui nécessitent de fréquentes installations de serveurs et aux éditeurs de logiciels qui embarquent un serveur d'application Websphere dans leur produit.

Websphere 6.1 permet une montée de version globale de la plateforme JEE. Java 5, J2EE 1.4, JVM optimisée, meilleur support des Web Services, autant d'innovations qui ouvrent la voie à des améliorations sensibles de la productivité des équipes de développement, à une simplification de certaines architectures logicielles et à des gains notables de performances.

Analyse d'impact

Impacts sur l'architecture technique

Serveurs et systèmes

Websphere propose de gérer la tenue en charge par une architecture de type « scalabilité horizontale » aussi appelée scale-out. Le principe de ces architectures est de mettre en cluster des serveurs de puissance limitée à faible coût plutôt que d'ajouter des ressources (CPU et RAM) à des serveurs complexes. Pour augmenter la puissance de traitement, il suffit d'ajouter un nouveau noeud au cluster. Les serveurs lames et les serveurs 1U sont particulièrement adaptés à ce type d'architecture.

Le mode de facturation de Websphere est particulièrement adapté aux serveurs lames (x86 et Power) pour lesquels IBM divise par deux le prix des licences¹. Par ailleurs, si les serveurs lames sont fréquemment utilisés avec Linux ou Windows², les Unix « historiques » sont aussi disponibles sous forme de lames³. Il est aussi intéressant de noter que les architectures serveurs lames sont compatibles avec la virtualisation des ressources.

L'architecture On Demand de Websphere eXtended Deployment est l'approche scale-out la plus poussée de la gamme Websphere. Il faut toutefois garder en mémoire que Websphere XD est adapté à des besoins tant en terme de tenue à la charge qu'en disponibilité. La plupart des architectures Websphere se contenteront des versions Base et ND.

Pour aller plus loin :

- IBM Education : [Websphere Application Server V6 - high Availability Overview - v6 : HA configuration \(pdf\)](#)
- Redbook : [Using WebSphere Extended Deployment V6.0 To Build an On Demand Production Environment](#)

¹50 Unités de Valeur en serveur lame contre 100 Unités de Valeur pour un serveur unix (AIX P-Series, HP 9000 ...), voir [IBM Value Unit Calculator](#)

²[IBM BladeCenter HS21](#), [HP Proliant BL20p Blade Server](#) ...

³[IBM BladeCenter JS21](#) pour AIX, [HP Integrity BL60p](#) pour HP-UX ...

Le tableau suivant indique les versions logicielles minimum pour installer Websphere Application Server V6.1 :

Système d'exploitation	Pré-requis	Commentaire
AIX	5L 5.2 avec Maintenance package 5200-07 5L 5.3 avec Service Pack 5300-04-01	
Linux x86	Red Hat Enterprise Linux AS/ES/WS, V3.5+ et V4.2+ SUSE Linux Enterprise Server, V9 SP2 + et V10+ Red Flag DC, Version 5.0 SP1+	L'installation sur d'autres distributions (Fedora, Ubuntu ...) est techniquement possible mais non supportée.
Windows	2000 SP4+ 2003 SP1+ XP SP3+	

Sauvegarde et restauration

Le stockage sous forme de fichiers de la configuration Websphere introduit avec la V4 permet l'utilisation de systèmes de sauvegarde incrémentale des fichiers. La fréquence de sauvegarde est adaptable suivant le type de fichier pour limiter l'impact des procédures de sauvegarde sur la performance des environnements :

Composant	Fréquence de sauvegarde
Configuration du Deployment Manager	Tous les jours
Configuration des Application Servers	Après chaque installation d'application
Binaires et configuration	Avant et après chaque installation de fixpack

Les systèmes génériques de sauvegarde et restauration tels que [BrightStor ARCserve Backup](#), [HP OpenView Storage Data Protector](#) ou [EMC Networker](#) ne nécessitent pas de mise à jour. Les équipes d'exploitation doivent seulement configurer le système pour qu'il sauvegarde les nouvelles arborescences de répertoires (voir « Annexe 2 : Fichiers et répertoires à prendre en compte lors des sauvegardes »).

Les systèmes de sauvegarde et restauration spécialisés avec un plugin Websphere tels que [IBM Tivoli Storage Manager for Application Servers](#) doivent d'abord être mis à jour pour prendre en compte Websphere 6.1 (l'arborescence des répertoires a changé). Ensuite, les équipes d'exploitation doivent configurer le système pour qu'il sauvegarde l'application.

Supervision

Les besoins d'exploitation et de supervision sont très différents selon les environnements. Les environnements de production et de pré-production nécessitent fiabilité et rigueur alors que les environnements de développement, d'intégration et de recette doivent faciliter les déploiements fréquents et le travail des équipes de développement.

Les procédures utilisées sur les environnements de production doivent être guidés par la rigueur et la fiabilité. Ce sont des environnements qui sont beaucoup plus à l'écoute des besoins des équipes de production que des besoins des équipes de développement.

Bien que celui puisse être contraignant, la même rigueur doit s'appliquer sur l'environnement de pré-production qui permet ainsi de valider les procédures destinées aux environnements de production.

A l'inverse, les environnements de développement (intégration, recette, tests ...) doivent eux être au service des équipes projet. La productivité doit y être privilégiée pour apporter de l'agilité aux équipes projets. Les aspects d'auditabilité, de traçabilité et de fiabilité sont moins importants hors de la production.

Les environnements de production et pré-production devraient donc être :

- Exploités par des scripts d'administration et des systèmes d'administration centralisés
- Supervisés par des systèmes de supervision intégrés
- Sécurisés avec des firewalls stricts et des mots de passe seulement connus des équipes d'exploitation

Les autres environnements (développement, intégration, tests, etc.) devraient pour leur part être :

- Exploités par la console d'administration de Websphere et des scripts. Les scripts sont préférés pour leur
- Supervisé avec parcimonie par un système centralisé de supervision (systèmes de fichier, CPU, mémoire)
- Peu sécurisés avec des mots de passe connus des équipes de développement et des firewalls lâches (cloisonnant les environnements les uns des autres mais pas entre les développeurs et les serveurs)

Websphere expose ses données de performance (Performance Monitoring Infrastructure - PMI) au travers de la [PerfServlet](#) et du [Connecteur JMX](#). La version 6.0 a introduit un nouveau format de données de performance qui repose sur [J2EE 1.4 Performance Data Framework](#) (un sous ensemble de [JSR-77 : J2EE Management](#)) ; ce nouveau format est incompatible avec le format des versions 5.x. Cependant, la PerfServlet supporte un mode compatible ascendant grâce à l'ajout du paramètre "version=5" dans l'URL d'accès (voir Infocenter : [PerfServlet output](#)).

Les systèmes d'administration et de supervision qui reposaient sur la PerfServlet et le connecteur JMX supporteront sans difficulté Websphere 6.1. La plupart des outils de supervisions pour Websphere supportent déjà la version 6.1. On retrouve notamment :

- [BMC Performance Manager for Web Application Servers](#) (anciennement BMC Patrol) (cf [BMC Product Availability/Compatibility](#))
- [HP OpenView Smart Plug-in for IBM WebSphere Application Server](#)
- [IBM Tivoli Composite Application Manager for WebSphere](#)
- [Mercury SiteScope Websphere Solution Template](#)
- [Hyperic HQ Agent plugin for Websphere](#)

Serveurs Web

IBM Http Server (IHS) est le serveur web le mieux intégré aux architectures Websphere. La version 6.1 a introduit des fonctionnalités qui rendent IHS désormais beaucoup plus intéressant que Apache.

IHS peut désormais être intégralement administré (stop, start et configuration) au travers de la console web et du scripting Websphere. Cette nouvelle fonctionnalité améliore la fiabilité des déploiements et diminue les coûts d'exploitation¹.

L'architecture Websphere de référence est d'utiliser un répartiteur IP qui va indifféremment répartir les requêtes sur tous les serveurs Web et déléguer à ces derniers le routage vers les serveurs d'applications. Le répartiteur fournit la haute disponibilité et les serveurs Web fournissent la tenue à la charge et le routage.

Déléguer à un répartiteur IP le routage est sensiblement plus complexe que de reposer sur les serveurs web car il faut mettre à jour la configuration du répartiteur lors des installations/désinstallations d'applications (les équipes sont le plus souvent différentes et les technologies ne sont pas intégrées) alors que la propagation des configurations aux plugins des serveurs web est elle transparente.

Voici les pré-requis logiciels pour les différents serveurs Web supportés :

Serveur Web	Pré-requis	Commentaire
IBM HTTP Server (IHS)	6.0.2 6.1	IHS est une version d'Apache HTTPD enrichie de fonctions spécifiques IBM (intégration à Websphere, SSL ...)
Apache HTTP Server	2.0.54	
Microsoft IIS	5.0 6.0	
Lotus Domino Enterprise Server	6.5.4 7.0	
Sun Java System Web Server	6.0 SP9 6.1 SP3	

Source : [List of supported software for WebSphere Application Server V6.1](#)

¹L'administration d'IHS depuis le Deployment Manager nécessite l'ouverture d'une communication HTTP depuis le Deployment vers le port d'administration du serveur IHS

Fonctionnalités des serveurs Web :

Fonctionnalité	IBM HTTP Server	Autre serveurs web (Apache , IIS ...)	Commentaires
Supervision du statut du serveur	✓	✓	
Arrêter / Démarrer le serveur	✓	✗	Le plugin Websphere peut automatiquement recharger sa configuration si elle a changée sans nécessiter de redémarrage du serveur web (activé par défaut toutes les 60 s)
Génération de la configuration du plugin	✓	✓	Génère le fichier plugin-cfg.xml sur le Deployment Manager mais ne le déploie pas sur les serveurs web cibles
Propagation du fichier de configuration du plugin	✓	✗	Déploie le fichier plugin-cfg.xml sur les serveurs web cibles
Visualisation des fichiers de log du plugin dans la console d'administration	✗	✗	
Visualisation des fichiers de log du serveur web dans la console d'administration	✓	✗	
Visualisation des fichiers de configuration déployés sur les plugins depuis la console d'administration Websphere	✗	✗	
Edition et visualisation du fichier de configuration du serveur web depuis la console d'administration Websphere	✓	✗	

Bases de données

L'accès aux bases de données depuis un serveur J2EE est maintenant complètement banalisé. Il est souhaitable d'utiliser des drivers JDBC de type 4 (100% pure Java) pour accéder aux SGBD.

Concernant l'utilisation des DataSources, Websphere 6.1 incite plus fortement au respect de la norme J2EE en rendant deprecated les lookups JNDI directs¹. Il faut dorénavant déclarer une `<resourceref>` dans le fichier web.xml (ou ejb-jar.xml) et utiliser la syntaxe `ctx.lookup("java:comp/env/jdbc/my-data-source")` au lieu de l'ancienne syntaxe `ctx.lookup("jdbc/my-data-source")`.

Voici les pré-requis logiciels pour les principales bases de données :

Base de données	Pré-requis
IBM DB2	DB2® pour iSeries™ 5.2, 5.3 ou 5.4 DB2® pour z/OS™ V7 ou V8 DB2® pour Linux, UNIX, and Windows V8.2 FP4a, V8.3 FP6, V9
Oracle	9.2.0.7 10.1.0.4 10.2.0.1
MS SQL Server	2000 SP4 2005
Sybase	12.5.2 15.0

Source : [List of supported software for WebSphere Application Server V6.1](#)

¹Voir Infocenter : [Direct and indirect JNDI lookup methods for data sources](#)

Serveur LDAP

La sécurité Websphere permet l'utilisation d'un serveur LDAP pour stocker les profils utilisateurs. Les architectures hautement disponibles devraient utiliser le mécanisme de clustering LDAP introduit dans la version 6.0. Il n'est alors plus nécessaire d'utiliser une solution externe telle qu'un répartiteur IP pour mettre en place la haute disponibilité LDAP.

Voici les pré-requis logiciels pour les principaux serveurs LDAP du marché :

Serveur LDAP	Pré-requis
IBM Tivoli® Directory Server	5.2
	6.0
IBM z/OS Security Server	1.6
	1.7
Lotus® Domino® Enterprise Server	6.5.4
	7.0
Novell eDirectory	8.7.3
	8.8
Sun ONE Directory Server	5.1 SP4
	5.2
Windows Active Directory	2000
	2003

Serveurs JMS

Les APIs d'accès aux middlewares de messages (MOM) ont connu une évolution très similaire à celles des drivers JDBC. Les premiers connecteurs de MOM en Java reposaient sur des bibliothèques au style de programmation C qui faisaient appel aux libraires natives (Websphere MQ Client ...) ; cette situation était comparable aux premiers driver JDBC de type 2 qui reposaient sur les libraires natives des SGBD (Oracle Call Interface / OCI ...).

Aujourd'hui, comme les drivers JDBC de type 4 remplacent leurs prédécesseurs de type 2, les connecteurs JMS d'accès à des MOM sont matures et vont remplacer les connecteurs historiques.

Il est de plus intéressant de noter que l'API JMS se répand au delà du monde Java et que les éditeurs de MOM développent des API similaires sur d'autres langages comme C++ ou .Net (en particulier Websphere MQ avec XMS¹ et Tibco EMS avec ses API .Net, C++ et Cobol).

Websphere offre désormais une forte intégration à JMS et les bénéfices de cette API sont nombreux pour les projets. En premier lieu, JMS offre tous les atouts d'un standard : découplage entre les éditeurs, qui se concentrent sur le développement du connecteur et les développeurs applicatifs, qui n'ont qu'une seule API à apprendre ; documentation pléthorique ; profusion des bibliothèques supportant le standard (en particulier le framework Spring) ; disponibilité de ressources formées sur le marché des développeurs Java. Il est d'ailleurs possible d'utiliser une implémentation JMS alternative lors des phases de développement afin de simplifier les tests unitaires (Apache ActiveMQ est à ce titre très intéressant). L'introduction dans JMS 1.1 d'une API unifiée pour les modèles point-à-point et publication-souscription améliore de surcroît l'évolutivité des applications.

Notons enfin que JMS est désormais intégré aux fonctionnalités d'administration et de supervision de Websphere, et qu'il est possible d'évoluer vers Embedded Messaging Engine de Websphere (SIBus) pour remplacer Websphere MQ.

Voici les pré-requis logiciels pour l'intégration de Websphere MQ :

Logiciel	Pré-requis	Commentaires
IBM Websphere MQ	5.3.12 6.0.1.1	La fonctionnalité de Publish-Subscribe disponible sous forme de support-pack pour Websphere MQ 5.3 est désormais incluse à Websphere MQ 5.3.12 et 6.x

¹Voir [Introducing XMS - The IBM Message Service API](#) in IBM DeveloperWorks

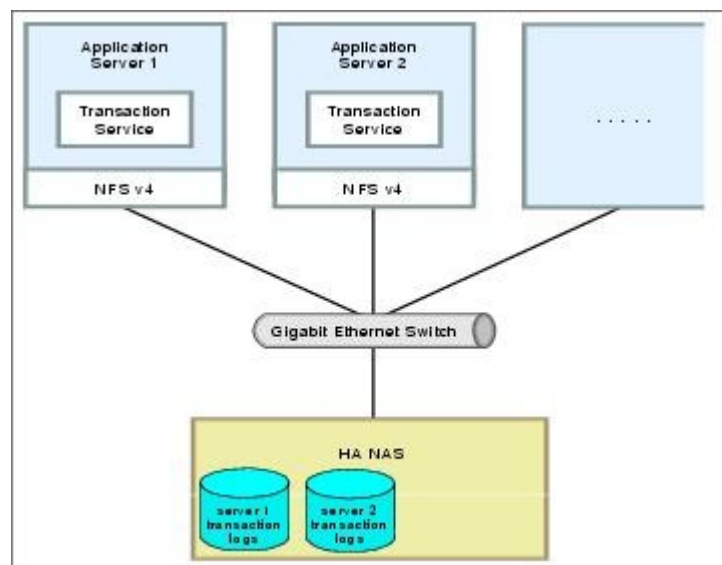
Haute disponibilité

Websphere 6.1 propose désormais d'emblée la haute disponibilité pour tous les services critiques. Le seul service qui n'offre pas de reprise sur panne transparent est le Deployment Manager pour ses fonctions de modification de la configuration et de déploiement des applications ; ce service n'est cependant pas critique pour l'exécution des applications déjà déployées.

Websphere 6 a introduit le 'hot standby' et le 'peer failover' des services centraux critiques (routage WLM, moteur de Messages JMS, gestionnaire de transactions ...). Le gestionnaire de haute disponibilité (High Availability Manager) exécute les services critiques (dont lui-même) sur un serveur disponible; si ce serveur devient indisponible, le service est transféré automatiquement sur un autre serveur disponible.

Par ailleurs, la version 6 a introduit le support de la haute disponibilité transparente du serveur LDAP grâce à un mécanisme de clustering¹. Il n'est donc plus nécessaire de recourir à des systèmes de haute disponibilité par un répartiteur IP comme c'était le cas sur les versions précédentes.

L'utilisation des transactions distribuées se répand avec l'essor des architectures SOA. Un scénario désormais fréquent est l'utilisation de transaction distribuées JDBC et JMS. Websphere 6 réduit grandement le temps de restauration des transactions distribuées en cas de panne d'un serveur grâce à l'introduction d'un nouveau mécanisme de 'failover' du Gestionnaire de Transaction basé sur un simple répertoire partagé².



[WAS ND V6 High Availability - Transaction service high availability using NAS](#)

Pour aller plus loin sur les aspects de haute disponibilité :

- IBM Education Assistant: [IBM WebSphere Application Server Workload management and high availability](#)
- IBM Redbook: [WAS ND v6 High Availability](#)

¹Voir Infocenter : [Security failover among multiple LDAP servers](#)

²Voir Redbook [WAS ND v6 High Availability](#) chapitre [6.7 Transaction Manager high availability](#)

Impacts sur l'exploitation et l'administration

Procédure d'installation de Websphere

La V6 a introduit la [Websphere Installation Factory](#) pour créer des packages d'installation de Websphere Application Server. Ces packages peuvent être composés des binaires avec des fixpacks, des paramètres de configuration et des EARs pré-déployés. Cet outil devrait être utilisé sur les grandes topologies Websphere qui nécessitent des installations répétées de serveurs. Dans ces scénarios, Websphere Installation Factory offrira une productivité accrue et une plus grande fiabilité grâce à l'automatisation du processus et la diminution du risque de facteur humain inhérent aux procédures d'installation complexes.

La nouvelle console d'administration Web

La nouvelle console d'administration offre la même ergonomie que celle de la version 5.1 avec une apparence visuelle nouvelle liée à la modularité de la console. Cette modularité permet à la gamme des produits Websphere d'utiliser une console d'administration unifiée dans laquelle les middlewares (serveur d'application, ESB, portail ...) s'intègrent sous forme de plugins.

Scripts d'administration

Les opérations d'administration de Websphere peuvent être réalisées par scripts aussi bien que par la console d'administration.

Les scripts offrent de nombreux avantages :

- Traçabilité : les scripts et leurs logs d'exécution peuvent être archivés.
- Répétabilité : les scripts peuvent être rejoués.
- Fiabilité : une fois les scripts validés (par exemple les scripts de déploiement d'applications), ils réduisent fortement le risque du facteur humain dans les procédures d'exploitation
- Productivité : l'exécution de scripts est beaucoup plus efficace que de déléguer des tâches d'administration à faible valeur ajoutée à des humains. Ce gain est particulièrement sensible lors des processus de build/déploiement.

Jython remplace désormais JACL comme langage de scripting privilégié. Jython est le langage de scripting privilégié de Weblogic et Websphere. Ce langage est en général perçu comme beaucoup plus facile d'utilisation que JACL (dérivé de TCL)

Les objets **AdminApp** et **AdminTask** (**AdminTask** a été introduit en V6) sont les APIs privilégiées de scripting de Websphere. Elles offrent des commandes orientées tâches de haut niveau quand **AdminConfig** est une API de plus bas niveau par laquelle on manipule directement les documents de configuration de Websphere.

Websphere offre un haut niveau de compatibilité des scripts entre les versions 5.x et 6.1.

- Compatibilité des langages de scripting :
 - JACL a été migré de la version 1.2.6 à la version 1.3.2. Cette nouvelle version introduit le support de la syntaxe d'expressions régulières de TCL qui n'est pas compatible ascendante avec les motifs d'expression régulières de la version 1.2.6
 - Jython a été migré de la version 2.1a3 à la version 2.1. Cette migration est compatible ascendante
- Changements des objets de configuration
 - L'attribut `transactionLogDirectory` a été déplacé de `ApplicationServer.TransactionService` à `ServerEntry.recoveryLog`. L'ancienne valeur reste disponible tant que la nouvelle valeur n'est pas modifiée

V5.x Jython syntax

```
wsadmin> txService = AdminConfig.list('TransactionService', server1)
wsadmin> txLogDirectory =
AdminConfig.showAttribute(txService, 'transactionLogDirectory')
```

V6.1 Jython syntax

```
wsadmin>serverEntry = AdminConfig.list('ServerEntry')
...
wsadmin> recoveryLog = AdminConfig.showAttribute(serverEntry, 'recoveryLog')
wsadmin> txLogDirectory = AdminConfig.showAttribute(recoveryLog,
'transactionLogDirectory')
```

- L'attribut `processDefinition` de l'objet `Server` a été renommé en `processDefinitions` qui est devenu une liste d'ID séparés par des caractères espaces (e.g. "[id1 id2 ... id~~x~~]").

V5.x Jython syntax

```
wsadmin>javaProcessDef = AdminConfig.showAttribute(server, 'processDefinition')
```

V6.1 Jython syntax

```
wsadmin>javaProcessDefsAsString = AdminConfig.showAttribute(server,
'processDefinitions')
wsadmin>javaProcessDefs = javaProcessDefsAsString[1:len(javaProcessDefsAsString) -
1].split(' ')
wsadmin>javaProcessDef = javaProcessDef[0]
```

Outils de diagnostique et de résolution des incidents

La principale évolution de la version 6.1 est la consolidation des outils de diagnostique et de résolution des incidents dans la console d'administration Web et dans le nouveau [IBM Support Assistant](#). Les outils spécifiques Websphere sont dorénavant tous disponibles au travers de la console d'administration Web et les outils génériques java sont eux regroupés dans IBM Support Assistant.

Les outils de diagnostic disponibles dans la console sont les suivants :

- **Fournisseurs de diagnostique**

Ces outils fournissent les informations de diagnostique de chacun des composants du serveur Websphere (données d'état, données de configuration et tests d'auto-diagnostique). Le nombre de ces outils est amené à augmenter avec les versions de Websphere. Ces outils sont similaires aux Defaults Tests disponibles dans Websphere MQ 6.

- **Tivoli Performance Viewer (TPV)**

Outil de visualisation des données de performances collectées par la l'infrastructure PMI (Performance Monitoring Infrastructure). TPV a migré d'un client lourd à une interface web intégrée dans la console d'administration. Le gain est une facilité accrue d'utilisation (il n'est plus nécessaire d'installer l'application cliente et de gérer les problèmes de firewalls) ; cependant, l'ergonomie n'est pas encore au niveau de celle qu'offrait la version précédente qui était distribuée sous forme de client lourd (c'est particulièrement le cas lorsqu'on suit un nombre important d'indicateurs).

- **Visionneuse de Classloader**

Cet outil permet de diagnostiquer les problèmes de classpath et de chargement de classe en affichant les jars chargés par les classloaders hiérarchiques d'une application. La visionneuse de classloader optionnelle en V5 est désormais intégré en standard à la version 6.

Les outils de diagnostic disponibles dans IBM Support Assistant sont les suivants :

▪ **[IBM Support Assistant \(ISA\)](#)**

ISA est le nouveau socle basé sur Eclipse des outils de diagnostic et de support IBM

- Regroupe les outils de diagnostics (pour Websphere, DB2, Rational ...)
- Offre un point d'entrée de recherche unifié pour les informations sur les produits IBM (fixes, technotes, infocenters ...)
- Facilite la communication avec le support IBM grâce à un assistant de création de PMR

▪ **[IBM Heap Analyzer \(aka Memory Dump Diagnostic for Java\)](#)**

Outil de diagnostic des fuites mémoire.

▪ **[IBM Pattern Modeling and Analysis Tool for Java Garbage Collector \(PMAT\)](#)**

Outil d'aide au paramétrage de la Heap des JVM et de diagnostic des problèmes de fragmentation du Heap. PMAT va être remplacé par le nouveau [Extensible Verbose GC Toolkit](#).

▪ **[IBM Thread and Monitor Dump Analyzer for Java Technology](#)**

Outil de diagnostic des problèmes de threads des JVM (threads suspendus, inter-verrous, goulets d'étranglement ...).

▪ **[Extensible Verbose GC Toolkit](#)**

EVTK est le successeur de PMAT. EVTK a été conçu comme un plugin de ISA, il est donc parfaitement intégré et préfigure l'ergonomie qu'offriront les futurs outils de troubleshooting IBM.

Impacts sur les développements

Quel environnement de développement (IDE) ?

Si Websphere 5.X était naturellement associé à WSAD , Websphere 6.1 nécessite d'utiliser un nouvel environnement de développement dont le choix est cette fois ci ouvert.

Rational Application Developer (RAD) est le successeur naturel de Websphere Studio Application Developer (WSAD). Dans la lignée de WSAD, RAD est l'IDE qui offre tous les assistants pour exploiter les fonctionnalités spécifiques de Websphere (EJB, Portlet, JSF, SOAP ...).

Cependant, la place de RAD est complètement différente aujourd'hui de celle que WSAD occupait à son époque. La banalisation des standards J2EE, le succès du lightweight J2EE (Struts/Spring/Hibernate) et l'essor d'Eclipse ont stimulé le marché des IDEs qui n'est plus le domaine réservé des éditeurs haut de gamme et des produits coûteux.

Eclipse Web Tools Project est devenu un IDE mature qui répond aux besoins de beaucoup de projets lightweight J2EE (Struts/Spring/Hibernate). Il est reconnu pour sa simplicité d'utilisation, sa gratuité et sa popularité auprès des développeurs.

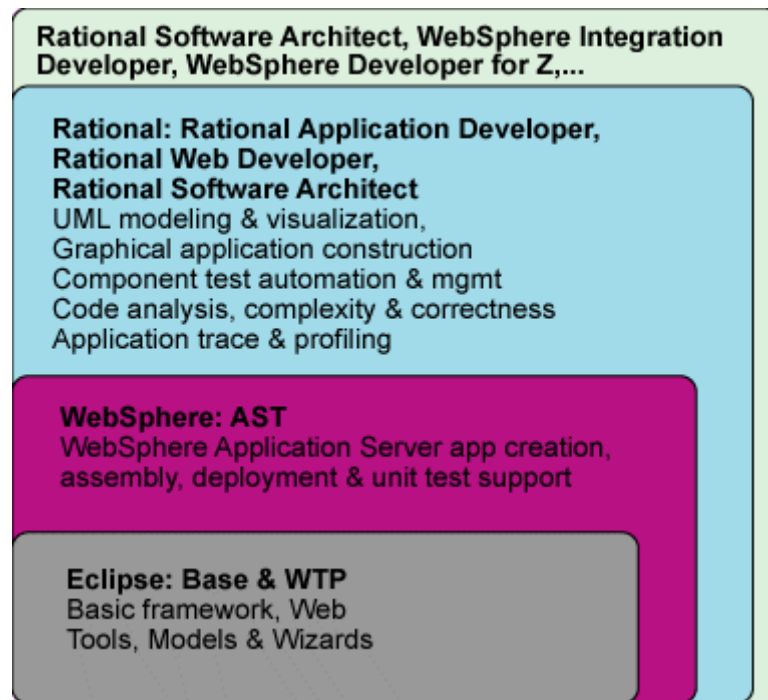
Pour répondre au succès d'Eclipse Web Tools Project, IBM a introduit Websphere Application Toolkit, un IDE intermédiaire entre Eclipse et le sophistiqué et onéreux RAD. Websphere AST peut être vu comme une Eclipse Web Tools Project avec les plugins spécifiques Websphere comme des éditeurs de fichier de configuration spécifiques Websphere et environnement de scripting wsadmin (voir la matrice des fonctionnalités infra). Cependant, l'absence d'un Tomcat Test Environment diminue sensiblement l'intérêt de Websphere AST comme IDE principal d'une équipe de développement.

L'appropriation du nouvel IDE (RAD, AST ou Eclipse) par les équipes de développement sera facilité du fait qu'ils disposent tous de l'ergonomie proposée par le projet Eclipse.

La richesse et la complémentarité de ces IDE et l'omniprésence d'Eclipse permet aujourd'hui d'équiper les équipes de développement d'un mix d'Eclipse, de Websphere AST et de RAD qui seront utilisés en fonctions des tâches à accomplir.

Ainsi, les projets Lightweight J2EE (Struts/Spring/Hibernate) pourront utiliser majoritairement Eclipse Web Tools Project complété de quelques postes Websphere AST voire RAD pour les tâches spécifiques Websphere. Spring, Hibernate et Struts étant validés aussi bien sur Websphere que sur Tomcat, il est possible de développer sur Tomcat Test Environment et d'intégrer sur Websphere.

En revanche, les projets 100% pure Websphere (Websphere-JSF, Websphere-SOAP, EJB ...) utiliseront principalement RAD pour apporter un Websphere Test Environment à chaque développeur. Eclipse sera alors utilisé à la marge pour des tâches très ponctuelles.



Famille des environnements de développement IBM

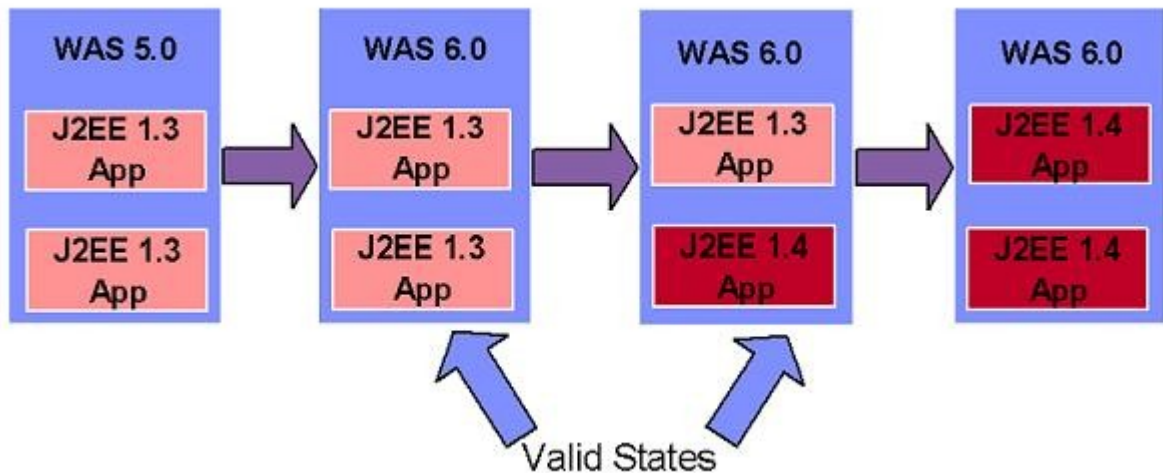
Cette matrice présente les fonctionnalités des différents IDE :

	Eclipse Web Tools Project	Websphere Application Server Toolkit (AST)	Rational Application Developer (RAD)	Commentaires
Java	✓	✓	✓	
JSP, Servlet, EJB	✓	✓	✓	
Java Server Faces (JSF)	✗ Release 1.0 annoncée pour Juillet 2007	✗	✓ <i>avec le support des spécificités Websphere</i>	RAD est le seul IDE qui exploite pleinement les spécificités de l'implémentation Websphere JSF
Tomcat Test Environment	✓	✗	✓	Nécessite l'installation de Tomcat sur le poste de travail
Websphere 6.0 Test Environment	? <i>Nécessite d'installer Websphere sur le poste de travail</i>	? <i>Nécessite d'installer Websphere sur le poste de travail</i>	✓	RAD est le seul IDE qui intègre un runtime Websphere 6.0. Une licence Websphere est nécessaire pour AST et Eclipse WTP
Websphere 6.1 Test Environment	✗ <i>Prévu pour WTP 2.0 annoncé en 3Q 2007</i>	? <i>Nécessite d'installer Websphere sur le poste de travail</i>	✓	RAD est le seul IDE qui intègre un runtime Websphere 6.1. Une licence Websphere est nécessaire pour AST et Eclipse WTP
Éditeurs graphiques pour les fichiers de configuration spécifiques WebSphere	✗	✓	✓	
Environnement de scripting Jython Websphere	✗	✓	✓	
Support des spécificités Websphere SIP & Portlet	✗	✓	✓	
Accès aux bases de données	✓	✓	✓	

	Eclipse Web Tools Project	Websphere Application Server Toolkit (AST)	Rational Application Developer (RAD)	Commentaires
Outils de contrôle de la qualité du code	✓	✓	✓	Des plugins open source complémentaires sont disponibles (CheckStyle, FindBugs ...)
Profiling	?	?	✓	Eclipse TPTP n'est pas encore au niveau des produits commerciaux
Modélisation UML	✗	✗	✗ Seulement disponible avec Rational Software Architect	
Cycle de nouvelles versions	Bug fréquents par 'Eclipse update' mais transparents pour les développeurs	Calme Suffisamment souvent pour bénéficier des dernières améliorations d'Eclipse sans prendre de vitesse les équipes de développement	Lent Pendant plusieurs mois, RAD ne supportait pas le nouveau Websphere 6.1	
Configuration matériel requise	Peu consommateur en CPU et mémoire	Peu consommateur en CPU et en RAM (similaire à Eclipse WTP)	Très consommateur en CPU et RAM. Un poste de travail récent et haut de gamme avec au 2 Go de RAM est nécessaire	
Prix	Gratuit et Open Source. Support commercial disponible auprès d'IBM et d'autres acteurs	Gratuit pour les clients WAS 6.1	Commercial. Plusieurs K€	IBM source : « The AST is licensed as a component part of WebSphere Application Server. Unlimited copies can be made provided the AST is used for developing applications for WebSphere Application Server V6.1 ».

Migration des API J2EE et des librairies Websphere

Les serveurs d'applications J2EE assurent une très grande compatibilité ascendante et permettent de déployer des EARs de différentes versions (J2EE 1.3, J2EE 1.4 ...) sur un même serveur.



[WAS V6 - Migration Guide / 3.1 J2EE compatibility / 3.1.1 Incremental migration](#)

Le point de vigilance concerne les bibliothèques embarquées par Websphere. Les applications devraient en théorie être qualifiées avec les nouvelles versions de librairies embarquées par Websphere (Xalan, Xerces ...). Cependant, ces librairies sont très matures et une utilisation « dans leur esprit » ne devrait pas présenter de problème de compatibilité ascendante ; il est donc souvent suffisant de valider ces nouvelles librairies lors du test de non régression globale de passage à la V6.1 sur l'environnement de recette/intégration.



JDom et Websphere 6.1

JDom n'est plus embarqué dans WAS 6.1. Voici un plan de migration en deux étapes pour les applications qui utilisent JDom sans l'embarquer dans leur EAR :

- Déclarer jdom.jar comme librairie partagée Websphere et lier l'application à cette librairie (il n'est pas nécessaire de repackager l'EAR)
- Ajouter jdom.jar comme dépendance de l'application et déployer le nouvel EAR qui embarque jdom.jar.

Stratégie de migration

Stratégie de migration pour l'exploitation

Migration des scripts d'administration

La migration des scripts d'administration doit être réalisée avant l'installation du Deployment Manager sur lequel ces scripts sont utilisés.

La forte compatibilité ascendante des APIs d'administration diminue les risques liés à cette opération mais il est sécurisant d'anticiper l'éventuelle nécessité de recourir à des procédures d'exploitation manuelle en attendant la correction des derniers problèmes sur les scripts d'administration.

L'environnement de prototypage est particulièrement adapté à la validation des nouveaux scripts.

Stratégie de migration de l'infrastructure

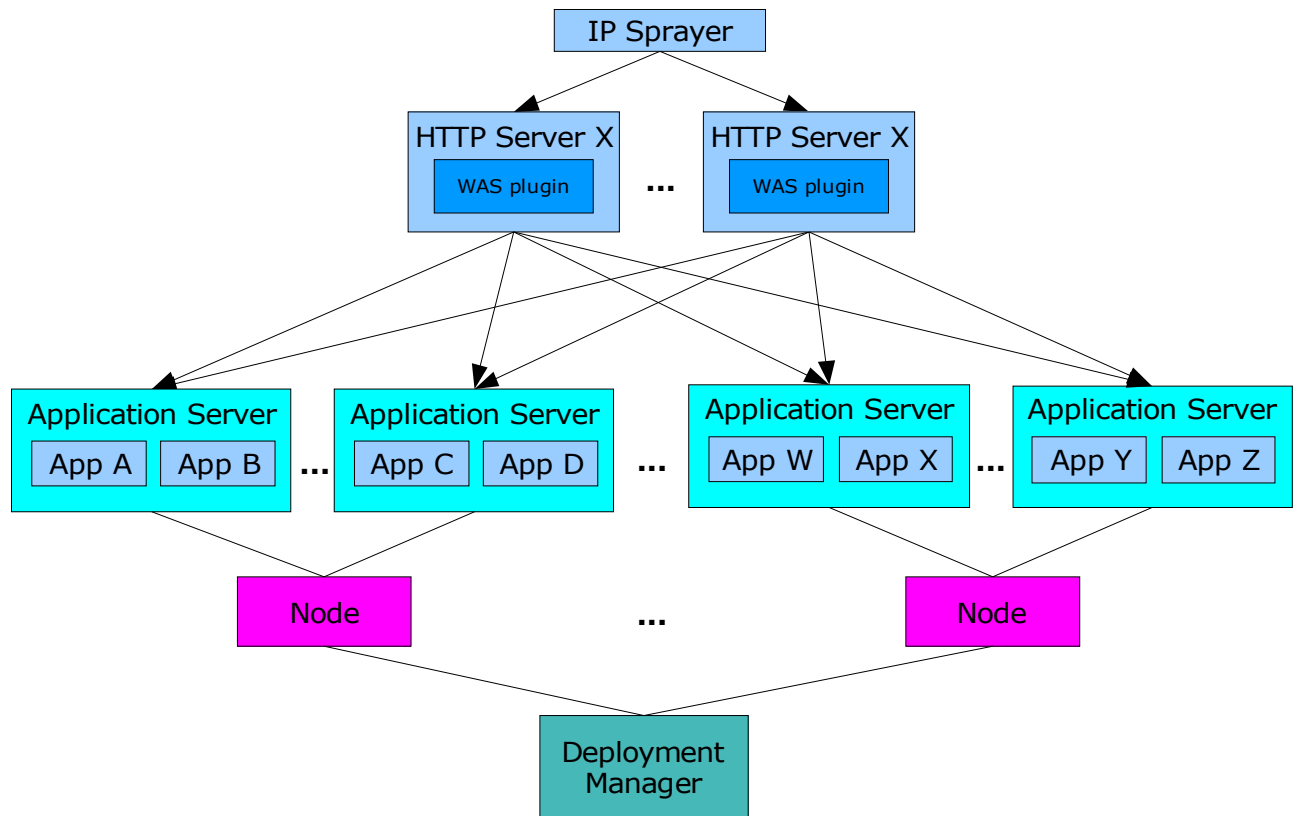
Scénario simplifié de migration de serveurs isolés

Ce scénario simplifié ne décrit pas les opérations de migration des outils de monitoring, de sauvegarde & restauration. C'est le scénario du cas simple d'un serveur non fédéré dans une cellule Websphere Network Deployment.

- Installation des binaires de Websphere 6.1 à côté de ceux de la version existante
- Application du dernier fixpack de Websphere 6.1
- Configuration des outils d'exploitation (sauvegarde & restauration, supervision ...)
- Utilisation du "Migration Wizard" pour configurer Websphere 6.1
- Arrêt de l'ancienne version de Websphere et démarrage de la version 6.1
- Activation des outils d'exploitation
- Lorsque la nouvelle version est stable (plusieurs semaines), désinstaller l'ancienne version

Scénario de migration d'une cellule

Une cellule Websphere est composée d'un Deployment Manager, d'un ou de plusieurs noeuds, d'un ou plusieurs serveurs et éventuellement de serveurs web.



Principaux composants d'une cellule Websphere Network Deployment

- **Big Bang**

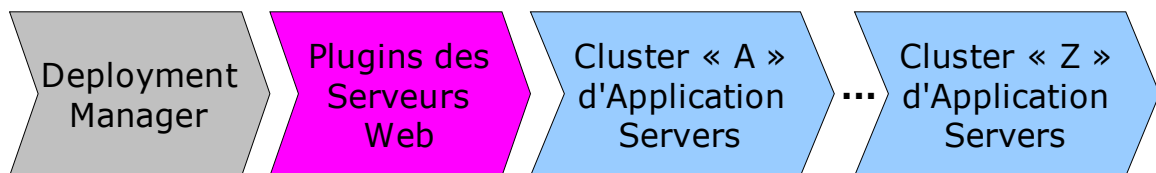
Le scénario de migration par Big Bang mettant à jour tous les serveurs simultanément est fortement déconseillé car cette approche est inutilement risquée. Websphere est conçu pour permettre une migration progressive avec la possibilité de faire coexister des serveurs 5.x et 6.x dans une cellule 6.1. De plus, il est possible de revenir en arrière sur la plupart des opérations de migration et de redémarrer le composant problématique en version 5.x.

■ Migration progressive

La migration progressive est le scénario privilégié. Il minimise les risques de régression et permet aux équipes d'avancer à leur rythme. Les grandes étapes d'une migration progressive sont :

- Migration du Deployment Manager
- Migration des plugins des serveurs web
- Migration des serveurs avec une granularité au cluster (tous les serveurs d'un cluster simultanément, les clusters sont migrés les uns après les autres)

💡 Attendre qu'un cluster soit stable avant de migrer le suivant.



Étapes de migration d'une cellule Websphere

Stratégie de migration des applications

Migration des IDE

Websphere permettant de déployer des EARs des précédentes versions J2EE (1.2 et 1.3), la migration de l'IDE et des processus de développement n'est pas contrainte à être strictement en phase avec la migration de l'infrastructure.

La migration des IDE se déroule dans un processus classique de définition des besoins suivie de l'évaluation et du prototypage puis enfin du déploiement dans les équipes.

⚠ Le principal point de vigilance est la migration des projets J2EE depuis Websphere Studio 5.x vers RAD, AST ou Eclipse. IBM fournit une documentation détaillée sur [Rational Application Developer Infocenter : Migrating from WebSphere® Studio V5.1, 5.1.1, or 5.1.2.](#)

💡 Pour les migrations vers un mix Eclipse/AST/RAD, l'évaluation précise de la proportion entre les différents IDE dès le début du projet n'est pas fondamentale en termes de gestion du changement. Il est toujours possible de réajuster l'équilibre RAD/AST/Eclipse lors du déploiement dans les équipes. Si un rééquilibrage entre AST et Eclipse sera transparent, en revanche, l'ajout de postes RAD aura un impact sur le coût en licences et la puissance des postes de développement (Eclipse et AST sont tous deux gratuits et ont des consommations de ressources CPU et RAM similaires).

Mise à jour des APIs J2EE

La mise à jour des API J2EE sera faite lors de l'évolution des applications si cette migration s'avère nécessaire. Il n'y a aucune urgence à procéder à cette évolution.

IBM fournit dans Websphere AST et dans RAD un J2EE Application Migration Wizard qui simplifiera la tâche des équipes de développement.

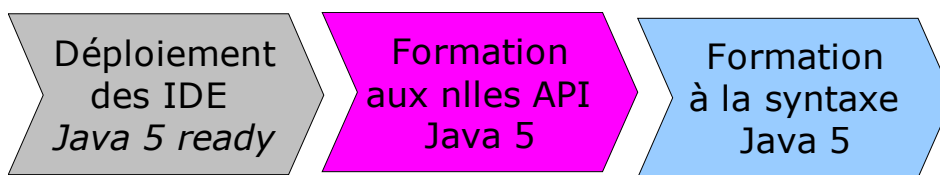
Migration à Java 5

⚠ Si le passage à Java 5 n'est pas un pré-requis de la migration à Websphere 6.1, en revanche, ce volet doit être traité dès la fourniture des nouveaux IDEs aux équipes de développement qui sinon commenceront à utiliser les nouvelles fonctionnalités Java 5 hors de tout accompagnement. On risquerait alors une perte d'homogénéité des programmes et l'utilisation de nouvelles technologies « pour se faire plaisir » plutôt que par une réelle pertinence projet.

La migration à Java 5 est essentiellement un projet de formation des équipes et d'accompagnement. Les grandes étapes en sont :

- Déploiement des IDEs « Java 5 ready » dans les équipes de développement. L'amélioration de l'ergonomie de ces IDE de dernière génération aura un impact fort sur la productivité et la satisfaction des équipes de développement.
- Formation à l'utilisation des nouvelles bibliothèques de Java 5 (util.concurrent ...) sans utiliser encore les nouvelles syntaxes Java 5.
- Enfin, formation aux nouvelles syntaxes Java 5 (génériques, annotations, types énumérés ...). Les plus grands gains seront les annotations (Hibernate et Spring) et les génériques pour clarifier les listes (Hibernate).

Cette approche permettra d'accompagner les équipes de développements pour garantir la cohérence de l'utilisation de ces nouvelles fonctionnalités.



Plan de migration à Java 5

Planning de migration / séquençement

Phase de Prototypage

La phase de prototypage servira avant tout à valider les points nécessaires à la migration en environnement de développement.

La phase de prototypage peut se focaliser sur la validation des infrastructures et des procédures d'exploitation de la nouvelle plate-forme.

Rôles de l'environnement de prototypage :

- Découverte de Websphere 6.1
- Validation des nouvelles procédures d'exploitation (scripts ...)
- Validation des outils d'exploitation (sauvegarde et restauration, monitoring, troubleshooting ...)

La migration des applications peut être directement réalisée sur le nouvel environnement de développement. Ce raccourci ne présente pas de risque particulier pour le projet car la probabilité de problème bloquant lors de la migration des applications est très faible ; les serveurs J2EE offrent une grande compatibilité ascendante pour les applications et il s'agit en général d'ajustement mineurs (montée de version de JDom ...).

Supprimer la phase de prototypage pour la migration des applications présente l'avantage de raccourcir la durée de la migration et d'économiser des ressources matérielles et logicielles.

Le prototypage sera réalisé sur une petite grappe de serveurs isolés qui reproduiront la topologie Websphere de production. Il ne faudra pas oublier de valider des points d'infrastructure comme les serveurs d'application clusterisés ou la connexion aux serveurs LDAP.

Phase de déploiement

Les environnements (développement, intégration, pré-production puis production) doivent être migrés l'un après l'autre en exécutant les étapes suivantes :

Mise à jour des serveurs web

- Si nécessaire, mise à jour du serveur web lui même (IHS devrait être migré en V6.1) :
 - Installation des binaires du serveur web V6.1.x
 - Configuration du nouveau serveur
 - Configuration des outils d'exploitation (sauvegarde & restauration, supervision ...) du serveur web v6.1: mettre à jour les outils si nécessaire puis configurer
 - Arrêt de l'ancien serveur web
 - Démarrage du serveur web v6.1
 - Mise à jour du plugin du serveur web
 - Installer les plugins v6.1 sur les serveurs web
 - Configurer le serveur web pour utiliser le plugin v6.1
 - Redémarrer le serveur web pour qu'il utilise le nouveau plugin.
 - La configuration des plugins des serveurs web doit être régénérée et redéployée lors de la migration de chaque noeud

Mise à jour du Deployment Manager

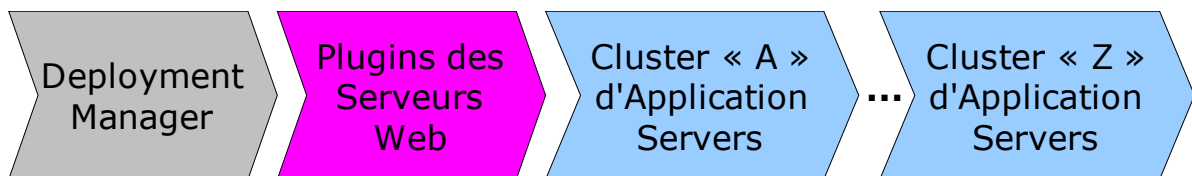
- Installation des binaires V6.1.x du Deployment Manager à côté de ceux de l'ancienne version
- Configuration des outils d'exploitation (sauvegarde & restauration, supervision ...) du Deployment Manager v6.1: mettre à jour les outils si nécessaire puis configurer
- Configuration du Deployment Manager v6.1 en important la configuration du Deployment Manager v5.x/6.0 avec le Websphere Migration Wizard
- Vérification des logs de l'exécution du Websphere Migration Wizard
- Arrêt du Deployment Manager V5.x/V6.0
- Démarrage du Deployment Manager v6.1
- Activation des outils d'exploitation du Deployment Manager (sauvegarde & restauration, supervision)
- Régénération et redéploiement de la configuration des plugins des serveurs web si le Deployment Manager est accédé au travers de serveurs web
- Tests

💡 Si la migration a échoué, il suffit de redémarrer l'ancien Deployment Manager v5.x/6.0 pour revenir en arrière

Mise à jour des nœuds l'un après l'autre

- Installer les binaires de la version 6.1.x à côté de ceux de l'ancienne version
- Configuration des outils d'exploitation (sauvegarde & restauration ...) du nœud et des serveurs d'application : mettre à jour les outils si nécessaire puis configurer
- Configuration du nœud et des serveurs d'application v6.1 en important la configuration du nœud et des serveurs d'application v5.x/6.0 avec le Websphere Migration Wizard
- Vérification des logs de l'exécution du Websphere Migration Wizard
- Arrêt du nœud et des serveurs d'application V5.x/V6.0
- Démarrage du nœud et des serveurs d'application v6.1
- Activation des outils d'exploitation du Nœud (sauvegarde & restauration, supervision)
- Régénération de redéploiement de la configuration des plugins des serveurs web
- Tests

💡 Si la migration a échoué, il suffit de redémarrer les anciens nœuds et serveurs d'application v5.x/6.0 pour revenir en arrière



Phases de migration d'une cellule Websphere Network Deployment

Phase de rationalisation

Une fois les environnements migrés, la phase de rationalisation peut démarrer. C'est une phase itérative dont le but est de simplifier et de clarifier l'architecture en suivant les bonnes pratiques. Cette phase est une opportunité de baisser le Coût Total de Possession et d'améliorer la disponibilité de la plate forme.

Cette phase est aussi l'occasion de préparer l'avenir et en particulier le passage à Java EE 5 avec ses EJB3 et son nouveau framework de persistance Java Persistence API (JPA).

Si JBoss, Weblogic et Websphere Community Edition (Apache Geronimo) supportent déjà Java EE 5, il faudra attendre la version 7 de Websphere qui est annoncée pour le premier trimestre 2008. Cependant, il est déjà possible de préparer cette évolution en utilisant des frameworks qui mettent déjà en oeuvre les API ou les principes JavaEE 5 :

- Utiliser [Hibernate](#) ou [Apache OpenJPA](#) pour la persistance en utilisant l'API `javax.persistence.EntityManager`
- Utiliser un framework d'assemblages des composants comme [SpringFramework](#)

Bibliographie

IBM propose une documentation détaillée mais parfois complexe d'accès sur Websphere et en particulier sur la migration de version. Les principales sources sont l'[Infocenter Websphere](#), les [Redbooks IBM](#) et les articles du site [IBM DeveloperWorks Websphere](#).

Voici une liste des documents de référence pour mener un projet de migration Websphere.

- IBM Education: Websphere Application Server 6.1 - Migration to V6.1
http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/InstallationAndMigration/WASv61_Migration/player.html
- Redbook: WebSphere Application Server V6 Migration Guide
<http://www.redbooks.ibm.com/abstracts/sg246369.html?Open>
- Deprecated and removed features in Websphere InfoCenter
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html
- A quick guide for migrating to WebSphere Application Server V6.1 by IBM Software Services for WebSphere in IBM DeveloperWorks
http://www-128.ibm.com/developerworks/websphere/library/techarticles/0608_chalmers/0608_chalmers.html
- WebSphere Application Server V6.1: What's new in Version 6.1? by IBM Software Services for WebSphere in IBM DeveloperWorks
http://www-128.ibm.com/developerworks/websphere/library/techarticles/0606_petersonr/0606_petersonr.html
- Recommended reading list: J2EE and WebSphere Application Server by IBM Software Services for WebSphere in IBM DeveloperWorks
http://www-128.ibm.com/developerworks/websphere/library/techarticles/0305_issw/recommendedreading.html

Conclusion

La migration vers Websphere 6.1 en vaut-elle la peine ? Au-delà des problématiques de fin de support, la question se pose légitimement. Certes Websphere 6.1 apporte Java 5 et les gains de productivité que l'on peut en attendre ; il vient également avec son lot d'amélioration en matière d'infrastructure, d'administration et de supervision. Mais les API J2EE n'ont évolué qu'à la marge et les innovations du produit, aussi valables soient-elles, sont et restent des changements, avec leur lot de problèmes intrinsèques. Ajoutons que Websphere 7 devrait arriver rapidement en 2008 pour proposer le support de Java EE 5 – que les concurrents d'IBM (en particulier BEA, Jboss et Apache Geronimo) proposent depuis plusieurs mois pour certains.

D'une façon générale, c'est le principe même des middlewares monolithiques qui est mis en cause. Dans le monde J2EE, pour bénéficier d'une innovation même mineure de la spécification, il est nécessaire de migrer sa plateforme complète – avec un niveau de complexité et de risque qui augmente à mesure que la criticité des applications déployées s'accroît. Cette particularité de J2EE semble de moins en moins justifiable.

Dès juillet 2005, Billy Newport, l'un des architectes de Websphere, appelait à la modularisation de Websphere sur le modèle de Linux ou de certaines bibliothèques Java (voir son article : [End of the road for invasive middleware?](#)). Cette tendance à la modularité est confirmée par l'annonce du Gartner Group en février 2007 selon laquelle Vista, le nouveau système d'exploitation de Microsoft, serait la dernière version monolithique de Windows.

■ Websphere 7 sera-t-il le dernier Websphere monolithique ?

Annexe 1 - Matrices des versions d'APIs et bibliothèques Websphere

Versions des API java supportées par Websphere Application Server

API	Websphere 3.5	Websphere 4.0	Websphere 5.0	Websphere 5.1	Websphere 6.0	Websphere 6.1	Commentaires
Java	1.2	1.3	1.3	1.4	1.4	5.0	
J2EE		1.2	1.3	1.3	1.4	1.4	
JDBC	1.1/2.0	2.0	2.0	2.0	3.0	3.0	
JAF		1.0	1.0	1.0	1.0	1.0	
Java Mail		1.1	1.2	1.2	1.3	1.3	
JMX			1.1	1.1	1.2	1.2	
Servlet	2.1/2.2	2.2	2.3	2.3	2.4	2.4	
JSP	0.91/1.0/1.1	1.1	1.2	1.2	2.0	2.0	
EJB	1.0	1.1	1.1/2.0	2.1	2.1	2.1	
JMS		1.0	1.02	1.02	1.1	1.1	
JTA	1.01	1.01	1.01	1.01	1.01	1.01	
RMI/IIOP	1.0	1.0	1.0	1.0	1.0	1.0	
JNDI	1.2	1.2	1.2	1.2	1.2	1.2	
JAXP			1.1	1.1	1.2	1.2	
JCA		1.03	1.03	1.03	1.5	1.5	
JAAS			1.0	1.0	1.0	1.0	
JACC					1.0	1.0	
Timer (JSR 236)					1.1	1.1	JSR pas encore intégrée à la spécification J2EE
Workmanager (JSR 237)					1.1	1.1	
Web Services					1.1	1.1	

JAX-RPC					1.1	1.1	
SAAJ					1.2	1.2	
JAXR					1.0	1.0	
J2EE Management					1.0	1.0	
J2EE Deployment					1.1	1.1	

Versions des librairies tierces embarquées par Websphere application Server

Librairie	Websphere 3.5	Websphere 4.0	Websphere 5.0	Websphere 5.1	Websphere 6.0	Websphere 6.1	Commentaires
Xalan		LotusXSL 2.2	?	XSLT4J Java 2.6.1	?	XSLT4J Java 2.7.5	Xerces and Xalan locations and versions (technote)
Xerces		Xerces 1.4.2	XML4J 4.0.12	XML4J 4.3.1	?	XML4J 4.4.6	Xerces and Xalan locations and versions (technote)
Jdom			0.7 (aka 1.0 beta 7)	0.7 (aka 1.0 beta 7)	0.7 (aka 1.0 beta 7)	Plus embarqué dans Websphere	How to use the newest JDOM.jar? (technote)
Jakarta Commons Logging			1.0.3	1.0.3	1.0.3	1.0.3	
Jakarta Commons Discovery			0.2	0.2			

Annexe 2 - Fichiers et répertoires à prendre en compte lors des sauvegardes

L'ensemble des fichiers Websphere (binaires, configuration et fichiers temporaires) se situent sous le même répertoire parent <WAS_HOME> (par défaut « C:\Program Files\ibm\WebSphere\AppServer » sur Windows). Les outils de sauvegardent n'ont donc qu'à gérer une seule arborescence de répertoire en veillant à sauvegarder plus fréquemment la configuration et à exclure les fichiers temporaires.

Liste des fichiers de configuration :

Répertoire	Fichiers de configuration à sauvegarder
<WAS_HOME>/bin/	Seulement sauvegarder les scripts shell (*.bat et *.sh) pour les cas de modifications par les équipes d'exploitation
<WAS_HOME>/java/*	*.properties, *.policy, cacerts, *.security et *.access
<WAS_HOME>/profiles/<SERVER_NAME>/bin/	Seulement sauvegarder les scripts shell (*.bat et *.sh) pour les cas de modification
<WAS_HOME>/profiles/<SERVER_NAME>/config/	Toute l'arborescence du répertoire excepté les répertoires backup et temp
<WAS_HOME>/profiles/<SERVER_NAME>/configuration/	Toute l'arborescence du répertoire
<WAS_HOME>/profiles/<SERVER_NAME>/etc/	Toute l'arborescence du répertoire
<WAS_HOME>/profiles/<SERVER_NAME>/properties/	Toute l'arborescence du répertoire

Liste des fichiers temporaires à exclure des sauvegardes :

Répertoire à exclure des sauvegardes	Fichiers à exclure
<WAS_HOME>/logs	Logs de l'installation initiale, de l'installation des fixpacks et de la création des profiles
<WAS_HOME>/tmp	Répertoire temporairement de périmètre server d'application (fixpacks, création de profiles ...)
<WAS_HOME>/profiles/<SERVER_NAME>/logs	Répertoire de logs
<WAS_HOME>/profiles/<SERVER_NAME>/temp	Répertoire temporaire
<WAS_HOME>/profiles/<SERVER_NAME>/config/temp	Répertoire temporaire pour la configuration
<WAS_HOME>/profiles/<SERVER_NAME>/tranlog/	Fichiers de logs des transactions
<WAS_HOME>/profiles/<SERVER_NAME>/wstemp/	Répertoire temporaire utilisé pour stocker toutes les modifications de configuration jusqu'à leur sauvegarde (aussi bien via la console d'administration que par des scripts wsadmin)