

Objecteering

Version 6.0 Patch P08170: Notes de version

1. INTRODUCTION.....	3
2. INSTALLATION DU PATCH P08170 POUR OBJECTEERING 6.0 SP2 (P07430)	4
2.1. CONFIGURATION NECESSAIRE.....	4
2.2. PROCEDURE D'INSTALLATION WINDOWS.....	4
2.2.1. <i>Procédure de déploiement du patch</i>	4
2.2.2. <i>Fichiers déployés</i>	4
2.3. PROCEDURE D'INSTALLATION LINUX.....	5
2.3.1. <i>Procédure de déploiement du patch</i>	5
2.3.2. <i>Fichiers déployés</i>	5
3. EVOLUTIONS ET FAITS TECHNIQUES CORRIGES ..	6
3.1. OBJECTEERING 6.0 PATCH P08170	6
3.1.1. <i>UML Modeler</i>	6
3.1.2. <i>C++ Developer (CxxDeveloper)</i>	6
3.1.3. <i>Fortran Developer</i>	6
3.1.4. <i>Java Developer</i>	6
3.2. OBJECTEERING 6.0 PATCH P08040	6
3.2.1. <i>UML Modeler</i>	6
3.2.2. <i>C# Developer (CsDeveloper)</i>	6
3.2.3. <i>C++ Developer (CxxDeveloper)</i>	7
3.2.4. <i>Documentation</i>	7
3.2.5. <i>Java Developer</i>	7
3.2.6. <i>Requirements (RequirementsManager)</i>	7
3.2.7. <i>SQL Designer</i>	8

1. Introduction

Ce document vous fournit toutes les informations nécessaires pour votre prise en main de l'outil Objecteering 6.0. Il ne remplace pas la documentation utilisateur livrée avec Objecteering 6.0, mais apporte des informations importantes pour vous aider à démarrer vos travaux avec l'outil. Il vous est fortement recommandé de lire ce document avant de commencer à travailler avec Objecteering 6.0.

2. Installation du patch P08170 pour Objecteering 6.0 SP2 (P07430)

2.1. Configuration nécessaire

Objecteering 6.0 SP2 (P07430) avec le patch P08170 a été validé sur les plates-formes suivantes :

- Windows 2000 SP4 (x86)
- Windows XP SP1/SP2 (x86)
- Windows 2003 (serveur uniquement)
- Linux Red Hat 9 (x86)
- Linux Red Hat Enterprise 4.2
- Solaris 2.8, 9 et 10 (SPARC)(serveur uniquement)

2.2. Procédure d'installation Windows

Le patch Objecteering 6.0 P08170 nécessite une installation préalable d'Objecteering 6.0 SP2 (P07430). Il s'agit d'un patch cumulatif qui inclus les patchs précédemment applicables sur le SP2 (en l'occurrence le P08040).

2.2.1. Procédure de déploiement du patch

- Fermer tous les programmes Objecteering.
- Double cliquer sur le fichier " Objecteering6_P08170_WIN.exe".
- Cliquer sur le bouton "Next" pour installer le patch.
- Cliquer sur "Finish" pour terminer l'installation.

2.2.2. Fichiers déployés

Les fichiers suivants sont déployés par ce patch :

- Binaires :
 - Admtool.exe
 - Wobjing.exe
 - internalizeXml.exe
 - externalizeBase.exe
- Documentation :
 - Csharp Developer
 - Cxx Developer
 - Java Developer
 - UML Modeler
- UML Modeler MDACs :
 - CORBADesigner_V3.0.11.mdac
 - CsDeveloper_V2.0.29.mdac
 - CxxDeveloper_V1.1.23.mdac
 - Documentation_V5.0.09.mdac
 - DOORS_V3.0.15.mdac
 - FortranDeveloper_V1.2.13.mdac
 - JavaDeveloper_V4.1.40.mdac
 - Macros_V2.0.01.mdac
 - Metrics_V3.0.02.mdac
 - RequirementsAdmin_V3.0.15.mdac
 - RequirementsManager_V3.0.15.mdac
 - SQLDesigner_V2.0.12.mdac

- Ressources :
 - ObjecteeringPlugin-1.0.0.zip

2.3. Procédure d'installation Linux

2.3.1. Procédure de déploiement du patch

- Fermer tous les programmes Objecteering.
- Lancer la commande 'rpm -Uvh objecteering-6.0-08170.i586.rpm'

2.3.2. Fichiers déployés

Les fichiers suivants sont déployés par ce patch :

- Binaires :
 - admtool
 - wobjing
 - internalizeXml
 - externalizeBase
- Documentation :
 - Csharp Developer
 - Cxx Developer
 - Java Developer
 - UML Modeler
- UML Modeler MDACs :
 - CORBADesigner_V3.0.11.mdac
 - CsDeveloper_V2.0.29.mdac
 - CxxDeveloper_V1.1.23.mdac
 - Documentation_V5.0.09.mdac
 - DOORS_V3.0.15.mdac
 - FortranDeveloper_V1.2.13.mdac
 - JavaDeveloper_V4.1.40.mdac
 - Macros_V2.0.01.mdac
 - Metrics_V3.0.02.mdac
 - RequirementsAdmin_V3.0.15.mdac
 - RequirementsManager_V3.0.15.mdac
 - SQLDesigner_V2.0.12.mdac

3. Evolutions et faits techniques corrigés

3.1. Objecteering 6.0 Patch P08170

3.1.1. UML Modeler

- Corrections dans la migration des diagrammes de séquences.
- Il est maintenant possible de créer des Classes sous les Interfaces.
- La Recherche avancée dans les diagrammes est opérationnelle.
- Les projets volumineux consommant beaucoup de mémoire virtuelle peuvent maintenant outrepasser la limitation Windows de 2GB en modifiant le fichier boot.ini. (Pour plus d'informations sur la configuration du fichier boot.ini voir le site de Microsoft : <http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.mspx>).

3.1.2. C++ Developer (CxxDeveloper)

- Corrections dans la génération des Parts/Ports.

3.1.3. Fortran Developer

- Amélioration générale de la génération de code.

3.1.4. Java Developer

- Amélioration générale du reverse.
- La tagged value {}JavaNoCode est maintenant prise en compte lors de la migration des liens de généralisation et des interface réalisations.
- Les types JavaLong, JavaByte et JavaShort sont maintenant inclus dans la signature des paramètres.
- La génération de classes génériques contenant des classes imbriquées est opérationnelle.
- La tagged value {}JavaNoCode est maintenant prise en compte sur les éléments contenus dans une interface.
- Amélioration de la génération des imports.

3.2. Objecteering 6.0 Patch P08040

3.2.1. UML Modeler

- Amélioration de la gestion des erreurs réseau.
- Les accents sont correctement migrés depuis la version 5.3.0.
- Il est maintenant possible de créer des diagrammes statiques et UseCase sous les Interfaces.

3.2.2. C# Developer (CsDeveloper)

- **Les nouveautés C# 2.0 (Generics, Iterators, Property accessor visibility, Nullable types, Partial classes) sont supportés.**
- **Le support de la fonctionnalité RAMC a été ajouté, elle permet d'inclure les fichiers sources et les bibliothèques générés depuis le modèle dans un composant de modèle.**
- **Un RAMC FrameworkClassLibrary.ramc contenant la bibliothèque de classes du Framework .NET 2.0 est fourni (mdaplugs\CsDeveloper\2.0.29\ramc), il permet de déployer les namespaces System et Microsoft.Csharp sous forme de composants autonomes dans le modèle.**
- Ajout d'un paramètre "SDK Version 2.0 and upper" permettant de choisir la version de C# utilisée (1.1 ou 2.0).
- Ajout d'une annotation "objingid" pour les classes, interfaces, énumérations, attributs, associations et opérations, permettant à l'outil de retrouver l'élément de modèle dans le projet Objecteering quand il est reversé en mode Round Trip.

- Une référence à "System" est ajoutée systématiquement au projet Visual Studio généré par Objecteering.
- Les énumérations disposent d'un onglet C#.
- Le code généré depuis les notes CsCode est indenté.
- Les héritages multiples entre interfaces sont correctement générés.
- Les projets Visual sont correctement générés.

3.2.3. C++ Developer (CxxDeveloper)

- **Le support de la fonctionnalité RAMC a été ajouté, elle permet d'inclure les fichiers sources générés depuis le modèle dans un composant de modèle.**
- Amélioration de la génération des Makefiles.
- Les types personnels peuvent maintenant être ajoutés par une simple modification des ACTs.
- Le code du constructeur Base est pris en compte par la mise à jour de code.
- Ajout d'une tagged value {}Cxx.Union permettant de générer une union au lieu d'une classe.
- Les attributs types 'Std::multimap' génèrent un include '#include <map>'.
- Le champs Code externe des packages fonctionne correctement.
- Corrections dans la gestion des classes externes.
- Corrections dans la génération des classes imbriqués.
- Corrections dans l'IHM de la boîte d'édition C++ des opérations.
- Ajout d'un warning dans le cas où la génération intelligente ne génère pas de fichiers.

3.2.4. Documentation

- Les sous-classes sont maintenant prises en compte lors de la génération au format RTF.
- Corrections dans les générations PostScript et RTF sous Linux.

3.2.5. Java Developer

- **Le support de la fonctionnalité RAMC a été ajouté, elle permet d'inclure les fichiers sources générés depuis le modèle dans un composant de modèle.**
- **Un RAMC JavaDeveloper.ramc contenant le profil de Documentation, les templates et les méthodes J redéfinissables est fourni avec le MDAC (mdaplugs\JavaDeveloper\4.1.40\mdamodel), et permet d'étendre le MDAC JavaDeveloper.**
- Objecteering génère maintenant des fichiers ANT au lieu de fichiers makefiles. Des boutons 'Generate java code', 'Generate Ant file', 'Build Ant file', 'Create a Jar file' et 'Execute Ant target' ont été ajoutés à l'onglet Java des JarFiles.
- Ajout d'une annotation "objingid" sur les classes, interfaces, énumérations, attributs, associations et opérations, permettant à l'outil de retrouver l'élément de modèle dans le projet Objecteering quand il est reversé en mode Round Trip.
- Amélioration générale du reverse.
- Optimisation des temps de génération.
- Forte et Visual Age ont été retirés de l'option IDE.
- Les interfaces sont prises en compte par la mise à jour de code depuis un package.
- La tagged value {}JavaNoCode est maintenant prise en compte sur les liens de généralisation et les interface réalisations.
- La collection 'ConcurrentLinkedQueue as Queue' est correctement générée.
- Les EJB ne sont plus supportés.

3.2.6. Requirements (RequirementsManager)

- Les liens de traçabilité créés automatiquement pointent de l'élément de modèle UML vers l'élément Requirement.

3.2.7. SQL Designer

- Les valeurs par défaut définies dans les options du MDAC SQL Designer sont prises en comptes lors de la création d'un produit de génération DDL.