

IBM Cúram Social Program Management



Guide de conformité du développement Cúram

Version 6.0.5

IBM Cúram Social Program Management



Guide de conformité du développement Cúram

Version 6.0.5

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 29.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette édition s'applique à IBM Cúram Social Program Management v6.0 5 et à toutes les révisions suivantes, sauf indication contraire dans de nouvelles éditions.

Eléments sous licence - Propriété d'IBM.

© Copyright IBM Corporation 2012, 2013.

© Cúram Software Limited. 2011. Tous droits réservés.

Table des matières

Figures	v
Tableaux	vii
Avis aux lecteurs canadiens.	ix
Chapitre 1. Introduction	1
1.1 Objectif	1
1.2 Public concerné	1
1.3 Conditions préalables	1
Chapitre 2. Développement conforme à Cúram	3
2.1 Présentation	3
2.2 Démarrage d'un nouveau projet.	3
2.2.1 En savoir plus sur la structure du répertoire de développement	3
2.2.2 Contrôle par le code source	3
2.3 Changement des artefacts source du serveur	4
2.3.1 Code source pour les nouvelles méthodes et classes	4
2.3.2 Changement des jeux de règles Cúram Express Rules (CER).	4
2.3.3 Extension des tables de codes	4
2.4 Code source et API	5
2.4.1 API internes.	5
2.4.2 API externes.	6
2.4.3 Mécanismes d'extension.	6
2.4.4 Récapitulatif.	8
2.5 Mise en garde contre les pièges de conformité communs	9
2.5.1 Utilisation de préfixes spécifiques au projet dans les noms d'artefacts	9
2.5.2 Utilisation d'identificateurs numériques dans les données de démonstration et initiales personnalisées	10
2.5.3 Changements internes interdits sur les fichiers d'application	10

2.5.4 Création de dépendances interdite sur les artefacts exemples ou de démonstration	11
2.5.5 Répercussion des changements apportés aux types d'artefacts dynamiques dans le système de développement	11
2.5.6 Création de dépendances interdite sur les API internes	11

Annexe A. Détails sur la conformité des composants	13
A.1 Introduction.	13

Annexe B. Mécanismes d'extension déconseillés	15
B.1 Introduction	15
B.2 Classes d'extension	15
B.2.1 Entity	15
B.2.2 Struct	16
B.2.3 Process, Facade, WebService, WSInbound	17
B.3 Sous-classe avec remplacement	18
B.3.1 Entity	18
B.3.2 Process, Facade, WebService, WSInbound	20
B.4 Sous-classe sans remplacement	22
B.4.1 Entity	22
B.4.2 Process, Facade, WebService, WSInbound	23
B.5 Substitution de domaine	25
B.5.1 Définitions de domaines	25
B.6 Relations	27
B.6.1 Affectable	27
B.6.2 Agrégation.	27
B.6.3 Clé externe.	27
B.6.4 Index	27
B.6.5 Index à entrées uniques	27
B.7 Autres mécanismes	28
B.7.1 Exclusions	28

Remarques	29
Marques	31

Figures

Tableaux

1. Détails sur la conformité des composants	13	7. Sous-classe sans remplacement appliquée aux classes Entity	22
2. Classes d'extension appliquées aux classes Entity	15	8. Sous-classe sans remplacement appliquée à d'autres classes modélisées	23
3. Classes d'extension appliquées aux classes Struct	16	9. Substitution des définitions de domaines	25
4. Classes d'extension appliquées à d'autres classes modélisées	17	10. Relations affectables.	27
5. Sous-classe avec remplacement appliquée aux classes Entity	18	11. Agrégations	27
6. Sous-classe avec remplacement appliquée à d'autres classes modélisées	20	12. Clés externes	27
		13. Index	27
		14. Index à entrées uniques	27
		15. Exclusions	28

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Chapitre 1. Introduction

1.1 Objectif

Ce document fournit des instructions adaptées sur la création de fonctionnalités personnalisées.

Notez que depuis la version 6.0.3, certaines de ces instructions ont changé. Même si tous les mécanismes de personnalisation d'applications sont toujours pris en charge pour les clients qui les ont déjà utilisés, certains sont désormais déconseillés pour les nouveaux développements.

1.2 Public concerné

Ce document est destiné aux concepteurs et aux développeurs des équipes de projets qui créent les applications Cúram.

1.3 Conditions préalables

Une connaissance pratique de l'environnement de développement d'applications est nécessaire pour comprendre ce document. Ce guide fournit des références sur la documentation de développement.

Chapitre 2. Développement conforme à Cúram

2.1 Présentation

Ce chapitre décrit les considérations importantes qui doivent être prises en considération pour un développement conforme à Cúram. Ces considérations sont essentielles pour que le support puisse aider les clients à appliquer leurs propres personnalisations à l'application. En suivant ces considérations, les clients trouveront également plus facile la mise à niveau vers des versions ultérieures.

2.2 Démarrage d'un nouveau projet

Au démarrage d'un nouveau projet, il est important de comprendre la structure du répertoire de développement. Il est également important de le placer sous le contrôle du code source.

2.2.1 En savoir plus sur la structure du répertoire de développement

Une bonne connaissance de la structure du répertoire de développement est requise pour comprendre où se trouvent les artefacts de développement, comment ils sont organisés et où doivent être enregistrés les changements apportés à ces artefacts. Notez que pour accéder à la structure du répertoire de développement, vous devez commencer par installer une version de développement de l'application.

La liste suivante décrit les répertoires dans lesquels sont installés les artefacts de développement client et serveur :

- Les artefacts de développement client sont installés dans le répertoire `webclient`. Pour plus d'informations sur le mode de développement des applications client, voir le guide *Cúram Web Client Reference Manual*.
- Les artefacts de développement serveur sont installés dans le répertoire `EJBServer`.

Dans les répertoires `webclient` et `EJBServer`, il existe un sous-répertoire `components` possédant un sous-répertoire supplémentaire appelé `custom`. Le sous-répertoire `custom` correspond à l'emplacement de tous les artefacts de développement spécifiques au projet. Les autres sous-répertoires `components` contiennent tous les artefacts de développement d'application fournis avec le produit.

Important : Le dossier `custom` contient une structure de démarrage pour la première utilisation. Dans la documentation du développeur, il est signalé comme la zone dans laquelle doivent être développés tous les artefacts. Notez que cela n'est pas imposé. Pour chaque projet, vous pouvez choisir d'effectuer le développement dans ce composant ou de créer un nouveau composant adapté à votre projet.

Le répertoire `EJBServer\components\custom\model` comporte un fichier modèle de démarrage et des fragments de modèles.

2.2.2 Contrôle par le code source

Pour conserver une trace de tous les changements apportés aux artefacts source, la structure du répertoire de développement doit être placée sous le contrôle du code source. Une fois sous le contrôle du code source, tous les artefacts de développement doivent être balisés. Assurez-vous que la balise fait référence à la version de l'application. A tout moment, il est possible de générer un rapport, à l'aide de différentes fonctionnalités, de tous les fichiers ajoutés ou modifiés pour implémenter les fonctionnalités du projet. Ce rapport est utile lors de l'utilisation d'une nouvelle édition de l'application.

Notez que depuis la version 6.0.3, des changements ont été apportés au mode de fourniture du code source Java™. Voir la section 4 ci-dessous pour plus d'informations.

2.3 Changement des artefacts source du serveur

Il existe plusieurs types d'artefacts serveur, dont certains sont des classes d'application. Certains d'entre eux sont représentés dans un modèle d'application. D'autres interfaces Java sont "créées à la main". Il existe des implémentations d'application de ces deux catégories de classe et il est important de pouvoir faire la distinction entre les deux. Même s'il est possible de changer les aspects d'une interface modélisée en modifiant le modèle et en régénérant le code, il n'est pas possible de changer une interface créée à la main.

Interfaces modélisées :

- Apparaissent dans le modèle UML de l'application

Interfaces créées à la main :

- N'apparaissent pas dans le modèle UML de l'application
- Apparaissent dans les répertoires de composant de votre environnement de développement
- Ne peuvent pas être personnalisées
- Contiennent l'annotation Google Guice `@ImplementedBy` pour indiquer la classe d'implémentation de l'application

Certains composants peuvent contenir des interfaces qui n'appartiennent à aucune des catégories ci-dessus, mais ceux-ci seront toujours décrits dans la documentation qui leur est spécifique. Les interfaces d'application modélisées et créées à la main peuvent comporter des implémentations qui peuvent être personnalisées.

Pour plus de détails sur l'implémentation d'artefacts source, voir le document *Cúram Server - Guide de développement* pour les implémentations d'interfaces modélisées et le document *Guide de Persistence* pour les implémentations d'interfaces créées à la main. Il est nécessaire d'examiner l'interface implémentée pour déterminer la catégorie. Les recommandations sur le changement des artefacts source du serveur ont changé avec la version 6.0.3. Notez que les recommandations contenues dans ce document (Guide de conformité du développement Cúram) sont définitives.

2.3.1 Code source pour les nouvelles méthodes et classes

Les nouvelles classes spécifiques à un client, les classes qui encapsulent les classes existantes ou, dans certains cas, les nouvelles sous-classes de classes existantes doivent être écrites dans de nouveaux fichiers source. Tous les nouveaux fichiers source doivent être placés dans le sous-répertoire source du répertoire `EJBServer\components\custom`.

Pour les classes modélisées, la hiérarchie de classes générée indique la structure de package des nouveaux fichiers source. Pour plus d'informations sur la modélisation des nouvelles classes et le remplacement des implémentations existantes, voir le guide *Cúram Server Developer's Guide*.

Pour les implémentations à la main, vous pouvez choisir le package de la nouvelle classe. Pour plus d'informations sur la configuration de nouvelles sous-classes à l'aide de Google Guice, voir le guide *Persistence Cookbook*.

2.3.2 Changement des jeux de règles Cúram Express Rules (CER)

CER Editor stocke ses jeux de règles dans la base de données plutôt que sur le système de fichiers. Tous les jeux de règles présents dans le composant de base NE doivent PAS être personnalisés. Des solutions peuvent disposer de leurs propres déclarations de conformité concernant leurs jeux de règles.

2.3.3 Extension des tables de codes

Notez que la documentation est maintenant fournie pour indiquer les tables de codes dont l'extension est sûre et celles qui nécessitent que les clients sollicitent le support avant toute personnalisation. Une liste

des tables de codes qui ne peuvent pas être étendues sans contacter le support est fournie dans la structure de répertoire de la documentation du projet pour chaque installation (dans un dossier appelé `RestrictedCodeTables`). Si vous souhaitez personnaliser une table de codes répertoriée dans cette liste, vous devez créer un dossier de support.

2.4 Code source et API

Toutes les fonctionnalités Java de l'application sont désormais distribuées sous forme de fichiers JAR préconfigurés. Cela a toujours été le cas pour les modules Enterprise introduits depuis la version 5.0 (pour lesquels le code source n'a jamais été fourni), mais désormais cela est universel. Les fonctionnalités de l'application sont désormais uniquement régénérées et reconstruites dans une installation client, si cela est requis par l'utilisation des mécanismes d'extension client. Cela signifie que le processus de construction client n'a plus besoin de régénérer l'intégralité du code source Java ; seuls le code source spécifique au projet et le code source Java dépendant régénéré ont désormais besoin d'être régénérés.

Pour un nombre limité de zones fonctionnelles clés de la version 6.0.3 et des versions ultérieures, le code source Java n'est plus distribué sous aucune forme. Le code source du reste de l'application continue d'être fourni (en tant qu'exemple), mais à des fins de documentation uniquement. Ce code n'est pas directement impliqué dans le processus de construction depuis la version 6.0.3. Cet exemple de code source est distribué par composant dans des fichiers JAR, comme suit : `EJBServer\components\\sample\src.zip`. Les versions de construction de chaque composant sont disponibles à l'emplacement suivant : `EJBServer\components\\lib\.jar`.

De plus, depuis la version 6.0.3, les opérations de classe sont marquées comme internes ou externes via des annotations.

Les opérations externes forment l'API officielle de l'application, que les clients sont encouragés à utiliser et à appeler depuis leur propre code.

Important : Les classes sans annotation sont internes par défaut.

2.4.1 API internes

Même s'il est possible d'appeler et d'associer à une sous-classe des API internes à partir du code personnalisé, cela est déconseillé dans la version 6.0.3. Ces API sont annotées avec `@AccessLevel (INTERNAL)`.

Important : Dans ce contexte, 'déconseillé' signifie que leur utilisation continue d'être prise en charge, mais que ces API peuvent être changées ou supprimées dans les éditions ultérieures, lorsqu'un délai de préavis minimal d'un an a été accordé aux clients à l'égard de tout changement ou suppression de ce type.

Remarque : Aucun préavis de ce type n'est accordé pour l'une des API marquées comme internes dans la version 6.0.3 (c'est-à-dire qu'il n'existe pas de plans en cours permettant de changer l'une des API marquées comme internes dans la version 6.0.3). Par conséquent, les clients doivent pouvoir bénéficier d'un délai approprié pour planifier ce type de migrations.

Les références client existantes aux API marquées comme internes à partir de la version 6.0.3 continueront à fonctionner comme auparavant, à cela près que les avertissements dissuasifs seront générés dans les projets Eclipse qui disposent de ces dépendances.

Les projets doivent s'efforcer de s'écarter de ces dépendances concernant les API internes au fil du temps. De plus, ils ne doivent pas introduire de nouvelles dépendances associées (dans la mesure du raisonnable et en fonction du stade où se situe un projet client dans son processus de conception/développement, cela peut être inévitable à court terme). La plupart des clients existants verront des références déconseillées signalées après le passage à la version 6.0.3 ou versions ultérieures. De plus, il n'est pas

prévu que les clients les corrigent immédiatement dans le cadre de ce passage. Comme mentionné plus haut, cela n'aura pas d'incidence sur leurs droits de support.

Notez que comme avec les versions précédentes de l'application, certaines API internes ont été configurées pour générer des erreurs "d'accès restreint" dans Eclipse en cas de référencement (ces API sont annotées avec `@AccessLevel(RESTRICTED)`) et ces références ne seront pas prises en charge dans les projets client. Ces API ont toujours été internes et n'ont jamais été prises en charge pour l'utilisation client ; il sera alors facile de les identifier (les API d'accès restreint génèrent des erreurs Eclipse et les API déconseillées génèrent des avertissements Eclipse).

2.4.2 API externes

Les API externes peuvent être référencées directement par des projets client. Ces API sont annotées avec `@AccessLevel(EXTERNAL)`. Un Javadoc est fourni pour toutes les API externes, et ce par composant. Celui-ci se trouve dans `EJBServer\components\\doc\api.zip`. Notez que certains composants peuvent ne pas avoir de Javadoc, puisqu'ils ne disposent d'aucune API externe. Seules les classes documentées dans le Javadoc doivent être référencées à partir du code client ; le référencement d'autres classes générera des avertissements dissuasifs ou des erreurs d'accès restreint. De plus ces classes ne seraient pas prises en charge.

Notez également que, comme avec toutes les API, il est prévu que celles marquées comme externes évolueront au fil du temps (tout en restant compatibles en amont). Si vous disposez d'une exigence qui vous semble ne pas pouvoir être satisfaite via une combinaison de l'utilisation d'API externes et de mécanismes d'extension autorisés, vous devez consulter le support à ce sujet. Le cas échéant, une nouvelle approche basée sur une API, un point d'ancrage de personnalisation, un modèle de stratégie ou encore une configuration sera disponible. De plus, ces nouvelles API peuvent être fournies dans des modules de fonctions. Sinon, une API interne existante peut, dans certains cas, être transformée en API externe (le cas échéant).

2.4.3 Mécanismes d'extension

La suppression du code source des domaines de fonctionnalités clés susmentionnés a entraîné un changement de la méthode recommandée pour l'utilisation des mécanismes d'extension sur des projets client. Auparavant, si les clients souhaitaient utiliser les divers mécanismes d'extension de l'application (par exemple, classes d'extension, sous-classe avec ou sans remplacement, agrégation), ils pouvaient effectuer une recherche sur l'ensemble du codebase afin de voir où et comment les classes cible ont été appelées dans le code d'application. Ils pouvaient alors procéder à une évaluation des effets fonctionnels de l'extension prise en compte.

Avec la version 6.0.3, les clients ne disposent plus du code source pour certains domaines de fonctionnalités clés. De plus, de nombreuses API ont été marquées comme internes. La section suivante récapitule les changements concernant les pratiques d'extensions recommandées pour les projets client.

Notez que cette section fait uniquement référence à des restrictions lors de l'extension d'artefacts d'application. Tous les mécanismes d'extension peuvent continuer à être utilisés sur les classes définies par le client. De plus, tous les artefacts de ce type peuvent évidemment être de nature externe et appelés à partir de toute autre partie d'une implémentation client.

Important : Cette section fournit uniquement un récapitulatif de haut niveau. Des détails complets concernant quels mécanismes sont autorisés sur quels types de classe de la version 6.0.3 sont fournis dans l'Annexe B, «Mécanismes d'extension déconseillés», à la page 15. Lorsque les mécanismes ont été déconseillés, cette annexe recommandera, le cas échéant, des mécanismes alternatifs utilisables par les clients.

2.4.3.1 Classes Entity

A quelques exceptions près, l'utilisation et la modification client directes des classes Entity de l'application sont maintenant déconseillées. Dans de nombreux cas, les opérations de la classe Entity de

l'application disposent d'équivalents directs de couche de façade qui ont été marqués comme externes ; ils peuvent être utilisés par les clients. L'ajout d'opérations stéréotypées et non stéréotypées à des entités de l'application est toutefois toujours autorisé, tout comme la définition d'un grand nombre d'options Entity.

Avant la version 6.0.3, les attributs pouvaient être ajoutés à des classes Entity de l'application utilisant une extension. Toutefois, étant donné que le code source est supprimé pour les domaines de fonctionnalités clés, les clients n'auront plus de visibilité quant à savoir si les attributs ajoutés via les classes d'extension seront mappés aux API externes. Pour cette raison, l'ajout d'attributs à des classes Entity de l'application est désormais déconseillé.

Les clients qui souhaitent ajouter des données à des écrans d'application doivent ajouter de nouvelles classes Entity spécifiques au client et doivent encapsuler des opérations de maintenance de l'application externe dans leurs propres classes de processus afin de gérer les deux tables atomiquement. Les écrans d'application peuvent ensuite être changés afin de pointer vers les nouvelles classes de processus.

Remarque : Les entités représentant les types de preuves constituent une exception à cette règle. Les clients peuvent continuer à ajouter des attributs à ce type d'entité de preuve de l'application utilisant une extension, comme cela est requis par Evidence Generator.

Dans la version 6.0.3, les entités de preuve de l'application ont été incorrectement marquées comme internes ; cela sera corrigé dans une révision suivante. Pour l'instant, les clients utilisant une extension sur des entités de preuve pour ajouter des attributs peuvent lire des avertissements dissuasifs dans Eclipse concernant ces classes ; ces avertissements spécifiques peuvent être ignorés.

Cette remarque s'applique uniquement aux entités représentant des types de preuves, et non à toute autre classe Entity de l'application.

2.4.3.2 Définitions de domaines

En général, l'utilisation et la substitution client des définitions de domaines de l'application sont toujours autorisées. Toutefois, le changement du type fondamental d'une définition de domaine est à présent déconseillé, car il change également de nombreuses options liées aux tables de codes.

2.4.3.3 Classes Struct

Les classes Struct de l'application sont toutes externes de nature, car elles peuvent être référencées dans des fonctionnalités spécifiques au client.

Il est déconseillé aux clients de créer directement des agrégations depuis des structures d'application sur toute autre structure (car elles n'ont plus de visibilité complète sur l'emplacement d'utilisation de ces structures d'application). Les clients peuvent toutefois continuer à utiliser des agrégations pour inclure des structures d'application à leurs propres structures de projets.

2.4.3.4 Autres classes modélisées

Pour les autres classes modélisées de l'application (comme Process, Facade, WSInbound et WebService), l'utilisation de tous les mécanismes d'extension est désormais déconseillée.

Avant la version 6.0.3, la sous-classe avec remplacement était un mécanisme fréquemment utilisé pour ajouter et changer des opérations sur les classes Process et Facade de l'application. Avec l'extension des classes Entity de l'application, il est désormais potentiellement dangereux. Les clients ne bénéficieront donc plus nécessairement d'une visibilité totale quant aux emplacements d'utilisation de ces classes.

Comme avec les classes Entity, les clients doivent plutôt modéliser et coder leurs propres classes Process, Facade ou WSInbound, en encapsulant les API externes existantes ou en implémentant de nouvelles fonctionnalités. Pour les opérations Facade, les pages concernées du gestionnaire d'interface utilisateur peuvent être repointées vers les nouvelles opérations Facade si nécessaire.

2.4.3.5 Classes non modélisées

Certains composants contiennent des classes non modélisées. Pour ces classes, l'utilisation de chaque interface ou classe externe est décrite dans le Javadoc de la classe.

Certaines classes non modélisées sont associées à des restrictions d'accès Eclipse en place pour fournir aux clients des conseils relatifs aux API qu'ils peuvent et ne peuvent pas appeler ou personnaliser. Certains packages et classes sont marqués comme restreints ; ces classes ne doivent pas être utilisées puisqu'il s'agit de classes internes pouvant changer au fil du temps. Les restrictions d'accès ne doivent pas être supprimées du fichier Eclipse.classpath car cela peut provoquer la consommation de classes restreintes pouvant entraîner des problèmes lors des mises à niveau.

Certains composants non modélisés contiennent des classes protégées de package ; ces classes ne doivent pas être utilisées en mode personnalisé. Les clients ne doivent pas placer de code personnalisé dans la même structure de package pour appeler ou référencer des classes protégées de package.

De nombreuses API non modélisées ne sont pas directement personnalisables. Seules les interfaces/classes marquées par l'annotation `@Implementable` peuvent être étendues ou implémentées. Pour ces classes, le Javadoc détaille leur personnalisation ou leur implémentation. Les classes non modélisées qui ne sont pas marquées avec l'annotation `@Implementable` ne doivent pas être étendues ou implémentées, car de nouvelles opérations peuvent être ajoutées au fil du temps et entraîner des répercussions sur la mise à niveau.

Pour les classes marquées avec l'annotation `@Implementable`, les mécanismes de personnalisation typiques pour ces types de classes sont les événements et les stratégies.

Les événements permettent aux clients d'ajouter une logique personnalisée à divers points de l'application. Pour plus de détails sur l'ajout de programmes d'écoute d'événements, veuillez consulter le document Guide de Persistence. Les classes d'événement s'appellent généralement 'xxxEvent' afin d'être facilement identifiées.

Les modèles de stratégies permettent aux clients de changer le comportement par défaut de certaines fonctions dans l'application. Chaque classe de stratégie dispose d'une implémentation par défaut fournie. Cependant, les clients peuvent choisir de substituer l'implémentation par défaut de l'une des opérations de stratégie grâce à l'utilisation de liaisons Guice. Pour plus de détails sur l'utilisation des liaisons Guice, veuillez consulter le document Guide de Persistence. Les classes de stratégie s'appellent généralement 'xxxStrategy', afin d'être facilement identifiées.

Remarque : Pour plus de détails sur la conformité par composant, veuillez consulter l'Annexe A, «Détails sur la conformité des composants», à la page 13.

2.4.4 Récapitulatif

En résumé :

Lorsque vous souhaitez référencer une classe d'application dans votre code personnalisé :

- Si la classe est externe, vous êtes autorisé à la référencer.
- Si la classe est interne, vous pouvez la référencer dans votre code existant mais cela est déconseillé. Vous ne devez pas la référencer dans un nouveau code.
- Si la classe est à accès restreint, vous ne pouvez pas la référencer.

Lorsque vous souhaitez personnaliser une classe d'application :

- Si la classe est modélisée, reportez-vous à l'Annexe B pour plus de détails sur les personnalisations autorisées.
- Si la classe n'est pas modélisée, reportez-vous au Javadoc et/ou aux guides de configuration/développement de son composant parent pour plus de détails sur les personnalisations.

2.5 Mise en garde contre les pièges de conformité communs

Cette section décrit les problèmes de conformité pouvant survenir et fournit des règles empiriques permettant de les éviter. Le respect des règles empiriques présentées dans ces sections à partir des premières étapes d'un projet est relativement simple. Toutefois, si elles ne sont pas respectées, elles peuvent entraîner ultérieurement de graves perturbations ; les corrections peuvent être à la fois coûteuses et complexes.

2.5.1 Utilisation de préfixes spécifiques au projet dans les noms d'artefacts

Vous devez ajouter un préfixe à tous les noms d'artefacts source (classes de modèles, fichiers source, messages, fichiers messages, etc.) en utilisant un acronyme ou une abréviation. Utilisez toujours le même acronyme ou la même abréviation. Au cours du projet, les ajouts aux artefacts principaux seront ainsi plus évidents. Cette distinction est très utile lorsque l'effort de développement grandit. En général, la plupart des projets sont décrits par des acronymes. Les acronymes sont parfaits comme préfixes.

L'utilisation d'un préfixe spécifique à un projet permet d'éviter tout conflit de noms entre les nouveaux artefacts que vous ajoutez et les nouveaux artefacts que Cúram ajoute ; les conflits de noms peuvent être coûteux et difficiles à corriger.

Par exemple, vous utilisez un module de mise à jour et vous découvrez que l'un de vos ajouts de zones de base de données personnalisées porte le même nom qu'une nouvelle zone d'application n'ayant pas les mêmes significations commerciales ou types de données. Sinon, vous utilisez un module de mise à jour et vous découvrez qu'une nouvelle application Élément de table de codes a été ajoutée et qu'elle est en conflit avec un élément personnalisé que vous avez ajouté et nommé de la même manière, mais avec une autre signification. Ces types de conflits peuvent être évités en vous assurant toujours de nommer les nouveaux artefacts personnalisés avec un préfixe cohérent.

Certains types d'artefacts possèdent plusieurs identificateurs, et cela doit être pris en compte lors du nommage. Les classes Entity et les éléments de tables de codes sont dans ce cas. Une classe Entity personnalisée possède un nom de table, qui partage le même espace de nom non hiérarchique (schéma de base de données) que les tables d'application, et doit comporter un nom de table unique dans cet espace de nom. Elle possède également un nom de classe Java, qui partage un espace de nom hiérarchique (structure de package) avec les classes Java de l'application. De même, un élément de table de codes personnalisé possède une valeur et un identificateur Java, qui partagent un espace de nom non hiérarchique avec les éléments d'application de la même table de codes.

Il est important de noter que l'utilisation de préfixes spécifiques au projet ne s'applique pas lorsque vous remplacez un artefact d'application, car le mécanisme de remplacement est généralement basé sur le fait que le nom de votre artefact personnalisé est exactement le même que celui de l'artefact d'application qu'il remplace.

Remarques supplémentaires :

- Les identificateurs sont multiples : nom de fichier, ID XML, nom de classe Java ou combinaison de tous ces éléments.
- Un préfixe court est conseillé car la longueur des noms est souvent restreinte (par exemple, pour certains types d'identificateurs de base de données).

Remarque : En plus des artefacts source, il est également important de prendre en compte les valeurs d'identificateurs pouvant entrer en conflit avec les valeurs utilisées par IBM.

Les API `TransactionInfo.setFacadeScopeObject` et `TransactionInfo.getFacadeScopeObject` permettent aux développeurs d'accéder aux objets associés à la transaction en cours. Lors de l'utilisation de cette API,

pour garantir que les données de la transaction n'entrent pas en conflit avec les données appartenant à IBM, vous devez utiliser String comme identificateur d'objet et choisir un mot approprié comme préfixe pour cette chaîne, comme décrit ci-dessus.

2.5.2 Utilisation d'identificateurs numériques dans les données de démonstration et initiales personnalisées

Les données de démonstration et initiales prédéfinies sont chargées dans une base de données d'application via des fichiers DMX. Ces données sont installées dans la base de données lors de la première configuration du système ou lors de la mise à niveau d'un système. Un ensemble de données de démonstration et initiales est fourni dans l'application. Les clients peuvent également ajouter leurs propres données de démonstration et/ou initiales.

Pour éviter tout conflit avec les données de démonstration et initiales fournies dans l'application et avec les données créées par le système d'exécution, il est important de créer les identificateurs (par exemple, les clés primaires) des données de démonstration et initiales du client dans des plages réservées. Par conséquent, un ensemble de plages est réservé pour le client.

2.5.2.1 Plages réservées

Les projets doivent utiliser des identificateurs (clés primaires) dans leurs données initiales et de démonstration personnalisées extraites des plages réservées suivantes :

- Clés primaires non lisibles : 45 000 à 49 999 (inclus)
- Clés primaires lisibles : 11 521 à 12 799 (inclus)
- Jeux de règles : 4 500 à 4 999 (inclus)

Les clients ayant déjà utilisé des identificateurs en dehors de ces plages seront aidés pour résoudre le problème avant l'exécution de leur prochaine mise à niveau.

2.5.2.2 Jeux de données volumineux

De temps à autre, il peut être nécessaire de générer des jeux de données très volumineux. Par exemple, cela peut être requis pour les tests de chargement. Dans ce type de cas, le nombre d'enregistrements requis dépasserait de loin les plages de clés attribuées et documentées ici. Dans ce cas, une approche différente doit être adoptée.

Au lieu d'utiliser les clés des plages attribuées, le serveur de clés doit être utilisé pour générer les valeurs de clé requises. Si ces données sont importées dans une base de données reconstruite, la valeur finale du jeu de clés doit également être extraite et chargée dans la table des jeux de clés, et ainsi remplacer la valeur de jeu de clés initiale fournie dans l'application. En cas de questions sur ce processus, veuillez contacter le support pour plus d'informations.

2.5.2.3 Exception de tables de codes

Veuillez noter que la déclaration ci-dessus ne s'applique pas aux tables de codes.

2.5.3 Changements internes interdits sur les fichiers d'application

Les service packs et les correctifs d'urgence doivent pouvoir déplacer, restructurer ou écraser des fichiers d'application en toute sécurité. Si vous modifiez ces fichiers, les service packs ou les correctifs d'urgence peuvent les écraser sans préavis. Il n'existe aucune garantie que ces changements seront compatibles avec les changements que vous avez apportés ; la réapplication ultérieure des changements internes peut donc ne pas être possible.

Il existe un très petit nombre d'exceptions à cette règle qui sont répertoriées ci-dessous :

- EJBServer
 - /project/config/datamanager_config.xml
 - /project/config/deployment_packaging.xml

- /project/properties/Bootstrap.properties
- .classpath
- .project
- Webclient
 - /JavaSource/curam/omega3/ApplicationConfiguration.properties
 - /JavaSource/curam/omega3/il8n/CDEJResources.properties
 - .classpath
 - .project

2.5.4 Création de dépendances interdite sur les artefacts exemples ou de démonstration

Les artefacts exemples et les artefacts de démonstration dépassent les limites pour les dépendances personnalisées, c'est-à-dire les références aux API, les fichiers du gestionnaire d'interface utilisateur, les tables de codes, les fichiers message, etc., dans ce type de composants du code personnalisé. Ces artefacts peuvent être modifiés sans préavis.

Différents domaines de produit dans Cúram ont adopté diverses approches afin de marquer les artefacts comme internes/exemples/de démonstration. Ce guide ne peut donc pas indiquer de manière concise comment les identifier. Cependant, il existe quelques règles empiriques fiables :

- Les artefacts dont le nom, le module de code, le package de modèle ou le chemin d'accès au fichier contiennent les mots 'Internal (interne)', 'Sample (exemple)' ou 'Demo (de démonstration)' (ou des dérivés évidents de ces mots)

En cas de doute, contactez le support IBM.

Important : Le dossier CPMSample est interne ; l'ensemble du code et des artefacts de ce dossier peuvent changer sans préavis. Si les clients souhaitent utiliser les fonctionnalités dans CPMSample, ils devront les dupliquer dans leur codebase.

2.5.5 Répercussion des changements apportés aux types d'artefacts dynamiques dans le système de développement

Si vous modifiez des types d'artefacts dynamiques dans les systèmes de production ou de test, vous devez toujours vérifier que ces changements sont répercutés dans le système de développement.

Divers artefacts de développement "dynamiques" existent dans l'application et peuvent être modifiés lors de l'exécution sur un système de production ou de test (par exemple, tables de codes, workflows, etc.). Les changements d'exécution apportés à ces artefacts doivent toujours être resynchronisés avec le codebase de développement afin que les changements de développement simultanés puissent être intégrés aux changements d'exécution avant le déploiement.

Les changements simultanés apportés à ces artefacts peuvent se produire au cours du développement de jalon de projet de routine, de l'adoption de service packs ou encore de l'exécution de mises à niveau de versions majeures/mineures. Dans tous les cas, il doit exister un emplacement central dans lequel les changements simultanés sont tous fusionnés et validés ; il s'agit du codebase de développement. Le système de l'enregistrement pour ces artefacts est le codebase de développement.

2.5.6 Création de dépendances interdite sur les API internes

Avec la version 6.0.3, les clients doivent éviter d'appeler ou de personnaliser les classes d'application et les opérations marquées comme "internes", puisque ces API peuvent changer dans les versions ultérieures de l'application.

Annexe A. Détails sur la conformité des composants

A.1 Introduction

Cette section contient des informations individuelles sur la conformité par composant.

Important : Sauf indication contraire, pour tous les composants (qu'ils soient ou non répertoriés ici), il peut être supposé que les déclarations de conformité générales suivantes s'appliquent :

Lorsque vous souhaitez référencer une classe d'application dans votre code personnalisé :

- Si la classe est externe, vous êtes autorisé à la référencer.
- Si la classe est interne, vous pouvez la référencer dans votre code existant mais cela est déconseillé. Vous ne devez pas la référencer dans un nouveau code.
- Si la classe est à accès restreint, vous ne pouvez pas la référencer.

Lorsque vous souhaitez personnaliser une classe d'application dans votre code personnalisé :

- Si la classe est modélisée, consultez l'annexe B pour connaître les actions que vous êtes autorisé à effectuer.
- Si la classe est non modélisée, consultez son Javadoc (EJBServer\components\

Tableau 1. Détails sur la conformité des composants.

Ce tableau répertorie les composants avec API non modélisées.

Composant	Détails
Cúram Client Development Environment	Pour obtenir des instructions supplémentaires sur la manière de personnaliser/d'utiliser ce composant, veuillez consulter le document <i>Cúram Web Client - Manuel de référence</i> . Veuillez noter que les fichiers du dossier CuramCDEJ seront copiés dans des dossiers de construction temporaire lors du processus de construction de l'application. La présence de ces fichiers en dehors du dossier CuramCDEJ ne les rend pas disponibles pour la personnalisation.
Cúram Server Development Environment	<p>Le Javadoc de ce composant détaille l'ensemble des points de personnalisation et des API externes. Seules les classes documentées dans le Javadoc doivent être référencées à partir du code client ; le référencement d'autres classes générera des avertissements dissuasifs ou des erreurs d'accès restreint. De plus ces classes ne seraient pas prises en charge (La fonctionnalité de cryptographie de Cúram n'est pas prise en charge pour une utilisation par le client au-delà de ce qui est indiqué dans le manuel <i>Cúram Server - Guide des développeurs</i> et dans le <i>Manuel de sécurité Cúram</i>).</p> <p>Le répertoire bin de ce composant contient des scripts de génération Apache Ant qui ne doivent pas être modifiés directement. Les mises à jour de ces scripts peuvent être effectuées en créant des scripts Ant personnalisés et en utilisant la fonctionnalité d'héritage Ant.</p> <p>Le dossier des pilotes de ce composant contient les pilotes de base de données utilisés pour accéder à la base de données d'application. Si nécessaire, ces pilotes peuvent être remplacés par le pilote approprié pour la base de données utilisée, à condition que la base de données soit une version de base de données prise en charge comme indiqué dans le document <i>Conditions préalables prises en charge Cúram</i>.</p> <p>Remarque : Si un problème survient avec un pilote qui n'a pas encore été inclus dans le produit (c'est-à-dire qui n'a pas été testé et vérifié pour une utilisation avec l'application), il peut être demandé au client de remplacer le pilote par une version qui a été testée, alors que le problème spécifique est soulevé auprès du fournisseur tiers. Veuillez noter que les fichiers du dossier CuramSDEJ seront copiés dans des dossiers de construction temporaire lors du processus de construction de l'application. La présence de ces fichiers en dehors du dossier CuramSDEJ ne les rend pas disponibles pour la personnalisation.</p>
Cúram Administration Suite	Notez qu'à partir de la version 6.0.3, la déclaration de conformité pour les classes dans Cúram Administration Suite n'est pas différente de celles de tout autre composant. Les API externes dans Administration Suite peuvent être encapsulées et appelées à partir du code personnalisé.
Persistence Infrastructure	Persistence Infrastructure ne peut pas être personnalisé. Les clients ne doivent pas placer de code personnalisé dans des modules de code de PI (curam.util.persistence et tous les sous-modules). Pour plus d'informations sur l'utilisation de ces API, veuillez lire le document <i>Guide de Persistence</i> .

Tableau 1. Détails sur la conformité des composants (suite).

Ce tableau répertorie les composants avec API non modélisées.

Composant	Détails
CER Infrastructure	La déclaration de conformité concernant CER Infrastructure se trouve dans le document <i>Cúram Express Rules - Manuel de référence</i> . Les entités CER (c'est-à-dire toute entité dont le nom est préfixé par le mot Creole) doivent être considérées comme internes et sujettes à modification. De plus, les clients ne doivent pas les mettre à jour ou les interroger, sauf via l'API CER ou les fichiers DMX.
Dependency Manager	<p>Dependency Manager inclut tous les artefacts du serveur dans le module de code curam.dependency et dans tous ses sous-modules.</p> <p>Les composants suivants contribuent au module de code Dependency Manager :</p> <ul style="list-style-type: none"> • CER Infrastructure ; et • l'application de base. <p>Dependency Manager ne peut être personnalisé en aucune manière. Toutes les API Dependency Manager sont destinées à un usage interne uniquement. La déclaration de conformité concernant Dependency Manager se trouve dans le document <i>Cúram Express Rules - Manuel de référence</i>.</p>
Eligibility and Entitlement Engine API	Pour obtenir des instructions sur la configuration et la personnalisation de ce composant, veuillez lire le document <i>Cúram Eligibility and Entitlement à l'aide du guide Cúram Express Rules</i> .
Evidence Generator	Evidence Generator est une infrastructure d'application fournie dans le cadre de la structure de répertoire Tools (EGTools). Pour plus d'informations sur l'utilisation d'Evidence Generator, voir le document <i>Spécification de Cúram Evidence Generator</i> .
DocMaker, Online Help Tools	Aucune partie de l'outil DocMaker ou d'Online Help Tools ne peut être personnalisée, sauf comme indiqué dans le document <i>Cúram Online Help System - Guide de développement</i> .
Pod Infrastructure	Pod Infrastructure est fourni dans les fichiers widget-inf.jar et widget-utility.jar. Pod Infrastructure ne peut pas être personnalisé. Les chargeurs de conteneurs ne peuvent pas non plus être personnalisés. Pour plus d'informations sur le développement de conteneurs, voir le document <i>Cúram Pod - Guide de développement</i> .
Funded Program Management	Pour obtenir des instructions sur la personnalisation de ce composant, veuillez lire le document <i>Funded Program Management - Guide de développement</i> et le Javadoc du composant.
Cúram Incidents	Pour obtenir des instructions sur la personnalisation d'entités d'incident ou sur le remplacement d'une implémentation d'incident, veuillez lire le document <i>Guide de Persistence</i> et le Javadoc du composant.
Cúram Citizen Context Viewer	Pour obtenir des instructions sur la personnalisation de ce composant, consultez le document <i>Cúram Citizen Context Viewer - Guide de configuration</i> et le Javadoc du composant.
Cúram Advisor	Les composants serveur suivants sont fournis avec Cúram Advisor : Advisor.
Cúram Common Intake	Les composants serveur suivants sont fournis avec Cúram Common Intake : Intake, PCR, CREOLEProgramRReccommendation, ReferralsLite et CPMReferralsLite.
Inbox	Pour obtenir des instructions sur la configuration et la personnalisation de ce composant, veuillez lire la partie VI du document <i>Cúram Workflow - Guide de référence</i> .
Cúram Waitlists	Pour obtenir des instructions sur la personnalisation de ce composant, veuillez lire le document <i>Cúram Waitlist - Guide de personnalisation</i> et le Javadoc du composant.
IBM Cúram Business Intelligence and Analytics	Pour obtenir des instructions sur la personnalisation de ce composant, veuillez lire le document <i>Cúram Business Intelligence Reporting - Guide de développement</i>
IBM Cúram Social Enterprise Collaboration	Les composants serveur suivants sont fournis avec Social Enterprise Collaboration : SocialEnterpriseCollaboration, CaseParticipantIndex et ClientAccess.
IBM Cúram Universal Access	Pour obtenir des instructions sur la personnalisation de ce composant, consultez le document <i>Cúram Universal Access - Guide de développement</i> et le Javadoc du composant.
IBM Cúram Outcome Management	Les composants serveur suivants sont fournis avec Cúram Outcome Management : AssessmentPlanning, AssessmentPlanningCPM, DecisionAssistAssessments et SimpleOutcomeManagement.
IBM Cúram Provider Management	Pour obtenir des instructions sur la personnalisation de ce composant, consultez le document <i>Cúram Provider Management - Guide de développement</i> et le Javadoc du composant.
IBM Cúram Youth Services(CYS)	Pour obtenir des instructions sur la personnalisation d'entités CYS ou sur le remplacement d'une implémentation CYS, veuillez lire le document <i>Guide de Persistence</i> et le Javadoc du composant.
IBM Cúram Child Care (CCC)	Pour obtenir des instructions sur la personnalisation d'entités CCC ou sur le remplacement d'une implémentation CCC, veuillez lire le document <i>Guide de Persistence</i> et le Javadoc du composant.

Annexe B. Mécanismes d'extension déconseillés

B.1 Introduction

Comme décrit plus haut, la plupart des mécanismes précédemment recommandés comme moyens d'étendre ou de remplacer des classes d'application dans les versions antérieures à 6.0.3 sont désormais déconseillés. Cette annexe détaille les mécanismes autorisés et déconseillés pour l'application aux types de classes. Elle fournit également des solutions si une combinaison de type mécanisme/classe que vous souhaitez utiliser est désormais déconseillée.

B.2 Classes d'extension

B.2.1 Entity

Tableau 2. Classes d'extension appliquées aux classes Entity

Action	Option de modèle	Déconseillée ?	Alternative
Ajouter une opération d'entité stéréotypée (par exemple, <<ns>>, <<nsreadmulti>>)	Aucune	Déconseillée	Au lieu d'utiliser une classe d'<<extension>>, ajouter l'opération stéréotypée via une sous-classe sans remplacement.
Changer une opération d'entité (par exemple, les paramètres)	Aucune	Déconseillée	Créer une opération stéréotypée avec la structure souhaitée à l'aide d'une sous-classe sans remplacement. Si vous avez une raison valable pour changer la structure de l'opération d'entité d'une application, créez un dossier de support.
Changer une option d'opération Entity	Zone ID auto Clé ID auto Aucun code SQL généré Verrouillage optimiste Tri par SQL Where	Déconseillée	Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement. Si vous avez une raison valable pour changer ces options dans les opérations Entity de l'application, créez un dossier de support.
	Contrôle au niveau de la table de base de données	Déconseillée	Cette option est paramétrable via les propriétés d'exécution. Pour plus d'informations sur cette procédure, voir la section 12.3.6 du guide Cúram Server Modeling Guide.

Tableau 2. Classes d'extension appliquées aux classes Entity (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	Opération en cas d'échec Opération postaccès aux données Opération préaccès aux données Treater ReadmultiMax comme Informational Exception Enregistrements ReadMultiMax renvoyés	Déconseillée	Les clients doivent uniquement changer ces options sur les opérations Entity de l'application en utilisant une sous-classe avec remplacement.
Changer une option de classe Entity	Activer la validation	Déconseillée	Les clients doivent uniquement changer cette option sur les opérations Entity de l'application en utilisant une sous-classe avec remplacement.
	Résumé Autoriser le verrouillage optimiste Aucun code SQL généré Remplacer la superclasse	Déconseillée	Si vous avez une raison valable pour changer ces options dans les opérations Entity de l'application, créez un dossier de support.
	Contrôler les zones Dernière mise à jour de la zone	Autorisée	Cette action est actuellement uniquement prise en charge via les classes d'extension, et elle continuera à l'être à partir de la version 6.0.3.
Ajouter un attribut Entity	Aucune	Déconseillée	Les clients qui souhaitent ajouter des données aux écrans d'application doivent ajouter de nouvelles classes Entity spécifiques au client et encapsuler les opérations Cúram CRUD dans leurs propres classes de processus afin de gérer les deux tables atomiquement. Les écrans Cúram peuvent ensuite être modifiés en vue de pointer vers les nouvelles classes de processus.
Changer une option d'attribut Entity	Autoriser les valeurs null	Déconseillée	Si vous avez une raison valable pour changer cette option sur les attributs Entity de l'application, créez un dossier de support.

B.2.2 Struct

Tableau 3. Classes d'extension appliquées aux classes Struct

Action	Option de modèle	Déconseillée ?	Alternative
Ajouter un attribut à une structure	Aucune	Déconseillée	Créer une structure spécifique au projet et agréger la structure d'application de la structure spécifique au projet vers la structure d'application (et non l'inverse). Utiliser la nouvelle structure "composite" dans la fonctionnalité spécifique au client requise.
Changer un attribut Struct	Aucune	Déconseillée	Créer une structure spécifique au projet et agréger la structure d'application de la structure spécifique au projet vers la structure d'application (et non l'inverse). Utiliser la nouvelle structure "composite" dans la fonctionnalité spécifique au client requise. Si vous avez une raison valable pour changer l'un des attributs d'une structure d'application, créez un dossier de support.

Tableau 3. Classes d'extension appliquées aux classes Struct (suite)

Action	Option de modèle	Déconseillée ?	Alternative
Changer une option Struct	Contrôler les zones	Déconseillée	Pour propager le contrôle des zones d'une entité via un écran, vous devez créer de nouvelles opérations stéréotypées afin de conserver le contrôle des zones, créer une façade qui encapsule les opérations CRUD existantes de l'entité et appelle les nouvelles opérations stéréotypées, et mettre à jour les pages du gestionnaire d'interface utilisateur si nécessaire.

B.2.3 Process, Facade, WebService, WSInbound

Tableau 4. Classes d'extension appliquées à d'autres classes modélisées

Action	Option de modèle	Déconseillée ?	Alternative
Changer une option de classe	Résumé Générer des ID format Remplacer la superclasse Style de liaison de Web Services Web Services est un document XML Type de document Générer un bean de façade Nom du fournisseur Demande gestionnaires Réponse gestionnaires Valider la demande Document XML Schéma XML	Déconseillée	Si vous avez une raison valable pour changer ces options dans les classes Process, Facade, WebService ou WSInbound de l'application, créez un dossier de support.
Ajouter une opération	Aucune	Déconseillée	Cela n'a jamais été conseillé via les classes d'extension, car les clients auraient dû effectuer la modification du code Java de l'application sur place. Si vous souhaitez ajouter une opération à une classe Process, Facade, WebService ou WSInbound, encapsulez la classe et l'opération dans vos propres classe et opération spécifiques au projet.
Changer une opération (par exemple, la visibilité de l'opération)	Aucune	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire. S'il n'existe aucun point d'extension approprié, mais si vous avez une raison valable pour changer le fonctionnement ou la structure d'une opération de l'application, créez un dossier de support.
Changer une option d'opération	Contrôler les appels de veille économique	Déconseillée	Cette option est paramétrable via les propriétés d'exécution. Pour plus d'informations sur cette procédure, voir la section 12.3.6 du guide Cúram Server Modeling Guide.

Tableau 4. Classes d'extension appliquées à d'autres classes modélisées (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	Zone de date métier Jeu de caractères pour le codage des messages en octets Générer la sécurité XA est-il transactionnel Type de message Nom JNDI de la fabrique de connecteur en file d'attente Nom JNDI de la file d'attente de réponses Dépassement du délai d'attente du message de réponse Type d'ombre Transactionnel Nom JNDI de la file d'attente de transmission	Déconseillée	Si vous avez une raison valable pour changer l'une de ces options dans les opérations Process, Facade, WSInbound ou WebService de l'application, créez un dossier de support.
	Sécuriser les zones	Déconseillée	Les clients qui souhaitent modifier les zones d'opération d'une application à traiter comme sécurisées doivent encapsuler l'opération dans leurs propres classe et opération Facade, puis définir l'option Sécuriser les zones de cette nouvelle opération sur le paramètre souhaité. Si nécessaire, les définitions d'écran affectées du gestionnaire d'interface utilisateur doivent être redirigées sur la nouvelle opération.
Changer une option du paramètre d'opération	Zones obligatoires	Déconseillée	Les clients qui souhaitent modifier les zones d'opération d'une application à traiter comme obligatoires doivent encapsuler l'opération dans leurs propres classe et opération Facade, puis définir l'option Zones obligatoires de cette nouvelle opération sur le paramètre souhaité. Si nécessaire, les définitions d'écran affectées du gestionnaire d'interface utilisateur doivent être redirigées sur la nouvelle opération.

B.3 Sous-classe avec remplacement

B.3.1 Entity

Tableau 5. Sous-classe avec remplacement appliquée aux classes Entity

Action	Option de modèle	Déconseillée ?	Alternative
Ajouter une opération Entity stéréotypée (par exemple, <<ns>>, <<nsreadmulti>>, etc.)	Aucune	Déconseillée	Au lieu d'utiliser une sous-classe avec remplacement, ajouter l'opération stéréotypée en utilisant une sous-classe sans remplacement. Cela permet de garantir que votre sous-classe (et, par conséquent, vos nouvelles opérations stéréotypées) sera traitée comme externe et que vous ne recevrez pas d'avertissements dans Eclipse lorsque vous la référencerez. Notez que vous recevrez des avertissements si vous référencez directement les opérations stéréotypées dans l'entité de base, car elles sont internes par défaut.

Tableau 5. Sous-classe avec remplacement appliquée aux classes Entity (suite)

Action	Option de modèle	Déconseillée ?	Alternative
Ajouter ou changer une opération Entity non stéréotypée	Aucune	Déconseillée	<p>Au lieu d'utiliser une sous-classe avec remplacement, ajouter une opération non stéréotypée en utilisant une sous-classe sans remplacement. Cela permet de garantir que votre sous-classe (et, par conséquent, vos nouvelles opérations non stéréotypées) sera traitée comme externe et que vous ne recevrez pas d'avertissements dans Eclipse lorsque vous la référencerez.</p> <p>Notez que vous recevrez des avertissements si vous référencez directement les opérations dans l'entité de base, car elles sont internes par défaut.</p> <p>Il est déconseillé aux clients de fournir de nouvelles implémentations pour les opérations Entity non stéréotypées de l'application.</p>
Changer la structure d'une opération Entity	Aucune	Déconseillée	<p>Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement.</p> <p>Si vous avez une raison valable pour changer la structure de l'opération d'entité d'une application, créez un dossier de support.</p>
Changer une option d'opération Entity	Zone ID auto Clé ID auto Aucun code SQL généré Verrouillage optimiste Tri par SQL Where	Déconseillée	<p>Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement.</p> <p>Si vous avez une raison valable pour changer ces options dans les opérations Entity de l'application, créez un dossier de support.</p>
	Contrôle au niveau de la table de base de données	Déconseillée	Cette option est paramétrable via les propriétés d'exécution. Pour plus d'informations sur cette procédure, voir la section 12.3.6 du guide Cúram Server Modeling Guide.
	Opération en cas d'échec Opération post-acès aux données Opération pré-acès aux données	Autorisée (en partie)	<p>Les clients sont toujours autorisés à implémenter les points d'exit Entity de l'application.</p> <p>Si les clients souhaitent effectuer un traitement dans les points d'exit pour lesquels il existe une implémentation par défaut, l'implémentation par défaut doit être appelée au début de l'implémentation du point d'exit client (c'est-à-dire qu'il doit y avoir un appel 'super()' au début).</p> <p>Les clients ne sont pas autorisés à quitter les implémentations de point d'exit de l'application.</p>
	Treater ReadMultiMax comme Informational Exception Enregistrements ReadMultiMax renvoyés	Autorisée	
Changer une option de classe Entity	Activer la validation	Autorisée (en partie)	<p>Les clients sont toujours autorisés à implémenter les points d'exit Entity de l'application.</p> <p>Si les clients souhaitent effectuer un traitement dans les points d'exit pour lesquels il existe une implémentation par défaut, l'implémentation par défaut doit être appelée au début de l'implémentation du point d'exit client (c'est-à-dire qu'il doit y avoir un appel 'super()' au début).</p>

Tableau 5. Sous-classe avec remplacement appliquée aux classes Entity (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	Résumé Autoriser le verrouillage optimiste Aucun code SQL généré	Déconseillée	Si vous avez une raison valable pour changer ces options dans les opérations Entity de l'application, créez un dossier de support.
	Contrôler les zones Dernière mise à jour de la zone	Déconseillée	Utiliser les classes d'extension pour substituer ces options sur une classe Entity de l'application.
	Remplacer la superclasse	Autorisée (en partie)	Implicitement autorisée pour prendre en charge les autres actions 'Autorisées' décrites dans cette section.

B.3.2 Process, Facade, WebService, WSInbound

Tableau 6. Sous-classe avec remplacement appliquée à d'autres classes modélisées

Action	Option de modèle	Déconseillée ?	Alternative
Changer une option de classe	Résumé Générer des ID format Remplacer la superclasse Style de liaison de Web Services Web Services est un document XML Type de document Générer un bean de façade Nom du fournisseur Demande gestionnaires Réponse gestionnaires Valider la demande Document XML Schéma XML	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire, et en définissant les options appropriées dans la nouvelle classe. Si vous avez une raison valable pour changer directement ces options dans les classes Process, Facade, WebService ou WSInbound de l'application, créez un dossier de support.
Ajouter une opération	Aucune	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire.
Changer une opération	Aucune	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire. Si vous avez une raison valable pour changer directement la structure des opérations dans les classes Process, Facade, WebService ou WSInbound de l'application, créez un dossier de support.
Changer une option d'opération	Contrôler les appels de veille économique	Déconseillée	Cette option est paramétrable via les propriétés d'exécution. Pour plus d'informations sur cette procédure, voir la section 12.3.6 du guide Cúram Server Modeling Guide.

Tableau 6. Sous-classe avec remplacement appliquée à d'autres classes modélisées (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	<p>Zone de date métier</p> <p>Jeu de caractères pour le codage des messages en octets</p> <p>Générer la sécurité</p> <p>XA est-il transactionnel</p> <p>Type de message</p> <p>Nom JNDI de la fabrique de connecteur en file d'attente</p> <p>Nom JNDI de la file d'attente de réponses</p> <p>Dépassement du délai d'attente du message de réponse</p> <p>Type d'ombre</p> <p>Transactionnel</p> <p>Nom JNDI de la file d'attente de transmission</p>	Déconseillée	Si vous avez une raison valable pour changer l'une de ces options dans les opérations Process, Facade, WSInbound ou WebService de l'application, créez un dossier de support.
	Sécuriser les zones	Déconseillée	<p>Les clients qui souhaitent modifier les zones d'opération d'une application à traiter comme sécurisées doivent encapsuler l'opération dans leur propre opération, puis définir l'option Sécuriser les zones de cette nouvelle opération sur le paramètre souhaité.</p> <p>Si nécessaire, les définitions d'écran affectées du gestionnaire d'interface utilisateur doivent être redirigées sur la nouvelle opération.</p>
Changer une option du paramètre d'opération	Zones obligatoires	Déconseillée	<p>Les clients qui souhaitent modifier les zones d'opération d'une application à traiter comme obligatoires doivent encapsuler l'opération dans leur propre opération, puis définir l'option Zones obligatoires de cette nouvelle opération sur le paramètre souhaité.</p> <p>Si nécessaire, les définitions d'écran affectées du gestionnaire d'interface utilisateur doivent être redirigées sur la nouvelle opération.</p>

B.4 Sous-classe sans remplacement

B.4.1 Entity

Tableau 7. Sous-classe sans remplacement appliquée aux classes Entity

Action	Option de modèle	Déconseillée ?	Alternative
Ajouter une opération Entity stéréotypée (par exemple, <<ns>>, <<nsreadmulti>>, etc.)	Aucune	Autorisée	<p>Au lieu d'utiliser une sous-classe avec remplacement, ajouter l'opération stéréotypée en utilisant une sous-classe sans remplacement. Cela permet de garantir que votre sous-classe (et, par conséquent, vos nouvelles opérations stéréotypées) sera traitée comme externe et que vous ne recevrez pas d'avertissements dans Eclipse lorsque vous la référencerez.</p> <p>Notez que vous recevrez des avertissements si vous référencez directement les opérations dans l'entité de base, car elles sont internes par défaut.</p>
Ajouter une opération Entity non stéréotypée	Aucune	Autorisée	<p>Au lieu d'utiliser une sous-classe avec remplacement, ajouter l'opération non stéréotypée en utilisant une sous-classe sans remplacement. Cela permet de garantir que votre sous-classe (et, par conséquent, vos nouvelles opérations non stéréotypées) sera traitée comme externe et que vous ne recevrez pas d'avertissements dans Eclipse lorsque vous la référencerez.</p> <p>Notez que vous recevrez des avertissements si vous référencez directement les opérations dans l'entité de base, car elles sont internes par défaut.</p>
Changer la structure d'une opération Entity	Aucune	Déconseillée	Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement.
Changer une option d'opération Entity	Zone ID auto Clé ID auto Aucun code SQL généré Verrouillage optimiste Tri par SQL Where	Déconseillée	Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement.
	Contrôle au niveau de la table de base de données	Déconseillée	<p>Cette option est paramétrable via les propriétés d'exécution si vous souhaitez changer le comportement des opérations d'application.</p> <p>Il est également possible de créer une opération stéréotypée pour implémenter la fonctionnalité requise en utilisant une sous-classe sans remplacement.</p>
	Opération en cas d'échec Opération postaccès aux données Opération préaccès aux données Treater ReadmultiMax comme InformationalException Enregistrements ReadMultiMax renvoyés	Déconseillée	<p>Utiliser la sous-classe avec remplacement pour substituer ces options sur une classe Entity de l'application.</p> <p>Il est également possible de créer une opération stéréotypée pour implémenter la fonctionnalité requise en utilisant une sous-classe sans remplacement.</p>
Changer une option de classe Entity	Activer la validation	Déconseillée	<p>Utiliser la sous-classe avec remplacement pour substituer cette option sur une classe Entity de l'application.</p> <p>Il est également possible de créer une opération stéréotypée pour implémenter la fonctionnalité requise en utilisant une sous-classe sans remplacement.</p>

Tableau 7. Sous-classe sans remplacement appliquée aux classes Entity (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	Résumé Autoriser le verrouillage optimiste Aucun code SQL généré	Déconseillée	Créer une opération stéréotypée à l'aide d'une sous-classe sans remplacement.
	Contrôler les zones Dernière mise à jour de la zone	Déconseillée	Utiliser les classes d'extension pour substituer ces options sur une classe Entity de l'application.

B.4.2 Process, Facade, WebService, WSInbound

Tableau 8. Sous-classe sans remplacement appliquée à d'autres classes modélisées

Action	Option de modèle	Déconseillée ?	Alternative
Changer une option de classe	Résumé Générer des ID format Remplacer la superclasse Style de liaison de Web Services Web Services est un document XML Type de document Générer un bean de façade Nom du fournisseur Demande gestionnaires Réponse gestionnaires Valider la demande Document XML Schéma XML	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire, et en définissant les options appropriées dans la nouvelle classe.
Ajouter une opération	Aucune	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire, et en définissant les options appropriées dans la nouvelle opération de classe.
Changer une opération	Aucune	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire.

Tableau 8. Sous-classe sans remplacement appliquée à d'autres classes modélisées (suite)

Action	Option de modèle	Déconseillée ?	Alternative
Changer une option d'opération	<p>Contrôler les appels de veille économique</p> <p>Zone de date métier</p> <p>Jeu de caractères pour le codage des messages en octets</p> <p>Générer la sécurité</p> <p>XA est-il transactionnel</p> <p>Type de message</p> <p>Nom JNDI de la fabrique de connecteur en file d'attente</p> <p>Nom JNDI de la file d'attente de réponses</p> <p>Dépassement du délai d'attente du message de réponse</p> <p>Type d'ombre</p> <p>Transactionnel</p> <p>Nom JNDI de la file d'attente de transmission</p> <p>Sécuriser les zones</p>	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire, et en définissant les options appropriées dans la nouvelle opération de classe.
Changer une option du paramètre d'opération	Zones obligatoires	Déconseillée	Créer une opération dans une classe spécifique au projet, en encapsulant les API externes de la fonctionnalité de la classe d'application si nécessaire, et en définissant les options appropriées dans la nouvelle classe.

B.5 Substitution de domaine

B.5.1 Définitions de domaines

Tableau 9. Substitution des définitions de domaines

Action	Option de modèle	Déconseillée ?	Alternative
Changer la définition de domaine d'une application spécifique		Déconseillée (en partie)	La personnalisation des définitions de domaines suivantes de l'application n'est pas autorisée par les clients : TRUNCATED_NOTE_TEXT NOTE_TEXT INCIDENT_DESCRIPTION SERVICE_DELIVERY_NOTE_TEXT_SMALL INJURY_DESCRIPTION ACTION_TAKEN CITIZEN_ACCOUNT_RICH_TEXT CW_RICH_STRING VIEW_LIFE_EVENTS_POST_SUBMIT_XML_DATA RICH_TEXT_EDITOR_WIDGET SEC_RICH_TEXT_VIEW_WIDGET RICH_TEXT PROGRESS_RICH_TEXT_SMALL
Changer une option de la définition de domaine	Nom de la table de codes Racine de la table de codes	Déconseillée	Créer une définition de domaine avec un nom et une racine appropriés pour la table de codes, puis l'encapsuler dans son propre traitement. Les clients ne sont pas autorisés à changer ces options pour les définitions de domaines de l'application.

Tableau 9. Substitution des définitions de domaines (suite)

Action	Option de modèle	Déconseillée ?	Alternative
	Compresser les espaces imbriqués Convertir en majuscule Nom de la fonction de validation personnalisée Valeur par défaut Valeur maximale Taille minimale Valeur minimale Critère de concordance Retirer les espaces à gauche Retirer les espaces de fin de chaîne Type de stockage	Autorisée	
	Taille maximale	Autorisée (en partie)	Autorisée pour l'augmentation de la taille uniquement. Si vous souhaitez réduire la taille de la définition de domaine d'une application, créez un dossier de support. Ne pas utiliser pour changer la taille maximale de la définition de domaine USERNAME.
Changer le type d'une définition de domaine	Aucune	Déconseillée	Créer une définition de domaine avec un type approprié, puis l'encapsuler dans son propre traitement. Les clients ne sont pas autorisés à changer les types fondamentaux des définitions de domaines de l'application.
Changer la longueur de chaîne d'une définition de domaine	Aucune	Autorisée (en partie)	Autorisée pour l'augmentation de la taille uniquement. Si vous souhaitez réduire la taille de la définition de domaine d'une application, créez un dossier de support. Ne pas utiliser pour changer la longueur de chaîne de la définition de domaine USERNAME.
Créer une définition de domaine en fonction de la définition de domaine d'une application	Aucune	Autorisée	

B.6 Relations

B.6.1 Affectable

Tableau 10. Relations affectables

Action	Déconseillée ?
Rendre une structure fournie par le client affectable à une structure d'application ou une entité	Autorisée
Rendre une structure d'application affectable à une autre structure d'application ou entité	Déconseillée

B.6.2 Agrégation

Tableau 11. Agrégations

Action	Déconseillée ?
Agréger une structure d'application dans une structure fournie par le client (c'est-à-dire créer une structure client 'contenant' une structure d'application)	Autorisée
Agréger une structure fournie par le client ou une structure d'application dans une structure d'application (c'est-à-dire ajouter une structure à une structure d'application grâce à l'agrégation)	Déconseillée

B.6.3 Clé externe

Tableau 12. Clés externes

Action	Déconseillée ?
Créer une clé externe lorsque l'entité fournie par le client est l'enfant	Autorisée
Créer une clé externe lorsque l'entité de l'application est l'enfant	Déconseillée

B.6.4 Index

Tableau 13. Index

Action	Déconseillée ?
Créer un index (sur une application ou une entité fournie par le client) à l'aide d'une structure fournie par le client	Autorisée
Créer un index (sur une application ou une entité fournie par le client) à l'aide d'une structure d'une application	Déconseillée

B.6.5 Index à entrées uniques

Tableau 14. Index à entrées uniques

Action	Déconseillée ?
Créer un index à entrées uniques dans une entité d'application	Déconseillée
Créer un index à entrées uniques dans une entité fournie par un fournisseur à l'aide d'une structure d'application	Déconseillée
Créer un index à entrées uniques dans une entité fournie par un client à l'aide d'une structure fournie par un client	Autorisée

B.7 Autres mécanismes

B.7.1 Exclusions

Tableau 15. Exclusions

Action	Déconseillée ?
Utiliser les exclusions pour tenter d'exclure des classes depuis une génération de serveur	Déconseillée ; les classes d'application ne sont pas systématiquement régénérées depuis la version 6.0.3.

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM. IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Licence sur la propriété intellectuelle
Mentions légales et droit de propriété intellectuelle
IBM Japon Ltd
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. INTERNATIONAL BUSINESS MACHINES CORPORATION FOURNIT CETTE PUBLICATION "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, EXPLICITE OU IMPLICITE, Y COMPRIS NOTAMMENT, LES GARANTIES IMPLICITES DE NON-CONTREFACON, DE QUALITE MARCHANDE OU D'ADEQUATION A UN USAGE PARTICULIER. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies. Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
Dept F6, Bldg 1
294 Route 100
Somers NY 10589-3216
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM, conformément aux dispositions du Livret contractuel, des Conditions Internationales d'Utilisation de Logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles.

IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être modifiés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les

exemples de programmes sont fournis "EN L'ETAT", sans garantie d'aucune sorte. IBM décline toute responsabilité relative aux dommages éventuels résultant de l'utilisation de ces exemples de programmes.

Toute copie intégrale ou partielle de ces exemples de programmes et des oeuvres qui en sont dérivées doit inclure une mention de droits d'auteur libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des exemples de programmes d'IBM Corp.

© Copyright IBM Corp. _entrez l'année ou les années_. Tous droits réservés.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

IBM, le logo IBM et [ibm.com](http://www.ibm.com) sont des marques ou des marques déposées d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. Une liste des marques commerciales actuelles d'IBM est disponible sur Internet sous "Droits d'auteur et marques" à l'adresse <http://www.ibm.com/legal/us/en/copytrade.shtml>.

D'autres noms peuvent être des marques de leurs propriétaires respectifs. Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

