

InstallAnywhere 6

Table des matières

Introduction	1
InstallAnywhere – Présentation générale	1
Structure des informations	1
Environnement requis	1
Versions d’InstallAnywhere	2
DTS – Services techniques Zero G pour développeurs.....	3
Enregistrement	4
Convertir des projets IA antérieurs à 5.5	5
Copyrights et marques déposées	6
Concepts de base.....	7
Environnements de développement d’installateurs	7
Types d’installateur	8
Modes de l’installateur	8
Types d’installation, fonctions et composants	9
Interface graphique (GUI) de l’installateur	10
Actions	12
Règles	12
Désinstallateur.....	12
LaunchAnywhere.....	13
PowerUpdate	13
Magic Folders.....	13
SpeedFolders.....	14
Variables InstallAnywhere.....	14
Sous-projets d’installateur – Modules de fusion.....	15
Modèles – Présentation générale	15
Fichier de projet.....	15
Didacticiels – À la découverte d’InstallAnywhere.....	17
Didacticiel d’introduction – Construire un installateur.....	17
Didacticiel avancé – Utiliser Advanced Designer	20
Solutions	29
Intégrer un environnement d’exécution Java (JRE) à l’installateur	29
Créer un lanceur pour une application Java	29
Organiser les fonctions et les composants	30
Améliorer les installations.....	32
Installateurs complémentaires.....	33
Équipe de développement.....	33
Bilan de l’installation.....	35

Table des matières

Débogage des installateurs	35
Assistant de projet	39
Nouveau projet	39
Définir les détails	39
Ajouter des fichiers.....	39
Choisir la classe main	40
Définir le classpath	40
Construire l'installateur	40
Tester l'installateur.....	40
Advanced Designer.....	41
Projet	41
Interface d'installation	46
Structure	48
Fichiers.....	50
Tâches Pré-installation, Post-installation et Désinstallation	52
Construction du projet	52
Menus InstallAnywhere	55
Menu Fichier	55
Menu Édition	56
Assistant de projet	58
Menu Aide.....	58
Actions InstallAnywhere	61
Actions d'installation	62
Actions générales	64
Actions par panneaux	68
Actions par consoles	71
Propriétés communes	72
Actions par panneaux – Paramètres	73
LaunchAnywhere et fichiers LAX	74
Format des fichiers de manifeste	80
Règles	83
Variables InstallAnywhere	85
Magic Folders et variables InstallAnywhere	93
Code personnalisé.....	99
Code personnalisé et variables InstallAnywhere	99
Créer et exécuter du code personnalisé dans un installateur	100
Internationaliser le code personnalisé.....	101
Plug-ins	102
Désinstallateur	105

Code personnalisé	105
Désinstaller certaines fonctions	105
Désinstallateur pour plusieurs produits	106
Modules de fusion et modèles	107
Modules de fusion	107
Modèles	109
Installeurs en mode Console ou en mode Silencieux	111
Installeurs en mode Silencieux	111
Installeurs par consoles	112
Remarques spécifiques aux différentes plates-formes	113
UNIX	113
Mac OS X	113
z/OS et OS/390	114
Outils de construction	117
Options des outils de construction.....	117
Codes en sortie	118
Construire avec l'outil Ant.....	120
Localisation	125
Localisation (installateurs en langues étrangères)	125
Texte statique et texte dynamique.....	125
Localiser les installateurs et Advanced Designer.....	126
Conseils de localisation	126
Codes de langue	128
Éléments localisables	129
Index	133

Introduction

InstallAnywhere – Présentation générale

Solution puissante pour le déploiement de logiciels multiplate-forme, InstallAnywhere supporte toutes les applications sur toutes les plates-formes. Avec InstallAnywhere, vous disposez d'un contrôle absolu, de performances inégalées et d'une interface graphique ergonomique qui vous permettront de créer des installateurs de qualité tout en réduisant sensiblement vos coûts de développement et de support.

Structure des informations

Les informations proposées sont regroupées en plusieurs sections :

Introduction – L'introduction décrit l'environnement requis pour l'exécution d'InstallAnywhere, et présente des informations sur les trois versions d'InstallAnywhere. Elle fournit également des informations sur les services professionnels, le support technique et la formation proposés par Zero G. Ce chapitre présente enfin des sections expliquant comment enregistrer InstallAnywhere auprès de Zero G, et comment convertir des projets InstallAnywhere existants.

Concepts de base – Cette section présente certains concepts de base qui doivent être connus des développeurs avant de créer des installateurs à l'aide d'InstallAnywhere. Cette description de l'environnement conceptuel d'InstallAnywhere doit permettre aux développeurs de comprendre et d'assimiler les nombreuses options disponibles. En général, une rubrique complète est consacrée à chaque concept.

Didacticiels – Ces deux didacticiels décrivent les étapes à suivre pour construire un installateur InstallAnywhere. Le didacticiel d'introduction présente l'Assistant de projet. Le didacticiel avancé décrit des procédures qui démontrent la puissance et la souplesse d'Advanced Designer. Il est fortement conseillé de suivre ces deux didacticiels.

Solutions – Cette section propose des solutions en réponse aux difficultés qui peuvent apparaître lors de la construction d'un installateur.

Autres sections – Présentent des informations détaillées sur les concepts de base, les fonctionnalités et les options d'InstallAnywhere.

Environnement requis

Environnement de développement d'installateurs

L'« environnement de développement d'installateurs » d'InstallAnywhere (Assistant de projet + Advanced Designer) exige un système avec au minimum 256 Mo de RAM et une résolution d'écran 1024 x 768 sur 8 bits (256 couleurs). L'environnement de développement d'InstallAnywhere supporte les versions les plus récentes – avec application des mises à jour, Service Packs et correctifs/patches appropriés – des systèmes suivants : Windows (98, Me, NT, 2000, XP, Server 2003), Mac OS X, Solaris, Linux, AIX, HP-UX et la plupart des autres plates-formes UNIX. Pour toute plate-forme non mentionnée explicitement ci-dessus, une VM Java 1.3 doit être installée avant InstallAnywhere.

Environnement de l'installateur

Les installateurs construits à l'aide d'InstallAnywhere peuvent être exécutés sur tout système doté au minimum de 64 Mo de RAM et d'une résolution d'écran 640 x 480 sur 8 bits (256 couleurs). Une VM Java 1.1.8 ou ultérieure (ou Microsoft VM 3167 ou ultérieure) est également requise ; elle peut être préinstallée sur le système cible ou intégrée à l'installateur lors de la construction de celui-ci. Zero G propose différents types de VM Java à l'adresse suivante : <http://www.zerog.com/goto/vmpacks>. Les installateurs construits par InstallAnywhere peuvent être exécutés sur la plupart des versions des systèmes suivants : Windows (95, 98, Me, NT, 2000, XP, Server 2003), Mac OS X, Solaris, Linux, AIX, HP-UX, Tru64, BSD, IRIX, NetWare, z/OS et OS/390. Par ailleurs, les installateurs InstallAnywhere peuvent être exécutés sur la plupart des plates-formes non mentionnées ci-dessus qui disposent d'une VM Java version 1.1.8 ou ultérieure.

Les installateurs InstallAnywhere sont compatibles avec les versions suivantes :

- Java 1.1.8 (1.1.8_10/16 conseillée), ou VM Microsoft 3167
- Java 1.2.1 - 1.2.2 (1.2.2_16 conseillée)
- Java 1.3.0 - 1.3.1 (1.3.1_09 conseillée)
- Java 1.4.0 - 1.4.2 (1.4.2_01 conseillée)

Versions d'InstallAnywhere

InstallAnywhere existe dans trois versions différentes. Chaque version a été conçue pour répondre à différents besoins de déploiement. La présente aide décrit les caractéristiques et les fonctionnalités de la version Enterprise. Dans certaines rubriques, les variations entre versions ont été signalées. Toutefois, pour consulter un tableau détaillé des fonctionnalités disponibles dans chaque version, nous vous invitons à consulter les pages suivantes : <http://www.zerog.com/goto/editions>.

InstallAnywhere Enterprise Edition

La version « Enterprise Edition » d'InstallAnywhere est la plus complète en matière d'options de configuration, d'interaction utilisateur et de fonctionnalités client/serveur. Elle facilite la mise en place d'installations complexes et elle met à la disposition des développeurs une souplesse de personnalisation exceptionnelle. Enterprise Edition est disponible dans différentes variantes linguistiques (anglais, français, allemand et japonais) et supporte la création d'installateurs en 29 langues.

InstallAnywhere Standard Edition

La version « Standard Edition » d'InstallAnywhere propose plus de fonctionnalités et de potentiel de personnalisation que les autres produits de sa catégorie ; idéale pour le déploiement des applications de bureau, elle supporte la création d'installateurs en neuf langues.

InstallAnywhere Now!

La version « Now! » d'InstallAnywhere est la solution la plus rapide, la plus simple et la plus économique pour la construction d'installateurs multiplate-forme.

DTS – Services techniques Zero G pour développeurs

Support technique

Les services DTS de Zero G proposent aux développeurs des services techniques complets et efficaces, pour notamment :

- Analyser les problèmes d'installation et de configuration
- Optimiser le développement en proposant des solutions rapides et efficaces aux problèmes rencontrés
- Faciliter le déploiement de mises à jour bénéficiant des technologies les plus avancées
- Proposer une assistance qualifiée pour permettre aux développeurs d'exploiter au mieux les fonctionnalités de leurs produit

Pour plus de détails sur les options de support technique proposées par Zero G, visitez les pages <http://www.zerog.com/support>

Support technique gratuit pour évaluation – Zero G propose des services de support technique avant-vente gratuits (e-mail et téléphone) pendant que les développeurs testent la version d'évaluation d'InstallAnywhere.

Support technique en ligne et en libre service – La base de connaissances de Zero G (Technical Support KnowledgeBase) et sa Foire aux questions (FAQ) sont accessibles en permanence sur notre site Web. Les forums mis en place par Zero G (Community Forums) permettent de poser des questions, de signaler des problèmes et de partager des idées sur InstallAnywhere avec les techniciens Zero G et les autres clients Zero G. La section « Plug-ins » du site Web de Zero G propose aux développeurs de nombreux produits en téléchargement gratuit (Packs VM, modules de fusion, code personnalisé, etc.).

Premium Support – Niveau le plus élevé du support technique Zero G, sur abonnement. Principaux avantages : support technique (téléphone et e-mail) sans aucune restriction, mises à jour gratuites et assistance aux solutions de déploiement des développeurs. Les abonnés des services Premium Support reçoivent une réponse en priorité aux messages qu'ils publient sur les forums Zero G (Community Forums).

Services de formation Zero G

Pour compléter rapidement les compétences InstallAnywhere des développeurs et les aider à surmonter les difficultés du déploiement multiplate-forme, Zero G a élaboré un programme de formation complet (Zero G University). Quel que soit le niveau de base des développeurs – du néophyte à l'expert confirmé –, la méthode de formation proposée par Zero G leur garantit la maîtrise d'InstallAnywhere en trois jours seulement ! Terminologie, conseils et « meilleures pratiques » sont proposés aux

développeurs par des instructeurs Zero G impliqués dans la conception et le développement des produits InstallAnywhere. Les cours étant limités à 15 personnes, les instructeurs ont toujours le temps de répondre aux questions des étudiants ou de discuter des problèmes spécifiques à un projet. Des exemples réels, un suivi intense et individualisé et des « travaux pratiques » préparent les développeurs à optimiser le déploiement de leurs logiciels sur toute plate-forme (client ou serveur). Sur demande, Zero G peut également organiser des cours privés ou sur site pour répondre à des besoins spécifiques. Pour plus de détails, consultez les pages <http://www.zerog.com/goto/training>.

Services professionnels

Zero G propose également une équipe de services professionnels (Professional Services), qui peut définir des solutions adaptées à un contexte spécifique ou à des besoins d'installation complexes.

Création d'installateurs par Zero G

Lorsqu'une solution d'installateur doit être disponible rapidement, l'équipe Professional Services de Zero G – qui a développé plusieurs milliers d'installateurs – peut résoudre les problèmes les plus complexes à votre place : sélectionnez les options d'installation requises et confiez à Zero G la création de vos installateurs dans le respect de vos délais.

Développement d'installateurs personnalisés par Zero G

Pour les installations très complexes et exigeant l'expertise de spécialistes, Zero G peut analyser tous les aspects de vos projets d'installateur. Les technologies implémentées dans InstallAnywhere sont puissantes, personnalisables et évolutives ; la souplesse d'InstallAnywhere permet à l'équipe Professional Services de Zero G de sélectionner les technologies les mieux adaptées aux besoins de chaque projet.

Pour plus de détails sur la création d'un installateur par Zero G, consultez les pages suivantes : <http://www.zerog.com/goto/professionalservices>.

Enregistrement

Les versions d'essai proposées par Zero G pour permettre l'évaluation d'InstallAnywhere ne sont pas limitées : elles permettent aux développeurs de tester l'ensemble des fonctionnalités du produit. Remarque : la durée de vie des installateurs créés à l'aide d'une version d'essai non enregistrée d'InstallAnywhere est limitée à trois jours. Pour construire des installateurs sans restrictions, il suffit d'acheter une licence et d'enregistrer une copie d'InstallAnywhere (y compris versions d'essai – voir ci-dessous). Pour tout renseignement d'ordre commercial, faites parvenir un e-mail à l'adresse suivante : sales@ZeroG.com.

Remarque : il n'est pas nécessaire de reconstruire les projets créés avec les versions d'essai d'InstallAnywhere ; après avoir procédé à l'enregistrement auprès de Zero G d'une version de ce type, il suffit de relancer InstallAnywhere et de cliquer sur **Construire** : les installateurs existants sont immédiatement disponibles sans aucune restriction.

Les développeurs qui disposent d'une licence et d'un numéro de série peuvent enregistrer et activer leur copie d'InstallAnywhere en ligne (via Internet). Cette activation transmet à Zero G les informations suivantes : adresse IP, numéro de série et informations d'enregistrement. Aucune autre information n'est transmise à Zero G. Si l'accès à Internet est impossible, il suffit d'enregistrer ces informations dans un fichier et de faire parvenir celui-ci à l'adresse suivante :

register@ZeroG.com.

- Zero G s'engage à ne pas divulguer ces informations d'enregistrement, à ne jamais les vendre, les distribuer ni les utiliser à l'extérieur de Zero G. Zero G utilise ces informations uniquement pour sa base de données d'utilisateurs enregistrés.
- Le Contrat de licence n'autorise pas l'utilisation de plusieurs copies d'InstallAnywhere avec le même numéro de série.
- Pour plus de détails sur l'enregistrement et l'utilisation d'InstallAnywhere, contactez sales@ZeroG.com.

Si InstallAnywhere est installé derrière un pare-feu/firewall, il peut arriver que l'exécution de l'enregistrement dépasse le délai imparti ; dans ce cas, le processus crée un fichier texte qui peut ensuite être envoyé à l'adresse suivante sales@ZeroG.com pour procéder à l'enregistrement. Remarque à l'intention des clients ayant enregistré et utilisant une version d'InstallAnywhere antérieure à 6.0 : le numéro de série de l'ancienne version ne permet pas d'activer la version 6.0. Pour plus de détails sur la mise à jour d'une version antérieure, contactez le service commercial de Zero G à l'adresse suivante : sales@ZeroG.com.

Pour enregistrer InstallAnywhere :

Ouvrez InstallAnywhere. Dans le premier écran d'accueil, cliquez sur **Préférences**. Vous pouvez également choisir **Aide | Enregistrer** à tout moment.

1. Activez l'onglet **Enregistrement**.
2. Tapez le numéro de série et autres informations demandées, puis cliquez sur **Enregistrer**.
3. Quittez et relancez InstallAnywhere.

Convertir des projets IA antérieurs à 5.5

Les projets construits à l'aide d'InstallAnywhere 4.5 (ou versions antérieures) sont automatiquement convertis au nouveau format – actuellement, InstallAnywhere 6. Lorsque vous chargez un projet de ce type dans la nouvelle version d'Advanced Designer, un message vous signale que ce projet original va être sauvegardé. Si des composants sont générés automatiquement pour un projet converti, ils sont accompagnés de la remarque suivante :

AUTOGENERATED COMMENT / COMMENTAIRE AUTOGÉNÉRÉ : The following comment is an autogenerated aggregation of all the Feature descriptions from which this component was generated during project conversion. / Le commentaire suivant correspond à un récapitulatif des fonctions exécutées pour générer ce composant (exigé pour la conversion du projet).

Pour plus de détails sur les types d'installation, les fonctions et les composants, consultez la rubrique « Types d'installation, fonctions et composants ».

Copyrights et marques déposées

Copyright 1997-2003 Zero G – L'utilisation des logiciels InstallAnywhere (et de la présente aide) est limitée aux termes et conditions énoncés dans le Contrat de licence InstallAnywhere.

Le logo Zero G et les noms InstallAnywhere, Zero G, LaunchAnywhere, SpeedFolder(s), PowerUpdate et ZeroG.com sont des marques ou des marques déposées de la société Zero G.

Concepts de base

Cette section présente l'environnement conceptuel proposé par InstallAnywhere pour le développement d'un installateur et décrit les options complexes proposées pour définir les fonctionnalités d'un installateur.

Environnements de développement d'installateurs

InstallAnywhere propose deux environnements de développement : l'Assistant de projet guide les développeurs qui souhaitent créer un installateur rapidement et simplement ; le module Advanced Designer permet un plus grand contrôle des fonctionnalités de l'installateur. Advanced Designer permet de définir plusieurs types d'installation, d'insérer des graphiques personnalisés dans les écrans d'accueil, d'exécuter des commandes pendant l'installation et d'exécuter de nombreuses autres fonctions évoluées. Il est possible de construire un installateur dans l'Assistant de projet, puis de l'ouvrir dans Advanced Designer pour lui ajouter d'autres fonctionnalités.

Assistant de projet

En suivant les conseils de l'Assistant de projet, les développeurs peuvent construire un installateur en six étapes et en moins de cinq minutes. Pour les applications Java, cet outil de conception intuitif se charge également de définir le classpath et de retrouver automatiquement la classe main. Pour consulter un exemple de projet réalisé en suivant les conseils de l'Assistant de projet, suivez le didacticiel d'introduction.

Advanced Designer

Le module Advanced Designer d'InstallAnywhere est contrôlé par une interface graphique intuitive qui permet aux développeurs de gérer tous les aspects de leurs projets d'installateur. Toutes les fonctionnalités d'InstallAnywhere sont disponibles dans cet environnement de développement intégré (IDE).

Pour lancer Advanced Designer, sélectionnez un fichier de projet existant (ou créez un projet), puis cliquez sur **Advanced Designer**.

Advanced Designer est divisé en « tâches », qui sont représentées par les onglets regroupés dans la partie gauche de la fenêtre ; chaque onglet présente les tâches et les paramètres propres au projet d'installation considéré.

- **Projet** – Contient les paramètres qui seront utilisés dans le projet actuel (paramètres généraux, paramètres des fichiers et paramètres de localisation).
- **Interface de l'installateur** – Définit l'apparence et le comportement (« look and feel ») de l'installateur en ajoutant images de fond, billboards et autres éléments graphiques.
- **Structure** – Assure la gestion des types d'installation, des fonctions, des composants et des modules de fusion.
- **Fichiers** – Assure la gestion de l'arborescence d'installation des fichiers et des actions d'installation.

- **Pré-installation** – Séquence ordonnée de panneaux et d’actions qui est exécutée avant l’installation des fichiers.
- **Post-installation** – Séquence ordonnée de panneaux et d’actions qui est exécutée après l’installation des fichiers.
- **Pré-désinstallation** – Séquence ordonnée de panneaux et d’actions qui est exécutée avant la désinstallation des fichiers.
- **Post-désinstallation** – Séquence ordonnée de panneaux et d’actions qui est exécutée après la désinstallation des fichiers.
- **Construire** – Assure la gestion des paramètres de construction, y compris l’intégration d’une VM Java.

Remarque : chaque tâche Advanced Designer propose des sous-onglets qui permettent de configurer et d’optimiser les fonctions InstallAnywhere (un exemple est proposé à la page 21, dans le didacticiel avancé, qui utilise Advanced Designer).

Types d’installateur

InstallAnywhere propose deux types d’installateur : installateurs Web et installateurs CD.

Installateurs Web

Les installateurs Web se présentent sous la forme d’un seul fichier exécutable auto-extractible par plate-forme et conviennent à une distribution via le Web ou par e-mail. Les fichiers source étant décompressés au début de l’installation, il est nécessaire de prévoir un espace disque temporaire en conséquence.

Installateurs CD

Dans un installateur CD, la plupart des fichiers sont prêts pour installation ; il existe toujours un fichier auto-extractible, mais celui-ci extrait uniquement le « moteur » de l’installateur. Les installateurs CD occupent un espace disque temporaire moins important et démarrent plus rapidement, mais ils ne conviennent pas à une distribution via le Web ou par autre moyen électronique. Les installateurs CD peuvent également être distribués sur DVD. Ils peuvent également être répartis sur plusieurs CD ou DVD.

Modes de l’installateur

Les installateurs InstallAnywhere peuvent s’exécuter dans plusieurs modes :

Mode GUI – Interface utilisateur graphique avec Assistant de projet et boîtes de dialogue.

Mode Console (ou « Interface à ligne de commande ») – Mode adapté aux installations distantes via Telnet, ou aux systèmes ne disposant pas d’un environnement graphique à fenêtres.

Mode Silencieux – Dans ce mode, l’installateur s’exécute sans interaction avec l’utilisateur. Ce mode convient dans le cas où tous les paramètres sont connus d’avance (ou fournis dans un « fichier de réponse »).

Types d'installation, fonctions et composants

Dans l'environnement de développement d'installateurs InstallAnywhere, les types d'installation, les fonctions et les composants font partie des concepts les plus importants à assimiler.

InstallAnywhere propose aux utilisateurs plusieurs ensembles d'options d'installation (chaque utilisateur peut sélectionner le type d'installation et les fonctions qui lui conviennent). Un « type d'installation » est un groupe de fonctions, par exemple « Installation standard » ou « Installation minimum ». Les types d'installation et les fonctions correspondent à des ensembles de fichiers, mais il n'existe pas nécessairement de corrélation directe entre ces groupes structurels et les fichiers.

Les composants sont des groupes de fichiers et d'actions de l'installateur qui restent invisibles pour l'utilisateur. Un composant peut également inclure un groupe de modifications de registre ou d'autres éléments nécessaires à l'exécution correcte d'une fonction. Les composants sont utilisés pour le partage des ressources et pour le suivi des versions ; ils sont identifiés de manière unique et ils constituent des outils pour les développeurs de l'installateur et non pour les utilisateurs de l'installateur. Les développeurs peuvent par exemple créer un composant pour chaque fonction de l'application, un composant pour les bibliothèques partagées par les fonctions et un composant pour le système d'aide.

Bien qu'il soit possible d'affecter des fichiers, des dossiers et des actions directement aux fonctions du produit, il est préférable de considérer les fonctions comme des groupes de composants. Les types d'installation tels que « Installation standard » ou « Installation minimum » sont des groupes de fonctions. L'interaction entre ces trois niveaux doit être prise en compte au moment de décider des options qui seront proposées à l'utilisateur.

Types d'installation

Le type d'installation est le concept structurel le plus simple et le plus général d'InstallAnywhere. Les types d'installation représentent les options d'installation de plus haut niveau dont la sélection est facile pour l'utilisateur (celui-ci pouvant sélectionner un seul type d'installation). Les types d'installation sont généralement désignés par des options telles que « Standard », « Minimum », « Client uniquement », « Client et serveur », etc. Le panneau (ou la console) **Sélectionnez un type d'installation** permet aux utilisateurs de sélectionner le type d'installation requis. Rappel : les types d'installation sont composés de fonctions.

Fonctions

Les fonctions sont proposées à l'utilisateur seulement lorsque celui décide de personnaliser le type d'installation qu'il a sélectionné auparavant. Pour les développeurs, les fonctions sont intéressantes surtout s'ils souhaitent que leurs utilisateurs puissent personnaliser leur installation.

Le nombre de fonctions n'est pas limité, mais le produit doit en comporter au moins une. Par ailleurs, les fonctions proposées peuvent être structurées en hiérarchie. Pour proposer aux utilisateurs un exemple de hiérarchie de fonctions, les développeurs peuvent inclure une sous-fonction « Documentation » et une sous-fonction « Exemples » sous la fonction « Aide » principale.

Chaque type d'installation est composé de plusieurs fonctions et chaque fonction peut appartenir à un ou plusieurs types d'installation. Pour sélectionner les fonctions requises, les utilisateurs activent les options correspondantes du panneau (ou de la console) **Sélectionnez un type d'installation.**

Des fichiers peuvent être affectés directement à des fonctions et vous pouvez affecter des composants aux fonctions ; en fait, si vous affectez des fichiers directement à des fonctions, InstallAnywhere crée les composants à votre place.

Si nécessaire, les utilisateurs peuvent désinstaller certaines fonctions.

Composants

Le composant est le plus petit élément d'installation manipulé par l'installateur. Du point de vue des développeurs, les composants sont les éléments de base des applications ou des fonctions. Les utilisateurs ne voient jamais les composants et n'interviennent jamais sur ceux-ci.

Certains composants peuvent être partagés par plusieurs installateurs, par plusieurs versions d'un produit ou par plusieurs produits (par exemple, deux logiciels d'une suite pourraient partager plusieurs composants).

Les composants sont affectés d'un numéro de version et d'un ID unique, ce qui permet de les rechercher et de vérifier que la version la plus récente a été installée dans un emplacement spécifique.

Chaque fonction est composée de plusieurs composants et chaque composant peut appartenir à une ou plusieurs fonctions. Les composants sont à base de fichiers et d'actions. Bien qu'il soit possible d'affecter directement des fichiers et des actions aux composants, il est également possible d'affecter des fichiers et des actions directement aux fonctions et de laisser InstallAnywhere créer automatiquement les composants correspondants.

Un fichier ou une action peut appartenir à un seul composant. Tous les installateurs doivent comporter au moins un composant, mais le nombre de composants d'un installateur n'est pas limité.

Interface graphique (GUI) de l'installateur

La plupart des caractéristiques de l'interface graphique (GUI) d'un installateur InstallAnywhere peuvent être modifiées. Les développeurs peuvent modifier les chaînes de texte et les éléments graphiques de l'installateur (écran d'accueil, écrans de l'installateur, additions des panneaux, images de fond, billboards, etc.) ; ils peuvent même insérer des fichiers GIF animés !

Localisation

La plupart des chaînes de texte d'un projet InstallAnywhere peuvent être localisées (la traduction du texte des écrans et des boîtes de dialogue InstallAnywhere est livrée avec le produit). La version « Enterprise Edition » supporte la création d'installateurs en 29 langues ; la version « Standard Edition » supporte neuf langues.

Pour modifier les chaînes de texte locale par locale, il suffit d'ouvrir les fichiers qui contiennent ces chaînes. Ces fichiers sont générés à chaque construction

de l'installateur et stockés dans le dossier « Project locales », qui réside à côté du dossier de sortie de construction. Le nom des fichiers de locale dépend du code standard de celle-ci ; par exemple, le fichier de locale par défaut pour l'anglais (code de locale : en) est affecté du nom custom_en.

Dans ces fichiers de locale, les chaînes de texte sont regroupées en fonction du nom de l'action à laquelle elles appartiennent. Les développeurs peuvent modifier ces chaînes de texte : lors de la prochaine construction de l'installateur, le nouveau texte localisé sera affiché dans l'action correspondante.

Paramètres d'affichage

De nombreuses options InstallAnywhere permettent de modifier l'apparence et le comportement (« look and feel ») de l'installateur.

Écran d'accueil

Au lancement d'un installateur InstallAnywhere, un « écran d'accueil » s'affiche pendant quelques secondes (nécessaires à l'installateur pour préparer l'Assistant de projet). Pour les utilisateurs, l'écran d'accueil est le point de départ de votre produit : vous devez donc vous efforcer de « créer une bonne impression ». L'écran d'accueil apparaît également sur la page HTML générée par l'applet d'installation Web. L'écran d'accueil peut être un fichier GIF, PNG ou JPEG – de préférence avec une taille de 470 x 265 pixels.

Additions aux panneaux de l'installateur à GUI

L'option **Additions aux panneaux de l'installateur à GUI** permet aux développeurs d'afficher des éléments complémentaires (par exemple, les étapes de l'exécution ou un élément graphique) dans la partie gauche du panneau de l'installateur.

Image de fond

Pour agrémenter la présentation des panneaux d'un installateur, vous pouvez afficher un élément graphique de votre choix comme image de fond. Remarque : les images de fond sont supportées uniquement par les installateurs Swing à GUI.

Billboards

Un « billboard » est un élément graphique qui s'affiche quelques instants pendant l'installation des fichiers. Les billboards défilent pendant l'installation des fichiers ; en général, ils affichent un message marketing, décrivent certaines fonctionnalités du produit, invitent l'utilisateur à s'enregistrer, etc. La durée d'affichage est la même pour tous les billboards ajoutés, en fonction des actions prévues pour l'installation. Par conséquent, si une installation porte sur un nombre limité de fichiers mais inclut un grand nombre de billboards, chaque billboard s'affichera pendant un temps très court.

Les billboards peuvent être des fichiers GIF, PNG ou JPEG, leur taille d'affichage est limitée à 587 x 312 pixels et ils supportent les fichiers GIF animés.

Actions

L'architecture d'InstallAnywhere est à base d'actions « extensibles » qui permettent aux développeurs d'exécuter les opérations requises pendant l'installation. Ces actions peuvent être très simples (installer des fichiers ou des dossiers) ou très complexes (créer ou modifier des fichiers texte, exécuter du code personnalisé pendant l'installation ou extraire le contenu d'un fichier compressé).

Certaines actions sollicitent une entrée utilisateur (par exemple, les actions par panneaux et les actions par consoles), d'autres s'exécutent en tâche de fond sans faire intervenir celui-ci (par exemple, les actions générales et les actions d'installation). Pour solliciter l'entrée utilisateur, les actions par panneaux affichent un élément graphique ; les actions par consoles affichent une ligne de commande.

Actions générales

La plupart des actions générales s'exécutent en mode transparent, soit sans solliciter d'entrée utilisateur.

Actions par panneaux

Les « actions par panneaux » sollicitent une entrée utilisateur et s'affichent dans l'Assistant d'installation.

Actions par consoles

Les « actions par consoles » sont des lignes de commande qui sollicitent une entrée utilisateur et qui sont utilisées en cas d'installation par ligne de commande.

Actions de code personnalisé (plug-ins)

Les développeurs peuvent intégrer du code personnalisé dans le module Advanced Designer d'InstallAnywhere ; ce code est ensuite disponible sous la forme d'un ou plusieurs plug-ins.

Règles

Les règles InstallAnywhere peuvent être appliquées à toute action de l'installateur InstallAnywhere et aux unités structurelles du projet (types d'installation, fonctions et composants).

Pour contrôler les comportements d'un installateur, InstallAnywhere utilise généralement des règles booléennes à base de variables. La logique de ces règles permet aux développeurs de créer des systèmes simples ou complexes qui déterminent les actions à déclencher. Les règles peuvent être structurées en fonction des entrées de l'utilisateur ou de conditions déterminées par l'installateur.

Désinstallateur

InstallAnywhere crée automatiquement un désinstallateur pour chaque projet (ce désinstallateur peut ensuite être supprimé manuellement). En général, le désinstallateur InstallAnywhere supprime l'ensemble des fichiers et des actions impliqués dans l'installation correspondante, mais il est possible de configurer ce comportement.

De structure semblable à celle de l'installateur, le désinstallateur se présente sous la forme de panneaux, de consoles et d'actions. Il consigne les détails des actions exécutées par l'installateur et des actions exécutées lors de l'installation.

Le désinstallateur exécute d'abord les panneaux, les actions et les consoles de pré-désinstallation ; il supprime ensuite les fichiers qui ont été installés et désinstalle les actions présentes dans l'arborescence des fichiers ; enfin, il exécute les opérations de post-désinstallation.

LaunchAnywhere

LaunchAnywhere est une technologie de lancement d'application développée par Zero G. Exécutable natif utilisé pour lancer une application Java, LaunchAnywhere crée des icônes double-cliquables sous Windows et sous Mac OS X (sur les plates-formes UNIX/Linux, l'application créée est à base de ligne de commande).

Le lanceur LaunchAnywhere détecte une VM Java (JVM) correcte (intégrée à l'application à installer ou présente dans le système cible), et définit les options et paramètres Java (par exemple, la taille de la mémoire/heap) correspondant aux spécifications des développeurs. LaunchAnywhere définit le classpath, redirige les données stdout et stderr, transmet les propriétés système, les variables d'environnement et les paramètres de ligne de commande et lance l'application Java. Pour les applications à interface graphique (GUI), LaunchAnywhere masque la fenêtre de console (comportement par défaut) ; pour les applications à interface texte, LaunchAnywhere peut être configuré de manière à afficher la console. Les paramètres LaunchAnywhere sont configurés dans InstallAnywhere et sont définis automatiquement lorsque le produit installe l'application.

PowerUpdate

PowerUpdate est la solution proposée par Zero G pour le déploiement et la mise à jour des logiciels multiplate-forme ; elle est disponible pour tous les types de logiciel et sur toute plate-forme client ou serveur. Avec PowerUpdate, les utilisateurs sont assurés de toujours disposer de la version la plus récente de l'application déployée : il n'est plus nécessaire de les contacter pour les informer de la disponibilité d'une nouvelle version/release, d'une mise à jour ou d'un correctif (patch). En outre, cette solution réduit le nombre des appels client vers vos services de support technique.

Pour ajouter les fonctions PowerUpdate à une application, il suffit d'exécuter l'action **Installer un client PowerUpdate**.

InstallAnywhere se met à jour automatiquement à l'aide de PowerUpdate. Si une mise à jour est disponible, un message permet aux développeurs de télécharger celle-ci.

Pour plus de détails sur PowerUpdate, visitez les pages suivantes : <http://www.zerog.com/goto/powerupdate>.

Magic Folders

Pour définir les emplacements d'installation, InstallAnywhere utilise des « Magic Folders ». InstallAnywhere peut installer l'application dans l'un des Magic Folders disponibles (y compris dans l'un de ses sous-dossiers). Un Magic Folder correspond à un emplacement spécifique, par exemple le dossier d'installation sélectionné par l'utilisateur, le bureau Windows ou l'emplacement des fichiers de bibliothèque. Lors de l'installation, l'installateur détecte le système d'exploitation utilisé par l'ordinateur et il associe les Magic Folders aux chemins absolus appropriés. Certains Magic

Folders sont spécifiques à la plate-forme cible, d'autres sont prédéfinis par InstallAnywhere. Si la plate-forme cible est supportée, l'installation peut être exécutée dans la plupart des emplacements standard. Par ailleurs, InstallAnywhere propose des Magic Folders qui peuvent être adaptés aux besoins des développeurs.

Remarque : les Magic Folders disponibles varient avec chaque version d'InstallAnywhere.

SpeedFolders

Les « SpeedFolders » augmentent considérablement la vitesse d'installation et l'efficacité de la mémoire. Les SpeedFolders sont des dossiers « virtuels » ; ils constituent un simple pointeur vers un dossier, mais ne contiennent pas vraiment de sous-dossiers et de fichiers comme les dossiers standard. Toutefois, contrairement aux dossiers standard, les SpeedFolders et le contenu qu'ils représentent sont traités comme une seule action (dans un dossier standard, chaque sous-dossier/fichier est considéré comme un élément distinct). Ce traitement global réduit les besoins en mémoire et accélère l'installation.

Les SpeedFolders sont parfaitement adaptés aux environnements de construction automatisés. Le contenu d'un SpeedFolder est déterminé au moment de la construction. Au moment de la construction de l'installateur, le contenu du dossier présent sur le système de construction (à l'exception des objets soumis à un filtre) est ajouté à l'installateur en mode récursif. Les SpeedFolders sont utilisés pour demander qu'un des dossiers standard présents sur le système de développement soit automatiquement mis à jour et inclus dans l'installateur au moment de la construction du projet. Avec les dossiers standard (autres que les SpeedFolders), les développeurs doivent ajouter ou supprimer « manuellement » les fichiers présents ou absents par rapport à la dernière construction de l'installateur – sinon, une erreur se produit.

Les SpeedFolders sont équipés de filtres qui permettent l'inclusion et l'exclusion des fichiers qui répondent à certains critères de nom. Les sous-dossiers et fichiers référencés par un SpeedFolder ne peuvent pas être affectés à des composants différents. Par ailleurs, les SpeedFolders ne peuvent pas être convertis directement en dossiers standard, ni les dossiers standard en SpeedFolders ; pour convertir un dossier d'un type à un autre, vous devez d'abord supprimer le dossier à convertir, puis l'écraser par l'autre dossier.

Variables InstallAnywhere

Pendant l'installation, InstallAnywhere assure le suivi des valeurs dynamiques à l'aide de variables. La plupart des valeurs dynamiques d'InstallAnywhere (par exemple, le chemin associé à un Magic Folder) sont représentées par une variable InstallAnywhere. Les variables InstallAnywhere peuvent être utilisées ou modifiées de manière à affecter les fonctionnalités ou le comportement d'un installateur.

Les variables InstallAnywhere constituent l'un des éléments essentiels des installateurs InstallAnywhere. Elles permettent aux développeurs de contrôler les flux d'information et le déroulement de l'installation, d'enregistrer les informations fournies par le système et saisies par les utilisateurs, puis de définir des règles d'exécution en fonction de ces informations. Les développeurs peuvent également diriger ces informations vers les fichiers de configuration, ou vers des ressources qui doivent être utilisées par l'application.

Sous-projets d'installateur – Modules de fusion

Les « modules de fusion » sont essentiellement des sous-projets d'installateur qui peuvent être créés de manière indépendante, puis fusionnés. En fait, un module de fusion est un ensemble réutilisable de fonctionnalités d'installation avec ses propres fonctions, composants, panneaux, actions et fichiers. Un module de fusion ne peut pas être installé s'il ne fait pas partie d'un projet InstallAnywhere ; en effet, les modules de fusion permettent d'inclure les fonctionnalités d'un installateur donné dans un autre installateur.

Modèles – Présentation générale

Le « modèle » est le point de départ de tout projet d'installateur. Un modèle peut être un projet vide, ou contenir les mêmes éléments qu'un projet standard (contrat de licence, éléments graphiques, billboards et même fichiers).

Fichier de projet

InstallAnywhere stocke chaque projet dans son propre fichier XML. Ces fichiers de projet XML peuvent être récupérés dans un système de gestion de fichiers source (puis replacés dans ce système) et être modifiés à l'aide d'un éditeur de texte ou d'un éditeur XML. Pour plus de souplesse, les XSL Transforms permettent de modifier les chemins et d'autres attributs des fichiers de projet. Des outils XML et XSL permettant de modifier le fichier XML de projet sont disponibles dans le dossier d'application d'InstallAnywhere, sous resource/extras/XML Project File Tools.

Didacticiels – À la découverte d'InstallAnywhere

InstallAnywhere propose deux solutions pour la création d'installateurs :

(1) L'Assistant de projet facilite le processus, mais en effectuant des choix à la place des développeurs. (2) Avec le module Advanced Designer, les développeurs bénéficient d'une plus grande souplesse pour leurs projets d'installateur.

Si la préférence par défaut n'a pas été changée via **Édition | Préférences**, InstallAnywhere affiche le premier écran de l'Assistant de projet. Pour accéder à un projet InstallAnywhere existant, ouvrez son fichier, puis sélectionnez **Advanced Designer** ; par contre, si vous sélectionnez **Suivant**, l'Assistant affiche ses autres écrans.

La séquence générale de développement d'un projet InstallAnywhere est la suivante :

1. Création du projet
2. Définition des informations
3. Actions pré-installation
4. Tâches de fichiers
5. Actions post-installation
6. Définition des options de l'interface d'installation
7. Configuration du désinstallateur de projet
8. Construction du projet
9. Test du projet

Le didacticiel d'introduction permet de construire un installateur en suivant les conseils de l'Assistant de projet (toutefois, cet Assistant ne permet pas de configurer des actions de pré-installation ou de post-installation).

Didacticiel d'introduction – Construire un installateur

En suivant les conseils de l'Assistant de projet, il est possible de construire un installateur élémentaire – en six étapes et en quelques minutes. Le présent didacticiel explique comment construire un installateur pour application Java appelé « OfficeSuite pour Java » (cet installateur-exemple est inclus dans le dossier InstallAnywhere).

1. Créer un projet
 - a. Lancez InstallAnywhere.

Dans le premier écran, vérifiez que l'option **Créer un projet** est sélectionnée.
 - b. Pour enregistrer le projet en lui donnant un nom de votre choix, cliquez sur **Enregistrer sous**.

La boîte de dialogue **Enregistrer nouveau projet sous** apparaît (par défaut, le projet est appelé MonProduit, mais vous pouvez utiliser le nom de votre choix).

- c. Cliquez sur **Enregistrer** pour confirmer le nom du projet et refermer la boîte de dialogue.
- d. Cliquez sur **Suivant** pour passer à la prochaine étape de l'Assistant de projet.

Définir les détails permet de fournir des informations de base sur l'installateur : nom du produit tel qu'il sera affiché dans l'installateur, nom de l'installateur à construire, nom du dossier de destination (dossier d'installation) et nom de l'application.

- e. Tapez ces informations dans les champs correspondants.

Dans le cadre de ce didacticiel, utilisez les valeurs suivantes :

Nom du produit : OfficeSuite

Nom du dossier d'installation : OfficeSuite

Raccourci de l'application (de type Raccourci) : OfficeSuite

- f. Cliquez sur **Suivant** pour passer à l'étape suivante de l'Assistant de projet.

2. Tâches de fichiers

- a. Ajoutez les fichiers requis.
 - i. Cliquez sur **Ajouter des fichiers**. La boîte de dialogue **Ajouter des fichiers dans le projet** apparaît. Retrouvez le sous-dossier OfficeSuiteSourceFiles (dossier d'installation d'InstallAnywhere).
 - ii. Cliquez sur **Ajouter tout** pour ajouter les dossiers « ImagesAndDocs » et « OfficeSuite2000 », qui résident dans le dossier « OfficeSuiteSourceFiles ».

Ces fichiers apparaissent maintenant dans la liste **Fichiers à ajouter**.

- iii. Cliquez sur **Terminé**.

Les fichiers sélectionnés apparaissent dans la hiérarchie de fichiers/dossiers.

- iv. Cliquez sur **Suivant** pour passer à l'étape suivante de l'Assistant de projet.

Remarque : si vous n'êtes pas en train d'installer une application Java, sautez cette étape en cliquant sur **Suivant** sans spécifier de classe main. La boîte de dialogue **Sélectionnez une classe main** apparaît : cliquez sur **Non** pour passer à l'étape suivante.

- b. Sélectionnez une classe main

Sélectionner une classe main sélectionne la classe initiale de l'application. Dans cet écran, les développeurs peuvent également spécifier les icônes de leur choix (au format GIF) pour le fichier exécutable LaunchAnywhere.

- i. Cliquez sur **Retrouver automatiquement les classes main** (au bas de l'écran).
- ii. Sélectionnez la classe main.

- iii. Pour associer une icône de votre choix au fichier exécutable LaunchAnywhere, cliquez sur **Changer** et sélectionnez un fichier GIF de 32x32 ou de 16x16 pixels (remarque : les fichiers ICO ne sont pas supportés). Dans le cadre de ce didacticiel, ouvrez le dossier Image and Docs et sélectionnez le fichier d'icône OfficeIcon.gif.
- iv. Cliquez sur **OK** pour confirmer cette sélection et refermer la boîte de dialogue.
L'icône s'affiche sur l'écran principal.
- v. Cliquez sur **Suivant** pour passer à l'étape suivante de l'Assistant de projet.

Remarque : si vous n'êtes pas en train d'installer une application Java, sautez cette étape en cliquant sur **Suivant** sans spécifier de classe main. La boîte de dialogue **Sélectionnez une classe main** apparaît : cliquez sur **Non** pour passer à l'étape suivante.

c. Définissez le classpath

Définir le classpath configure automatiquement le classpath d'une application Java.

- i. Cliquez sur Définir le classpath automatiquement.

InstallAnywhere sélectionne les fichiers qui doivent être déclarés dans le classpath : les lettres « CP » (classpath) apparaissent en bleu à la partie inférieure des icônes des dossiers contenant ces fichiers.

- ii. Cliquez sur **Suivant** pour passer à l'étape suivante de l'Assistant de projet.

3. Construire un installateur

Les sept premières options de l'écran **Construire un installateur – de Mac OS X à UNIX (tous)** correspondent aux installateurs qui peuvent être lancés par double-clic sur leur plate-forme respective. La dernière option, **Autres plates-formes prêtes pour Java**, correspond à un installateur « Pure Java » qui peut être appelé à partir de la ligne de commande de toute plate-forme supportant Java. Les développeurs peuvent également construire des installateurs avec VM Java intégrée (les installateurs construits sans VM ont une taille plus réduite et leur téléchargement est plus rapide). L'option d'installateur Web permet aux utilisateurs de sélectionner l'installateur approprié pour leur système.

- a. Sélectionnez la ou les plates-formes de destination et cliquez sur Construire.

Le dossier de l'installateur occupera le même emplacement que le fichier de projet (cet emplacement ne peut pas être modifié).

4. Tester le projet

- a. Pour tester l'installateur que vous venez de construire, cliquez sur Tester.
- b. Après avoir déployé l'installateur-exemple, procédez comme suit :
 - o Windows – Affichez le groupe de programmes OfficeSuite et sélectionnez **OfficeSuite**.

- o UNIX – Placez-vous (cd) au niveau du répertoire d'installation du programme et entrez **OfficeSuite**.
- o Mac OS X – Double-cliquez sur l'icône OfficeSuite du bureau.
- c. Après avoir lancé OfficeSuite pour Java, sélectionnez **File | Exit (Fichier | Quitter)**.

Il est possible de publier le dossier de l'installateur sur un serveur Web et d'installer le logiciel sur une autre plate-forme.

Conseil – (Windows uniquement) – Pour afficher la sortie de débogage, maintenez enfoncée la touche <Ctrl> pendant que l'installateur démarre.

- d. Exécutez l'installateur.

Si la construction est destinée à une plate-forme autre que la plate-forme de développement de l'installateur, transférez l'installateur sur la plate-forme cible et lancez-le manuellement. Par défaut, les installateurs sont stockés dans le sous-dossier Build_Output, qui réside dans le même dossier que le fichier de projet .iap.xml. Build_Output contient lui-même le sous-dossier Web_Installers et/ou (selon configuration) le sous-dossier CDRom_installers. Dans chacun de ces sous-dossiers, sélectionnez la plate-forme à tester. Pour un installateur CD, transférez le contenu intégral du dossier CDRom_installers.

Didacticiel avancé – Utiliser Advanced Designer

Dans ce didacticiel, vous allez construire le même installateur OfficeSuite que dans le didacticiel précédent, mais avec l'aide du module Advanced Designer intégré à InstallAnywhere. Advanced Designer permet un plus grand contrôle que l'Assistant de projet pour la configuration des options InstallAnywhere.

Dans la première fenêtre de l'Assistant de projet, sélectionnez **Advanced Designer**.

1. Créer un projet

- a. Lancez InstallAnywhere.

Dans le premier écran, vérifiez que l'option **Créer un projet** est sélectionnée.

- b. Sélectionnez le modèle « Basic Project ».

(Ce modèle devrait être sélectionné par défaut.)

- c. Pour enregistrer le projet en lui donnant un nom de votre choix, cliquez sur **Enregistrer sous**.

La boîte de dialogue **Enregistrer nouveau projet sous** apparaît. InstallAnywhere utilisera ce nom comme nom de produit dans le projet d'installateur.

- d. Cliquez sur Advanced Designer.

Le fichier de projet que vous venez de créer s'ouvre dans Advanced Designer. Advanced Designer lance la tâche **Projet | Infos**. Cette tâche permet de définir les options de base de l'installateur : nom du produit, titre et nom de l'installateur. Le nom de l'installateur est celui du fichier exécutable créé par InstallAnywhere. Cet onglet permet également de spécifier l'emplacement de construction de l'installateur et les paramètres affectant la génération des journaux d'installation.

- e. Renseignez les champs Titre de l'installateur et Nom du produit.

Dans le cadre de ce didacticiel, nous allons ignorer les tâches **Interface d'installation**, **Structure** et **Fichiers**.

2. Actions pré-installation

- a. Sélectionnez la tâche **Pré-installation**.

La tâche **Pré-installation** permet de définir les panneaux et les actions qui interviennent avant l'installation des fichiers. Par défaut, les nouveaux projets InstallAnywhere contiennent les panneaux suivants :

Introduction – Permet aux développeurs de présenter le produit ou de décrire le processus d'installation.

Sélectionner le dossier d'installation – Permet à l'utilisateur de désigner l'emplacement d'installation du produit.

Sélectionner le dossier d'alias/lien/raccourcis – Permet à l'utilisateur de désigner l'emplacement dans lequel doivent être stockés les alias Mac OS, les raccourcis Windows et les liens symboliques UNIX (utilisés comme raccourcis).

Récapitulatif de pré-installation – Présente à l'utilisateur un résumé des paramètres d'installation (avant installation des fichiers).

Les actions de la tâche **Pré-installation** sont exécutées dans l'ordre défini par la liste associée à celle-ci. Dans un projet par défaut, le panneau **Introduction** est suivi du panneau **Sélection du dossier d'installation**, puis du panneau **Sélectionner le dossier des alias/liens/raccourcis**, et ainsi de suite. L'ordre des panneaux et des actions peut être modifié à l'aide des boutons fléchés affichés dans la partie droite de l'écran Advanced Designer.

Pour modifier le contenu ou/et le comportement d'un panneau, il suffit de le sélectionner : les options affichées dans la partie inférieure de l'écran Advanced Designer changent en fonction du panneau sélectionné (cette zone d'options constitue le « personnaliseur » du panneau sélectionné ; elle est disponible pour chaque action et pour chaque panneau de l'installateur).

3. Définir les tâches d'installation

- a. Sélectionnez la tâche **Fichiers** (partie gauche d'Advanced Designer).

La tâche **Fichiers** définit les fichiers à installer, le dossier d'installation de ces fichiers et l'ordre des tâches qui doivent s'exécuter pendant l'installation des fichiers.

Par défaut, la tâche **Fichiers** est associée à un dossier appelé « Uninstall \$PRODUCT_NAME\$ », qui contient les actions de désinstallation InstallAnywhere et une action de commentaire fournissant des instructions sur le désinstallateur.

Les actions de la liste associée à la tâche **Fichiers** (y compris l'installation des fichiers) s'exécutent dans l'ordre dans lequel elles apparaissent dans cette liste – l'action citée « en haut » de la liste s'exécutant en premier.

Conseil – Advanced Designer supporte les fonctions de Glisser-déplacer dans la plupart des onglets et des tâches. Par exemple, dans la tâche **Installation**, les actions et les fichiers peuvent être déplacés de cette manière (un trait de soulignement foncé apparaît pour indiquer l'emplacement de destination de l'action ou du fichier).

Conseil – Bien que la structure des dossiers puisse être modifiée, il est conseillé de créer le désinstallateur dans son emplacement par défaut. Par ailleurs, il est généralement préférable de prévoir la création du désinstallateur en premier.

Le didacticiel avancé répétant les tâches exécutées par l'Assistant de projet pour l'installateur « OfficeSuite », les mêmes fichiers sont ajoutés.

- b. Ajoutez les fichiers requis.
 - i. Utilisez l'explorateur de fichiers pour parcourir le sous-dossier des fichiers source OfficeSuite (dossier d'installation d'InstallAnywhere). Ce dossier peut également être ajouté dans la tâche **Fichiers** par Glisser-déplacer.
 - ii. Ajoutez le dossier OfficeSuite et son contenu.

Une fois ajoutés, les fichiers apparaissent dans l'arborescence des fichiers d'installation d'Advanced Designer.

Dans la tâche **Installation** d'Advanced Designer, vous pouvez développer et réduire l'arborescence des fichiers en cliquant (respectivement) sur le bouton **+** ou sur le bouton **-** affichés au raccordement de chaque branche. Pour faire remonter ou redescendre un fichier d'un sous-dossier à un autre de l'arborescence, cliquez sur les boutons fléchés Droite/Gauche/Haut/Bas qui sont regroupés dans la partie droite de l'écran de la tâche **Fichiers**.

4. Ajouter un exécutable LaunchAnywhere dans la tâche **Fichiers**

- a. Cliquez sur Ajouter un lanceur.

Un « exécutable LaunchAnywhere » est un fichier natif unique (extension LAX) créé par InstallAnywhere pour lancer une application Java. Contrairement à l'Assistant de projet, qui demande à l'utilisateur de sélectionner une classe main et qui crée un seul lanceur, Advanced Designer permet aux développeurs d'ajouter autant de lanceurs qu'ils le souhaitent.

Pour ajouter un lanceur LaunchAnywhere dans un fichier de projet InstallAnywhere, vous devez utiliser l'option **Créer une application LaunchAnywhere pour Java** : cette option peut être sélectionnée à partir de la palette **Ajouter une action**, ou ajoutée en cliquant sur le bouton **Ajouter un lanceur** affiché au milieu de la barre de commande d'Advanced Designer.

- i. Sélectionnez **Dossier d'installation utilisateur** dans Advanced Designer, puis cliquez sur **Ajouter un lanceur**.
- ii. Cliquez sur **OK**.

Lorsque InstallAnywhere ajoute un lanceur, il explore les fichiers ajoutés (y compris ceux qui se trouvent dans les JAR ou dans les ZIP) pour retrouver les fichiers de classe contenant des méthodes main.

- iii. Choisissez **com.acme.OfficeSuite** comme classe main de l'application.

OfficeSuite étant un projet élémentaire, la seule classe affichée est `com.acme.OfficeSuite`.

- b. Cliquez sur **OK** pour continuer.

Remarque : le bouton **Ajouter un lanceur** a deux effets : ajouter le lanceur dans la structure des fichiers et créer un raccourci, un lien ou un alias dans le Magic Folder de destination des raccourcis. Cet emplacement est variable, et il est désigné par le panneau **Sélectionner le dossier des alias/liens/raccourcis** de la phase Pré-installation.

- c. Personnalisez l'icône du lanceur.

Vous pouvez maintenant personnaliser le raccourci (icône) du lanceur. Sélectionnez le lanceur. Le contenu du personnaliseur (partie inférieure de l'écran Advanced Designer) change en fonction des options de l'action **Créer une application LaunchAnywhere pour Java**.

Les boutons regroupés sous le champ des arguments permettent de modifier l'icône associée à ce lanceur (l'icône par défaut affiche une tasse de café et une fusée sur fond bleu canard).

- i. Pour valider l'icône personnalisée, cliquez sur **Changer**.
- ii. Dans la boîte de dialogue **Sélectionnez une icône**, cliquez sur **Sélectionner un fichier**.
- iii. Sélectionnez le fichier GIF ou JPG à utiliser comme icône. Dans le cadre de ce didacticiel, utilisez l'icône `OfficeSuiteIcon`, présente dans le sous-dossier `Images and Docs` du dossier `OfficeSuiteSourceFiles`.

Remarque : InstallAnywhere ne supporte pas les fichiers GIF à entrelacement (affichage progressif de l'icône). Le processus de conversion ne peut pas traiter ces fichiers et leur utilisation se traduit généralement par une icône vide.

Remarque : Pour Mac OS X, spécifiez un fichier ICNS (créé par `iconbuilder`, qui fait partie des outils de développement « Mac OS X Developer Tools »).

- d. Définissez le classpath pour InstallAnywhere.

(InstallAnywhere dispose d'un classpath général qui est utilisé par défaut pour créer les lanceurs d'applications Java.)

- i. Dans Advanced Designer, cliquez sur **Définir le classpath**.

Les lettres « CP » (classpath) apparaissent en bleu sur les icônes des dossiers et des archives déclarés dans le classpath.

- ii. Activez l'onglet **Projet | Java** (partie gauche de la fenêtre Advanced Designer) pour afficher le classpath qui vient d'être défini par l'action **Définir le classpath**.

OfficeSuite étant un produit élémentaire, nous disposons seulement du dossier `OfficeSuite2000`, qui contient divers fichiers de classe. Si notre projet-exemple incluait des fichiers JAR ou ZIP contenant des

classes, ceux-ci auraient également été ajoutés. Si un fichier est déclaré par erreur dans le classpath, il peut en être retiré à ce stade – ou plus tard en le sélectionnant dans l’arborescence d’installation et en décochant la case **Dans classpath** affichée dans le personnalisateur de ce fichier.

5. Actions post-installation

La tâche Post-installation indique les actions et les panneaux qui doivent intervenir après l’installation des fichiers dans une liste qui est ordonnée par ordre d’exécution, les actions situées « en haut » de cette liste ayant lieu en premier.

Par défaut, InstallAnywhere a ajouté deux actions au projet InstallAnywhere, soit :

Panneau : Installation terminée

Ce panneau s’affiche lorsque l’installation s’est terminée avec succès. Cette action est conditionnée par l’état de la variable \$INSTALL_SUCCESS\$: son panneau s’affiche seulement si \$INSTALL_SUCCESS\$ ne contient pas d’erreur.

Relancer Windows

Cette action relance Windows si l’installateur a déterminé que c’était nécessaire.

Les installations InstallAnywhere sont contrôlées essentiellement par les règles InstallAnywhere. Pour consulter un exemple de règle InstallAnywhere, sélectionnez l’action **Relancer Windows** dans le projet-exemple OfficeSuite ; dans le personnalisateur (partie inférieure de l’écran), activez l’onglet **Règles**.

Le personnalisateur de règles InstallAnywhere apparaît dans la partie inférieure d’Advanced Designer. Les règles qui contrôlent l’action **Relancer Windows** sont simples : elles se contentent de comparer les variables InstallAnywhere. Les règles InstallAnywhere sont booléennes : elles autorisent l’installation, l’affichage ou l’exécution du fichier, du panneau ou de l’action seulement si elles ont la valeur True.

- a. Cliquez sur **Ajouter une action** pour afficher la palette des actions.

Les onglets affichés dans la palette des actions varient en fonction de la tâche qui était active au moment où la palette a été appelée.

- b. Sélectionnez l’option **Exécuter le fichier cible** (sous l’onglet **Général**).

L’action **Exécuter le fichier cible** portant sur les fichiers inclus dans l’installation, elle est disponible uniquement lors des phases Installation et Post-installation ; elle n’est pas disponible lors de la phase Pré-installation dans la mesure où il n’est pas possible d’exécuter des fichiers qui... ne sont pas encore installés !

- c. Pour ajouter l’action, cliquez sur **Ajouter** (ou double-cliquez sur son nom).

La palette reste affichée pour permettre l’ajout d’autres actions.

- d. Dans le personnalisateur (partie inférieure de l’écran), tapez le nom de votre choix pour cette action.

Le nom affecté à l'action facilitera son identification dans l'arborescence graphique.

- e. Pour désigner la cible, cliquez sur **Sélectionner une cible**.

La boîte de dialogue **Sélectionnez une action** présente l'arborescence d'installation spécifiée dans la tâche **Fichiers**. À ce stade, les fichiers peuvent être exécutés. Pour exécuter l'application-exemple OfficeSuite que vous venez d'installer, sélectionnez son lanceur.

Attention – Vous devez sélectionner le lanceur OfficeSuite et non son raccourci (en général, ils partagent la même icône). N'oubliez pas que les raccourcis – surtout sous Windows et Mac OS – sont de simples pointeurs et non des exécutables (InstallAnywhere ne peut pas exécuter une application ou un fichier à partir de son raccourci).

Options du personnalisateur

Le champ Ligne de commande permet de modifier l'instruction d'exécution de fichier, par exemple en lui ajoutant l'adresse d'un module de gestion (handler) ou des arguments.

Remarque : il est conseillé de ne pas modifier ni supprimer l'entrée \$EXECUTE_FILE_TARGET\$, qui désigne le fichier à exécuter. Pour spécifier un module de gestion/handler, spécifiez un chemin exécutable en première position (prepend) ; pour ajouter un argument, spécifiez un chemin de fichier en dernière position (append). Ces chemins doivent impérativement être absolus, mais ils peuvent inclure des variables InstallAnywhere.

Cette action peut être modifiée à l'aide des champs d'option. L'une des options permet de marquer une pause dans l'installation jusqu'à ce que le processus en cours soit terminé, ce qui est particulièrement utile si des étapes ultérieures de l'installation dépendent des résultats de ce processus. Une autre option permet aux développeurs de définir une jauge de progression non précisée (avec message). Cette tâche peut être utilisée si l'exécution prend un certain temps (par exemple, installation d'un autre produit ou configuration d'une base de données ou d'une autre application).

L'option **Afficher le panneau « Veuillez patienter »** présente à l'utilisateur ce message de temporisation pendant l'exécution.

L'option **Supprimer la première fenêtre** (plates-formes Windows uniquement) permet aux développeurs d'empêcher l'affichage de la fenêtre cmd.exe, ce qui est particulièrement utile lors de l'exécution de fichiers batch ou d'exécutables à partir de la ligne de commande.

Remarque : si le panneau d'exécution d'action a été ajouté dans un élément d'interface autre que la partie inférieure de la tâche Post-installation, déplacez-le maintenant (utilisez la flèche Haut ou Bas, ou faites glisser l'action au bas de la liste des tâches).

6. Construire l'installateur

La tâche **Construction du projet** permet de définir les options à utiliser pour le ou les installateurs à construire : sélectionner les plates-formes cible,

configurer les options des VM intégrées, optimiser la plate-forme et définir le type du ou des installateurs.

Conseil – Pour simplifier les tests initiaux, il est conseillé de limiter la construction à la plate-forme de développement. En effet, toute plate-forme supplémentaire augmente le temps de construction par suite des nombreux cycles modification-exécution-construction. En outre, une construction rapide facilite les opérations ultérieures de développement.

Dans l'onglet **Cibles de la construction** de la tâche **Construction du projet**, sélectionnez la ou les plates-formes cible. Si vous sélectionnez l'option **Avec VM**, une VM sera intégrée à l'installateur. Remarque : l'option **Avec VM** est limitée aux plates-formes qui disposent d'un Pack VM. Les Packs VM doivent être stockés dans le dossier [InstallAnywhere]/resource/installer_vms. Ils deviennent disponibles seulement après redémarrage d'InstallAnywhere.

La tâche **Construction du projet** propose également les onglets **Distribution** et **Journal de construction**.

L'onglet **Distribution** permet de définir les options pour les types d'installateur à construire et les options d'optimisation pour chaque installateur. L'installateur OfficeSuite que vous construisez dans le cadre de ce didacticiel ne contenant pas de fichiers propres à une plate-forme donnée, il n'aura pas besoin d'options d'optimisation. Par contre, lorsqu'un installateur inclut des fichiers propres à une plate-forme donnée, ceux-ci doivent être optimisés par application des règles **Vérifier la plate-forme**.

L'onglet **Journal de construction** permet d'afficher le journal des constructions précédentes (fichier XML).

- a. Pour construire l'installateur OfficeSuite, cliquez sur **Construire projet**.

La boîte de dialogue **Détails de construction** apparaît.

- b. Cliquez sur la flèche bleue située dans le coin inférieur gauche de cette boîte de dialogue pour afficher les détails de construction.

Lorsque la construction est terminée, un message vous en informe (dans ce cas, la construction ne devrait pas prendre plus d'une minute).

7. Tester le projet

Lorsque la construction est terminée, testez l'installateur en cliquant sur **Tester l'installation Web** ou sur **Tester l'installateur**. Dans le cadre de ce didacticiel, cliquez sur **Tester l'installation Web** pour lancer le navigateur Zero G et la page Web d'installation générée par la construction.

- a. Cliquez sur **Tester l'installation Web** : la page Web d'installation se charge et sollicite un accès sécurisé.
- b. Autorisez cet accès pour permettre à l'applet d'installation Web d'exécuter l'installateur InstallAnywhere.

L'installateur Web peut ensuite être lancé en un seul clic.

- c. Cliquez sur le bouton **Lancer installateur pour Windows** affiché sous l'image.

L'applet vérifie si l'espace disque est suffisant, puis télécharge et exécute l'installateur.

d. Exécutez l'installateur.

Après affichage du panneau **Installation terminée**, l'installateur lance OfficeSuite. L'icône **OfficeSuite** peut maintenant être sélectionnée dans le menu Démarrer de Windows pour exécuter le produit installé.

Solutions

La rubrique d'aide « Solutions » présente quelques informations et suggestions en réponse à certaines situations susceptibles de se présenter lors de la construction d'un installateur.

Intégrer un environnement d'exécution Java (JRE) à l'installateur

Les installateurs InstallAnywhere reposant sur Java, leur exécution exige une VM Java. Les installateurs InstallAnywhere peuvent intégrer une VM Java, ce qui élimine le risque de ne pas détecter dans le système cible de VM correcte pour le produit à installer ou pour l'installateur lui-même.

Les développeurs peuvent décider de laisser à l'utilisateur le choix de la VM Java à utiliser pour le produit installé.

Les installateurs InstallAnywhere sont exécutés à partir de leur VM intégrée ; pour pouvoir être intégrée, une VM doit être se présenter sous forme de « Pack VM ». La section « Plug-ins » du site Web de Zero G (<http://www.zerog.com/goto/vmpacks>) propose des Packs VM en téléchargement gratuit.

Aucune VM ne sera installée directement avec l'installateur dans les cas suivants :

- L'installateur est construit sans VM intégrée.
- L'installateur est construit avec une VM intégrée mais l'utilisateur décide d'utiliser une VM présente sur son système (en sélectionnant l'option appropriée dans l'action **Sélectionner une VM Java** de l'installateur).
- Les exécutables LaunchAnywhere n'ont pas été installés (y compris le fichier exécutable du désinstallateur).

Une VM sera installée directement avec l'installateur dans les cas suivants :

- L'installateur est construit avec une VM intégrée et l'utilisateur décide d'installer cette VM (en sélectionnant l'option appropriée dans l'action **Sélectionner une VM Java** de l'installateur).
- L'installateur est construit avec une VM intégrée, et l'action **Sélectionner une VM Java** de l'installateur a été désactivée dans la tâche Pré-installation d'Advanced Designer. Exceptions : si aucune application LaunchAnywhere n'est installée ou si aucun désinstallateur n'est créé, aucune VM n'est nécessaire, donc aucune VM n'est installée).
- La VM intégrée à la tâche **Construction du projet** d'Advanced Designer est sélectionnée.
- Si l'utilisateur supprime la VM utilisée en dernier par LaunchAnywhere, LaunchAnywhere recherche une VM correspondant au paramètre spécifié dans sa propriété `lax.nl.valid.vm.list`. Les installateurs déployés sur des systèmes non latins exigent une VM Java internationale.

Créer un lanceur pour une application Java

Une action InstallAnywhere de la tâche **Fichiers** permet de créer un lanceur pour une application Java : cliquez sur **Ajouter un lanceur**. Cette action crée un fichier (« exécutable LAX ») adapté aux caractéristiques natives de la plate-forme cible et dont l'exécution lance une application Java. Ce fichier LAX permet de lancer une

application Java comme si elle était native de la plate-forme cible ; ainsi, l'exécutable LaunchAnywhere peut être lancé en double-cliquant sur une icône ou être appelé directement à partir d'une ligne de commande. L'exécutable LaunchAnywhere (.LAX) est également chargé de configurer et d'appeler l'environnement d'exécution de l'application Java et de transmettre les arguments de ligne de commande requis (le cas échéant) par cette application.

Lorsque les développeurs ajoutent une action dans LaunchAnywhere, le produit leur donne l'option de rechercher la classe main de l'application Java considérée : ils peuvent alors spécifier cette classe ou laisser InstallAnywhere la détecter. Ils peuvent également définir les arguments de ligne de commande qui doivent être fournis à l'application Java lorsque l'exécutable LaunchAnywhere est lancé.

Sous Windows, InstallAnywhere ajoute l'extension .exe au fichier LAX ; sous Mac OS X, InstallAnywhere crée une application .app.

Lors de la définition du type de lanceur, les développeurs peuvent préciser si l'application Java utilise une interface graphique/GUI qui appelle javaw.exe (par défaut, les données stdout et stderr sont supprimées), ou si elle utilise une console qui appelle java.exe (par défaut, les données stdout et stderr sont dirigées vers la console). En outre, sous Windows, les lanceurs à console redirigent les données stdout et stderr vers la fenêtre de console à partir de laquelle ils ont été appelés.

L'« icône de lanceur d'application LaunchAnywhere » permet aux développeurs de sélectionner l'icône à utiliser pour l'exécutable LAX sur les plates-formes Mac OS X et Windows (ces icônes personnalisées ne sont pas disponibles avec les installateurs UNIX). Remarque : les dimensions des icônes OS X sont limitées à 128 x 128.

Remarques :

- Sous Windows, l'exécutable LaunchAnywhere n'affiche pas directement l'icône spécifiée : il affiche seulement l'icône générique .exe ; un fichier d'icône .ico réside dans le même dossier que l'exécutable, mais tous les raccourcis créés pendant l'installation qui pointent sur LaunchAnywhere héritent de l'icône personnalisée.
- InstallAnywhere ne supporte pas les fichiers GIF ou ICO à entrelacement (affichage progressif de l'icône).

Organiser les fonctions et les composants

Le « composant » représente le niveau le plus élémentaire de la structure d'un installateur. Chaque produit doit comporter au moins un composant, mais par défaut, la plupart des installateurs comportent au moins deux composants puisque le désinstallateur est considéré comme un composant à part entière.

L'architecture InstallAnywhere à base de composants permet aux développeurs d'organiser leur plan de déploiement : versions futures, suites d'installateurs et toute autre solution réutilisant leurs modules logiciels.

À mesure que vous ajoutez des fichiers dans un projet en les associant à des fonctions, InstallAnywhere crée automatiquement les composants correspondants. Bien qu'il convienne à la plupart des projets, ce mécanisme n'autorise pas une souplesse optimale : pour tirer parti au mieux des avantages des logiciels à base de composants, il est conseillé de gérer manuellement la création de ceux-ci.

Recommandations générales pour la création de composants

Commencez par déterminer les composants requis et leur structure.

Conseils de base :

- Pour tout fichier qui doit être mis à jour séparément, il est conseillé de créer un composant unique. Exemple : une fonction « Aide » peut être associée à la fois à un « Guide de l'utilisateur » et à des Javadocs. Comme il est probable que le « Guide de l'utilisateur » sera mis à jour plus souvent que les Javadocs, il est conseillé de créer deux composants distincts, de telle sorte que le composant « Guide de l'utilisateur » puisse être mis à jour – et suivi par un contrôle de version – séparément.
- Les composants doivent suivre une certaine logique. Si vous créez une suite d'installateurs, vous devez tenir compte des composants d'application qui sont partagés par les différents produits. Pour assurer le contrôle des versions d'un produit à base de composants, vous devez donc spécifier la version du composant dans la tâche **Structure | Composant | Propriétés** lorsque vous ajoutez ce composant.

Ajouter des composants

La gestion des composants est assurée par la tâche **Structure | Composants**. Chaque composant est créé avec un **Nom**, (long), un **Nom court** et un **ID unique**. Des valeurs de **Version** et de **Fichier clé** peuvent également être spécifiées. Le « fichier clé » identifie le composant auquel il est associé ; il s'agit généralement d'un fichier qui sera toujours inclus dans un composant.

Associer des fichiers aux composants

Pour associer des fichiers à un composant, sélectionnez l'option **Associer des actions aux composants** de la tâche **Fichiers**.

Supprimer les composants vides

Lorsqu'un projet contient des composants « vides » (composants qui ne sont plus utilisés ou qui ne sont plus associés à des fichiers), il est conseillé de les supprimer en cliquant sur le bouton **Nettoyer les composants** de la tâche **Structure | Composants**.

Intégrer des composants déjà installés sur le système cible

Si l'un des composants nécessaires à l'installateur est déjà présent dans le système cible, utilisez la fonction **Rechercher un composant dans le registre Windows** pour le retrouver (la fonction recherche le composant par son identifiant unique UUID). L'emplacement du composant peut ensuite être utilisé par l'installateur comme emplacement d'installation.

Le cas échéant, l'installateur peut comparer les versions et retrouver la plus haute. Il peut également rechercher le fichier clé d'un composant.

Ajouter des fonctions

La gestion des fonctions est assurée par la tâche **Structure | Fonctions**. Vous pouvez ajouter et supprimer des fonctions et associer certaines fonctions à certains types d'installation.

Associer des composants à des fonctions

Vous pouvez associer des composants à des fonctions en mode manuel (à l'aide de la tâche **Structure | Composants**), ou en mode automatique en associant directement les fichiers et les fonctions – ce qui a pour effet de créer les liens entre les fonctions et les composants. Si vous associez un fichier à une fonction qui n'est pas encore associée à un composant, le composant requis est créé automatiquement.

Remarque : si vous travaillez en mode manuel, il est conseillé de ne pas demander l'association directe (automatique) des fichiers et des fonctions, dans la mesure où cette opération entraîne la création et la modification des liens des composants ; il est préférable de procéder à des associations manuelles entre fichiers et composants.

Améliorer les installations

Réduire la taille de l'installateur

Les suggestions qui suivent permettent de réduire la taille de l'installateur.

Dans **Construction du projet | Distribution**, sélectionnez l'option **Optimiser la taille de l'installateur selon plate-forme**.

- Installateurs Swing – N'affichez pas l'image de fond.
- Billboards – Utilisez des fichiers de taille aussi faible que possible.
- Labels et images des panneaux – Évitez leur emploi.
- Étapes d'installation – Utilisez uniquement les étapes essentielles.
- Désinstallateur – Ne pas inclure si non essentiel.
- SpeedFolders – Recommandés pour regrouper les fichiers installés. Les SpeedFolders permettant de traiter un dossier comme une seule action d'installation, ils réduisent l'espace occupé et augmentent les performances.
- Locales – Limitez la construction aux valeurs requises.
- JRE – Ne pas intégrer systématiquement. Si un JRE doit absolument être intégré, utilisez une version 1.1 non internationalisée (le JRE Java 1.1.8 « English-only / Anglais uniquement » est le plus petit). Si vous devez impérativement utiliser un JRE Java 2, choisissez une version qui n'inclut pas de ressources internationales.

Optimiser un installateur UNIX

InstallAnywhere crée un seul installateur UNIX sans VM intégrée qui peut être exécuté sur la plupart des plates-formes UNIX. Les installateurs avec VM intégrée sont construits spécifiquement pour chaque plate-forme (et peuvent être optimisés pour celle-ci). Cet installateur UNIX sans VM intégrée dispose

des ressources nécessaires à chaque plate-forme UNIX, ce qui peut lui faire occuper un espace supérieur à celui des installateurs optimisés pour les différentes plates-formes UNIX avec VM intégrée. (toutefois, cet espace reste inférieur à l'espace occupé par un installateur UNIX construit sans optimisation de plate-forme). Pour des raisons similaires, l'installateur Pure Java ne bénéficiant jamais d'une optimisation de plate-forme, il occupe toujours le même espace – quelle que soit l'optimisation appliquée.

Installateurs complémentaires

L'action **Exécuter le fichier cible** permet de lancer un installateur (non InstallAnywhere) à partir d'un installateur InstallAnywhere. Les modules de fusion permettent également de lancer d'autres installateurs InstallAnywhere en vue de bénéficier de la meilleure optimisation et des meilleures performances possibles.

Équipe de développement

Avec les « chemins source », les développeurs peuvent définir des pointeurs vers les fichiers en utilisant des chemins variables et non des chemins absolus, ce qui permet de partager un fichier de projet avec d'autres membres d'une équipe de développement, même si les fichiers résident dans différents emplacements des systèmes de développement considérés. Avec les chemins source, il est même possible d'utiliser le même fichier de projet dans différents types de système d'exploitation – par exemple, en partageant un projet entre UNIX et Windows.

Pour spécifier des chemins source, choisissez **Édition | Préférences**, puis activez l'onglet **Chemins source**.

Chemins source – Présentation générale

Les chemins source sont automatiquement remplacés par le chemin le plus complet possible. Exemple – Soit deux chemins source définis comme suit :

```
$SOURCEPATH1$ = D:\temp\dir\foo
```

```
$SOURCEPATH2$ = D:\temp
```

Si vous ajoutez le fichier D:\temp\dir\foo\hello.txt, celui-ci sera automatiquement référencé par \$SOURCEPATH1\$/hello.txt, puisque \$SOURCEPATH1\$ correspond au chemin le plus complet possible.

Si un membre de l'équipe de développement ouvre ce projet et qu'un chemin source n'est pas défini, un message lui demande de déterminer et de spécifier ce chemin source.

Tout projet comporte plusieurs chemins source prédéfinis qui ne peuvent pas être modifiés :

- \$IA_HOME\$ est l'emplacement du système réservé à l'exécution d'InstallAnywhere, en général : c:\Program Files\InstallAnywhere.
- \$IA_PROJECT\$ est l'emplacement du système réservé à l'installation du projet InstallAnywhere.
- \$USER_HOME\$ est le dossier Home de l'utilisateur.

Ajouter des chemins source

Pour ajouter des chemins source à l'aide des variables « Chemin source », vous disposez des deux solutions décrites ci-dessous (section « Préférences InstallAnywhere » et section « Définir une variable d'environnement système »).

1. Préférences InstallAnywhere.
 - a. Choisissez Édition | Préférences.
 - b. Activez l'onglet **Chemins source**.
 - c. Cliquez sur **Ajouter**.
 - d. Dans la table, tapez le nom de la variable du chemin source – par exemple : `RESOURCE` (sans les signes dollar/\$, qui sont ajoutés automatiquement chaque fois que la variable est utilisée).
 - e. Cliquez dans la zone vide située sous **Dossier** : le bouton **Choisissez un dossier** apparaît. Cliquez sur ce bouton pour sélectionner l'emplacement cible de la variable de chemin source (par exemple, `c:\resources\test.txt`).
2. Définir une variable d'environnement système.
 - a. Affichez les variables d'environnement du système considéré. Sous Windows, cliquez avec le bouton droit sur l'icône **Poste de travail** du bureau et choisissez Propriétés **avancées | Variables d'environnement**. Sous UNIX ou Mac OS X, modifiez le fichier de configuration de shell approprié ou affectez la valeur requise à la variable directement à l'aide du shell.
 - b. Ajoutez une variable d'environnement pour le chemin source, précédé de la balise `IA_PATH_`.

Exemple - pour définir le chemin source '`SOURCE_PATH`', ajoutez la variable `IA_PATH_SOURCE_PATH`.

Mettre à jour l'emplacement des fichiers et des ressources

Lorsque l'emplacement d'un fichier ou d'un dossier a changé, il suffit de modifier l'emplacement de dossier indiqué dans le chemin source : InstallAnywhere met automatiquement à jour les références qui pointent vers les ressources correspondantes.

Lorsque vous ouvrez un projet et qu'InstallAnywhere ne trouve pas certaines ressources nécessaires à celui-ci (parce qu'elles ont été supprimées ou déplacées), un message vous demande de spécifier manuellement l'emplacement de ces ressources ou de les supprimer du projet. Si vous préférez ouvrir un projet sans mettre à jour l'emplacement de ses ressources, choisissez **Édition | Préférences** et activez l'option **Jamais**, puis activez l'onglet **Paramètres généraux** et modifiez la préférence **Chargement du projet**. Avec cette configuration, InstallAnywhere ouvre les projets sans vérifier l'emplacement de chaque ressource (cette vérification est effectuée lors de la construction).

Bilan de l'installation

La variable InstallAnywhere \$INSTALL_STATUS\$ permet de déterminer le bilan de l'installation ; elle peut prendre l'une des valeurs suivantes : \$INSTALL_STATUS\$: SUCCESS, WARNING, NONFATAL_ERROR ou FATAL_ERROR. La variable \$UNINSTALL_STATUS\$ joue le même rôle pour les désinstallations.

Débogage des installateurs

Pour déboguer un installateur InstallAnywhere, vous disposez de deux méthodes, qui sont décrites ci-dessous : le débogage peut avoir lieu pendant le cycle de développement ou après développement si un utilisateur rencontre un problème.

La plupart des installateurs InstallAnywhere utilisent la technologie LaunchAnywhere de Zero G. Les « lanceurs » LaunchAnywhere proposent un environnement ergonomique (exécution par double-clic, comportement familier, etc.) et de nombreuses fonctions de débogage.

Déboguer l'installateur pendant le cycle de développement

InstallAnywhere peut enregistrer les informations de débogage dans un fichier.

Choisissez **Projet | Config** : les champs regroupés dans la zone **Sortie de débogage de l'installateur** indiquent le nom des fichiers qui reçoivent les données stdout et stderr. Actions possibles pour les champs de texte : laisser vides (les informations sont supprimées) ; entrer la valeur « console » (les informations sont affichées sur une console active) ; entrer les chemins de destination des données stdout/stderr (les deux entrées doivent pointer sur le même fichier ; les chemins spécifiés doivent être absolus ; vous pouvez spécifier des chemins Java plutôt que les chemins spécifiques au système cible). Cette fonction permet aux développeurs d'utiliser une seule entrée pour plusieurs plates-formes.

Exemples :

Envoyer stderr vers : /tmp/outputfile.text.

Envoyer stdout vers : /tmp/outputfile.text

Ces paramètres dirigent la sortie – respectivement – vers le fichier /tmp/outputfile.txt (systèmes sous UNIX ou dérivés) ou vers le fichier C:\tmp\outputfile.txt (systèmes sous Windows). Le fichier lui-même (dans cet exemple, outputfile.text) contiendra la plupart des informations – mais pas forcément toutes les informations – nécessaires au débogage des installations InstallAnywhere.

Ces fichiers ne pouvant pas être désinstallés, il est conseillé de désactiver cette fonction avant de construire l'installateur. Toutefois, certains développeurs préfèrent conserver ces informations de sortie pour faciliter le débogage qui pourrait être nécessaire après le développement de l'installateur.

Déboguer l'installateur après développement

Cette rubrique décrit les problèmes de débogage que peuvent rencontrer les utilisateurs et d'autres problèmes post-développement.

Déboguer un installateur Win32

Pour consulter ou capturer la sortie de débogage d'un installateur Win32, maintenez enfoncée la touche <Ctrl> immédiatement au lancement de l'installateur et jusqu'à ce qu'une fenêtre de console apparaisse. Avant de quitter l'installateur, vous pouvez copier la sortie affichée sur cette console dans un fichier texte pour analyse ultérieure.

Sur certains systèmes Windows NT, la procédure est la suivante : lancez l'installateur en maintenant la touche <Ctrl> enfoncée, réinitialisez le tampon de défilement arrière (scroll back buffer) de la fenêtre de console, puis quittez et relancez l'installation.

Si vous ne parvenez pas à capturer la sortie de la console (ce qui est souvent le cas avec les systèmes sous Win9x dont la capacité de console est limitée), appliquez la procédure ci-dessous. Lancez l'installateur et laissez-le extraire les fichiers nécessaires. Lorsque la fenêtre **Préparation de l'installation** apparaît, une option permet d'afficher le contenu du dossier Temp de Windows : recherchez un dossier Temp dont le nom commence par la lettre « I » suivie d'une séquence numérique (par exemple, « I1063988642 » sans les guillemets). Vérifiez qu'il s'agit du dossier le plus récent en triant la liste par date (colonne **Modifié le**). Ouvrez ce dossier Temp et supprimez le fichier **sea_loc**. Revenez à l'installateur, cliquez sur **OK** et annulez l'installation dès que possible.

Revenez au dossier Temp dans lequel vous avez supprimé le fichier **sea_loc**. Affichez le contenu du sous-dossier **Windows** et recherchez un fichier .exe (en général, install.exe). Ce sous-dossier doit également contenir un fichier portant le même nom, mais avec extension .lax ; ouvrez ce fichier .LAX dans un éditeur de texte et modifiez-le comme suit :

La ligne `lax.stderr.redirect=` et la ligne `lax.stdout.redirect=` deviennent respectivement :
`lax.stderr.redirect=output.txt` et `lax.stdout.redirect=output.txt`

Après avoir effectué ces modifications, enregistrez le fichier et lancez l'exécutable (.exe). Lorsque l'installation est terminée, le dossier qui contient ce fichier .lax doit également contenir un fichier output.txt. (ce dernier contient les mêmes informations que celles qui ont été affichées par la console).

Déboguer un installateur UNIX/Linux

Pour capturer la sortie de débogage à partir d'une ligne de commande UNIX, la procédure est la suivante : avant de lancer l'installateur, entrez l'une des commandes suivantes (selon shell) :

```
export LAX_DEBUG=true ou setenv LAX_DEBUG true ou
LAX_DEBUG=true ou set LAX_DEBUG (ou commande similaire adaptée au shell).
```

Lancez ensuite l'installateur : les données obtenues en sortie doivent permettre le débogage de l'installateur.

Déboguer un installateur Mac OS X

InstallAnywhere utilise les couches standard de Mac OS X pour afficher les données de sortie. Pour capturer la sortie de débogage d'un installateur OS X,

exécutez `console.app` : la sortie est générée dans `/Applications/Utilities`. Pour conserver ces informations, coupez-les dans la fenêtre de console et collez-les dans un fichier.

Déboguer un installateur Pure Java

Pour déboguer les installateurs Pure Java ou les installateurs destinés aux plates-formes autres que Windows, UNIX et Mac OS X, vous disposez de deux méthodes :

- Créez le fichier `ia_debug` dans le dossier du JAR (qui contient l'installateur) ; la sortie ne sera pas enregistrée dans ce fichier, mais l'existence de ce fichier entraînera la redirection de la sortie vers la console.
- Configurez les **Paramètres généraux** de manière à créer la sortie de débogage avant de construire l'installateur.

Déboguer les exécutables lancés par LaunchAnywhere :

Dans la mesure où les installateurs InstallAnywhere utilisent les exécutables LaunchAnywhere, les procédures décrites ci-dessus sont également utiles pour déboguer les applications installées qui font appel à la technologie de lanceur Java de LaunchAnywhere. En général, il est possible d'éditer le fichier LAX de manière à demander au lanceur de toujours générer la sortie de débogage (ce comportement peut ensuite être modifié lors des phases de qualification et de version finale/final release).

Pour générer la sortie de débogage :

1. Dans Advanced Designer, sélectionnez le lanceur requis.
2. Cliquez sur **Éditer les propriétés**.
3. Modifiez les valeurs de variable comme suit :
`lax.stderr.redirect=` et `lax.stdout.redirect=` deviennent :
`lax.stderr.redirect=output.txt` et `lax.stdout.redirect=output.txt`
4. Après installation normale, éditez le fichier `.lax` en appliquant les instructions précédentes.
5. Cette procédure doit être répétée pour chaque installation.

Remarque : pour UNIX, posez `LAX_DEBUG=true` ; pour Mac OS X, ouvrez le fichier `Console.app`.

Analyser les informations de débogage

Exemple de sortie de débogage InstallAnywhere (version tronquée) :

```
InstallAnywhere 6 Enterprise Build 2200
Wed Dec 07 17:13:04 PST 2002
```

```
Current Total Java heap = 24575 kB
Current Free Java heap = 22581 kB
```

```
No arguments.
```

```
java.class.path = C:\Program Files\InstallAnywhere 6
Enterprise\resource C:\Program Files\InstallAnywhere 6
Enterprise\resource\swingall.jar C:\Program Files\InstallAnywhere
```

```
6 Enterprise\resource\compiler.zip C:\Program
Files\InstallAnywhere 6
Enterprise\IAClasses.zip C:\Program Files\InstallAnywhere 6
Enterprise\lax.jar C:\program files\installanywhere 6
enterprise\jre\lib\rt.jar

ZGUtil.CLASS_PATH = C:\Program Files\InstallAnywhere 6
Enterprise\resource C:\Program Files\InstallAnywhere 6
Enterprise\resource\swingall.jar
C:\Program Files\InstallAnywhere 6
Enterprise\resource\compiler.zip
C:\Program Files\InstallAnywhere 6
Enterprise\IAClasses.zip C:\Program Files\InstallAnywhere 6
Enterprise\lax.jar

java.version = 1.3.1
java.vendor = Sun Microsystems Inc.
java.home = c:\program files\InstallAnywhere 6
enterprise\jre\bin\..
java.class.version = 45.3
```

Principales informations de débogage : VM sélectionnée (avec indication de sa version), locale, architecture et système d'exploitation du système cible, etc.

Utiliser l'action Afficher informations de débogage

L'action **Afficher informations de débogage** affiche les informations stockées par l'installateur (ou accessibles à celui-ci). Ces informations peuvent être dirigées vers la sortie standard ou enregistrées dans un fichier pour analyse ultérieure. Parmi les options disponibles dans l'action **Afficher informations de débogage**, les plus utiles pour le dépannage/débogage sont **Imprimer les variables InstallAnywhere** et **Imprimer les propriétés Java** car elles permettent d'accéder aux variables les plus souvent utilisée pour la formulation des règles. Si des erreurs sont provoquées par certaines règles – ou si une action prévue ne se déclenche pas –, l'action **Afficher informations de débogage** permet de vérifier les valeurs prises en compte par InstallAnywhere pour les variables et les propriétés Java utilisées par les règles InstallAnywhere.

Déboguer par message

Il peut être intéressant de déboguer certains modules d'une installation pendant le développement d'un installateur. L'une des fonctions d'InstallAnywhere (Enterprise Edition uniquement) permet aux développeurs d'ajouter un panneau de message pouvant afficher des valeurs spécifiques de variable InstallAnywhere.

Exemple :

1. Une règle énonce la restriction suivante : Installer seulement si \$prop.os.name\$=Solaris
2. L'installation continue mais l'action affectée à cette règle ne s'exécute pas.
3. Les développeurs ajoutent le message suivant : prop.os.name a la valeur suivante : \$prop.os.name\$
4. Lorsque l'installation s'exécute avec prop.os.name ayant la valeur SunOS (et non Solaris), la règle peut être reformulée de manière à correspondre au nom correct.

Assistant de projet

L'Assistant de projet guide les développeurs à travers les étapes de base de la construction d'un installateur. Par défaut, l'Assistant de projet construit un installateur Web. À partir de toute fenêtre de l'Assistant de projet, les développeurs peuvent passer à Advanced Designer.

Lorsque vous lancez InstallAnywhere, l'Assistant de projet apparaît – sauf si le mode de lancement par défaut a été modifié dans **Édition | Préférences**. Lorsque l'Assistant de projet démarre, il affiche la boîte de dialogue **Nouveau projet**.

Nouveau projet

La boîte de dialogue **Nouveau projet** permet de créer et de nommer des projets InstallAnywhere. Pour nommer un projet, cliquez sur **Enregistrer sous** : les options de la boîte de dialogue **Enregistrer nouveau projet** qui apparaît permettent de nommer un projet, de créer un dossier et de retrouver un projet.

Remarque : dans InstallAnywhere Enterprise Edition, vous devez sélectionner un modèle pour tout nouveau projet. Si c'est votre premier projet, utilisez le modèle **Basic Project**.

Pour ouvrir un projet existant, activez l'option **Ouvrir un projet existant**, sélectionnez le projet dans la liste qui apparaît et cliquez sur **Suivant**.

Définir les détails

La fenêtre **Définir les détails** de l'Assistant de projet présente les informations de base de l'installateur, y compris nom du dossier d'installation et nom du produit. Sur les plates-formes Windows, si « OfficeSuite » est sélectionné comme **Nom du dossier d'installation** et « OfficeSuiteAdvanced » comme **Raccourci de l'application**, il suffit de choisir **Démarrer | Programmes | OfficeSuite | OfficeSuite** pour lancer l'installateur OfficeSuite.

Toujours sous Windows, si « OSA » est sélectionné comme **Nom du dossier d'installation**, le système crée le dossier suivant : C:\Program Files\OSA. Le nom du fichier exécutable de l'installateur sera le texte contenu dans le champ **Nom de l'installateur**.

Ajouter des fichiers

La fenêtre **Ajouter des fichiers** de l'Assistant de projet permet aux développeurs d'ajouter et de supprimer des fichiers ou/et des dossiers du projet d'installateur en cours de création. Lorsqu'ils cliquent sur le bouton **Ajouter des fichiers**, la boîte de dialogue **Ajouter des fichiers dans le projet** s'affiche. **Ajouter Tout** sélectionne les fichiers ou le dossier affiché dans le panneau de sélection supérieur ; les objets sélectionnés s'affichent dans le panneau de sélection inférieur.

La boîte de dialogue **Ajouter des fichiers dans le projet** dispose de la barre de navigation standard d'InstallAnywhere.

Choisir la classe main

La fenêtre **Sélectionner la classe main** de l'Assistant de projet permet de sélectionner une classe main pour l'installateur. Dans Java, la classe main est le point de départ pour l'exécution d'un fichier JAR. Si vous cliquez sur le bouton **Retrouver automatiquement les classes main**, l'Assistant de projet affiche les classes main disponibles pour vous permettre d'en sélectionner une.

Définir le classpath

La fenêtre **Définir le classpath** de l'Assistant de projet permet de sélectionner le classpath nécessaire à l'installateur. Le classpath permet aux applications Java de retrouver les classes Java dont elles ont besoin pour s'exécuter. Cliquez sur **Définir le classpath automatiquement**.

Construire l'installateur

La fenêtre **Construire Installateur** de l'Assistant de projet permet de sélectionner la ou les plates-formes cible de l'installateur. Dans cette fenêtre, vous pouvez également sélectionner la VM Java qui sera intégrée à l'installateur. Après avoir sélectionné la ou les plates-formes et la VM, cliquez sur **Construire** pour construire l'installateur.

Tester l'installateur

La fenêtre **Tester l'installateur** de l'Assistant de projet permet aux développeurs de tester l'installateur Web construit par l'Assistant de projet. Si vous cliquez sur le bouton **Tester**, le navigateur par défaut du système s'affiche et le produit est installé sur la machine de développement.

Advanced Designer

Alors que l'Assistant de projet facilite la création d'un installateur en proposant aux développeurs quelques étapes simples, Advanced Designer permet de disposer d'une plus grande précision et d'accéder à l'ensemble des puissantes fonctions d'InstallAnywhere. Avec Advanced Designer, les développeurs peuvent affecter des fichiers et des actions à des ensembles de fonctions (feature sets), permettant ainsi à l'utilisateur de désigner les fichiers à installer. En outre, les développeurs peuvent définir des règles qui – en fonction de la plate-forme cible – installent des fichiers différents ou sélectionnent un emplacement d'installation différent. Avec Advanced Designer, les développeurs disposent d'un contrôle plus précis sur l'ensemble des options de création de leurs projets d'installation.

Les tâches et sous-tâches Advanced Designer sont regroupées dans la partie gauche de l'environnement de développement d'Advanced Designer. Ces tâches et sous-tâches sont décomposées en sous-ensembles distincts du processus de création de l'installateur.

Projet

La tâche **Projet** affiche les informations relatives au projet InstallAnywhere en cours, assure la configuration du projet, propose les options d'intégration ou de définition des VM correctes et définit les options du produit, d'installation des fichiers et de localisation.

Infos

La tâche **Projet | Infos** définit les informations de base de l'installateur à créer, affiche des informations sur le projet d'installateur InstallAnywhere et permet aux développeurs de prendre des décisions en fonction des informations consignées dans le journal d'installation de l'installateur.

Informations du projet

Dans la zone **Informations du projet**, les développeurs définissent les détails nécessaires à l'installateur, par exemple le nom du produit à installer. Par ailleurs, les développeurs définissent le nom de l'installateur lui-même, ainsi que les informations qui seront affichées dans la barre de titre de l'installateur.

Journal d'installation

Cette zone permet de demander la génération d'un journal d'installation et de préciser les fichiers installés et les actions exécutées par l'installateur. Les développeurs peuvent définir le format de ce journal (texte ou XML) et préciser si le journal d'installation doit être supprimé lorsque l'application est désinstallée.

Remarque : pour définir l'emplacement du journal d'installation, il suffit d'affecter une valeur à la variable InstallAnywhere \$INSTALL_LOG_DESTINATION\$

Description

Dans la sous-tâche **Projet | Description**, les développeurs peuvent saisir les informations de produit et de fournisseur de manière à identifier leur installateur sous une forme unique dans le registre du produit. Pour les systèmes d'exploitation cible, le « registre du produit » est essentiellement une base de données de valeurs de configuration qui assure la mémorisation et le suivi des fonctions et des composants des produits installés et qui exécute des tâches internes, par exemple associer les extensions de nom de fichier avec les applications appropriées. Il est impératif de définir correctement l'ID et la version du produit pour pouvoir utiliser ultérieurement l'action **Rechercher un composant dans le registre** ; en effet, InstallAnywhere utilise les valeurs d'ID de produit pour retrouver (dans le registre) l'emplacement des composants requis.

Params de fichier

Les sous-tâches **Params de fichier** permettent aux développeurs de définir les « horodates » (date/heure) des fichiers installés et le type de remplacement (écrasement) à appliquer lorsque les emplacements cible de l'installation contiennent des fichiers portant le même nom que les fichiers à installer.

Les horodates sont comparées lorsque le système cible contient un fichier portant le même nom que le fichier à installer.

Horodates des fichiers installés

La zone **Comportement de l'horodate de modification de fichier** propose aux développeurs plusieurs options pour traiter l'horodate des fichiers installés :

- **Conserver l'horodate du fichier** – Conserve l'horodate affectée par défaut au fichier, soit la date/heure de dernière modification de ce fichier (telle qu'affectée automatiquement par le système d'exploitation dans lequel ce fichier a été créé et/ou enregistré). Ainsi, ce fichier afficherait l'heure de sa dernière modification et non l'heure de son installation.
- **Horodate d'installation** – Les horodates de création et de modification des fichiers correspondent à l'horodate d'installation des fichiers sur le système cible. Lorsque cette option est activée, tous les fichiers installés sont donc affectés des mêmes propriétés (horodates) de création et de modification.
- **Spécifier l'horodate** – Permet aux développeurs d'associer une horodate de leur choix aux fichiers installés (les valeurs spécifiques d'horodate peuvent être sélectionnées dans les listes déroulantes).

Remarques relatives aux horodates :

- La propriété d'horodate des fichiers peut être fixée avant ou après la date actuelle.
- Les VM Java 2 traitent les horodates correctement, ce qui n'est pas toujours le cas des VM Java antérieures.
- InstallAnywhere affiche les horodates en fonction du fuseau horaire actuel du système cible. En fait, InstallAnywhere assure la gestion des horodates en fonction de l'heure GMT (méridien de Greenwich/heure de Londres), mais les affiche en fonction du fuseau horaire local.

Comportement des fichiers en service (Windows uniquement)

Si le système cible de l'installation est sous Windows, il est possible que certains fichiers soient déjà en service. Lorsque l'option **Après redémarrage, remplacer les fichiers en service** est sélectionnée, l'installateur détermine si les fichiers en cours d'installation risquent de remplacer (écraser) des fichiers qui sont déjà en service ; si c'est le cas, InstallAnywhere déclare ces fichiers dans le registre Windows afin qu'ils soient correctement installés au prochain redémarrage du système.

Comportement par défaut de l'écrasement de fichier

Lorsque l'installation porte sur des fichiers qui existent déjà dans le ou les emplacements d'installation sélectionnés sur le système cible, l'installateur doit adapter son comportement en ce qui concerne l'écrasement de ces fichiers. Les fichiers sont considérés comme identiques s'ils sont affectés des mêmes valeurs de nom et de chemin d'accès. La décision prise par l'installateur (écraser les fichiers existants ou ne pas installer les fichiers identiques) s'appuie sur une comparaison entre l'horodate des fichiers présents dans l'ordinateur cible et l'horodate des fichiers à installer.

Comportement par défaut : écraser les fichiers existants s'ils sont plus anciens, afficher un message si les fichiers existants sont plus récents.

Les options d'écrasement de fichier peuvent être définies par défaut ou proposées à la validation de l'utilisateur sous forme de messages.

Option d'écrasement	Résultat
Toujours écraser	Installer les fichiers sans donner à l'utilisateur la possibilité de décider de l'écrasement des fichiers qui existent déjà sur son ordinateur.
Ne jamais écraser	Conserver les fichiers identiques qui existent dans l'ordinateur de l'utilisateur / ne pas les remplacer par les fichiers identiques en cours d'installation. Aucune option n'est proposée à l'utilisateur.

Écraser si plus ancien, ne pas installer si plus récent	Si les fichiers en cours d'installation sont plus récents (horodate postérieure) que les fichiers identiques existant dans l'ordinateur de l'utilisateur, écraser ces derniers. Aucune option n'est proposée à l'utilisateur.
Écraser si plus ancien, message si plus récent	Écraser les fichiers existant dans l'ordinateur de l'utilisateur et qui sont identiques aux fichiers en cours d'installation si ces derniers sont plus récents (horodate postérieure) que les fichiers existants sans proposer d'option à l'utilisateur ; par contre, si les fichiers à installer sont plus anciens (horodate antérieure) que les fichiers existants, afficher un message pour permettre à l'utilisateur de prendre une décision.
Demander à l'utilisateur si plus ancien, ne pas installer si plus récent	Demander à l'utilisateur si les fichiers existant dans son ordinateur sont plus anciens que les fichiers en cours d'installation : si les fichiers existants sont plus récents (horodate postérieure), les fichiers InstallAnywhere ne sont pas installés et aucun message ne s'affiche.
Toujours demander à l'utilisateur	Afficher un message à l'intention de l'utilisateur chaque fois qu'un des fichiers à installer est identique à un fichier existant dans l'ordinateur cible.

Plates-formes

InstallAnywhere peut être exécuté sur toute plate-forme supportant Java, mais certains aspects de l'installation doivent être définis séparément en fonction du système d'exploitation cible. Exemples : dossiers par défaut d'installation et des liens (UNIX/Linux), dossiers par défaut d'installation et des raccourcis (Windows), dossiers par défaut d'installation et des alias (Macintosh).

Pour Windows, il suffit de retrouver les dossiers par défaut d'installation et des raccourcis. Mac OS X définit des autorisations par défaut pour les dossiers et les fichiers qui doivent être créés sur le système cible. Les développeurs peuvent également activer une fonction d'authentification (en accordant à l'utilisateur les autorisations lui permettant de procéder à l'installation même s'il n'est pas connecté avec les privilèges appropriés) et sélectionner les versions de VM Java correctes. Pour les installations sous Linux, InstallAnywhere ajoute les paramètres RPM (Red Hat). RPM est un ensemble d'outils d'installation utilisés par les installateurs InstallAnywhere dans l'environnement Linux et dans certains environnements UNIX. RPM permet à l'installateur d'interagir avec la base de données RPM, y compris par écriture de valeurs.

Locales

La sous-tâche **Locales** définit les langues pour lesquelles l'installateur doit être créé. Une locale est activée lorsque sa case est cochée.

Les locales activées génèrent un « fichier de locale », qui est enregistré dans un sous-dossier du dossier contenant le fichier de projet InstallAnywhere.

Pour personnaliser une locale, il suffit d'éditer ce fichier. Pour plus de détails sur la localisation et les locales, voir la rubrique **Localisation**, ci-dessous.

Règles

La sous-tâche **Projet | Règles** permet aux développeurs d'ajouter des tâches à un installateur avant toutes tâches d'installation ou de pré-installation. Les développeurs peuvent utiliser cette option pour vérifier si la plate-forme cible convient à l'installation envisagée ou si l'utilisateur est connecté au dossier racine/root (ou s'il dispose des autorisations nécessaires pour exécuter l'installation).

Configuration

La sous-tâche **Projet | Config** permet aux développeurs d'établir la liste des VM Java pouvant être utilisées par l'installateur et de définir la taille de mémoire/heap pour ces VM. Les développeurs peuvent activer une sortie de débogage et préciser si cette sortie doit être dirigée vers un fichier ou vers une console. La sous-tâche **Config** permet également aux développeurs de définir des commandes pour la VM Java en complément de celles qui ont déjà été définies par l'installateur.

Sortie de débogage de l'installateur

Si les champs **Envoyer les données stderr vers** ou **Envoyer les données stdout vers** sont vides, la sortie de l'installateur est ignorée. Pour surveiller les données de sortie, il suffit de les diriger vers une console en entrant console dans le champ ; pour enregistrer la sortie dans un fichier, entrez le nom de ce fichier dans le champ.

Liste des VM correctes pour l'installateur Cette liste permet aux développeurs de sélectionner les VM Java pouvant être utilisées par l'installateur. Cette option est généralement utilisée pour sélectionner des VM qui ont été complètement testées. LaunchAnywhere recherche les VM séquentiellement, en fonction de leur type (JRE Java 1, JRE Java 2, JDK, etc.). Les types de VM corrects sont listés dans la propriété `lax.nl.valid.vm.list` de l'exécutable LaunchAnywhere.

Selon la plate-forme considérée, la recherche LaunchAnywhere se déroule comme suit :

- Windows – Recherche du chemin système, puis du registre système.
- UNIX – Recherche du chemin système.
- Mac OS X – LaunchAnywhere utilise la VM spécifiée par la tâche **Projet | Plates-formes | Mac OS X**.

Taille mémoire (heap) pour VM de l'installateur

Si vous rencontrez des problèmes de mémoire insuffisante, il suffit d'augmenter la taille de la mémoire « heap » (cette situation se produit souvent avec les projets qui installent un grand nombre de fichiers).

Arguments d'installateur (facultatif)

Pour disposer des options de configuration de VM Java qui ne sont pas disponibles dans Advanced Designer, il suffit de spécifier des paramètres

complémentaires en ligne de commande dans le champ (facultatif)

Arguments | Autres arguments. Ces arguments seront transmis à la VM Java utilisée par l'installateur.

Java

La sous-tâche **Projet | Java** permet aux développeurs d'optimiser les paramètres du classpath et de décider s'ils doivent installer ou non la VM Java livrée avec InstallAnywhere. Les développeurs peuvent décider de ne pas installer de VM, d'installer une VM seulement pendant l'installation ou d'installer et de conserver une VM sur système cible. Si les développeurs décident d'installer une VM, la liste **Dossier d'installation de la VM** suggère différents emplacements.

Interface d'installation

Paramètres d'affichage

La sous-tâche **Interface d'installation | Params d'affichage** présente trois onglets qui permettent de configurer l'apparence et le comportement (« look and feel ») de l'installateur. Ces onglets permettent aux développeurs de personnaliser la plupart des éléments graphiques et des panneaux de progression de l'installateur. Chaque onglet comporte un bouton **Aperçu** qui permet de prévisualiser l'installateur dans sa configuration actuelle.

L'onglet **Interface d'installation | Params d'affichage | Paramètres généraux d'UI** permet de définir les paramètres généraux d'apparence et de comportement de l'interface de l'installateur. Les développeurs peuvent sélectionner le mode de travail de l'installateur (GUI, console ou silencieux – certains installateurs pouvant supporter ces trois modes en même temps) et le mode graphique, soit AWT (bibliothèque de classes « Abstract Windows Toolkit », en code natif) ou Swing (produit d'accompagnement d'AWT, écrit en Java). Swing supporte l'affichage d'une image de fond (par exemple, le logo d'une société) dans l'installateur ; cette image peut également être définie dans l'onglet **Interface d'installation | Params d'affichage | Paramètres généraux d'UI**.

Pour renseigner l'utilisateur sur l'exécution de l'installation à l'aide de « panneaux de progression », cochez la case **Additions aux panneaux de l'installateur de GUI** ; si cette case n'est pas cochée, les onglets **Additions du panneau d'installateur** et **Panneau de progression d'installation** ne sont pas accessibles parce qu'il n'existe alors aucun panneau de progression complémentaire à modifier (les panneaux « complémentaires » décrivent les images ou les labels des actions).

L'onglet **Params d'affichage | Additions du panneau d'installateur** permet aux développeurs de sélectionner les panneaux d'installateur qui s'afficheront dans le rectangle vertical de gauche. Si le bouton radio **Liste des étapes d'installateur** est activé, les labels d'étape personnalisables s'affichent ; si le bouton radio **Images** est activé, aucun label d'étape n'est affiché. Lorsque le bouton radio **Images** est activé, les développeurs peuvent sélectionner une image par défaut dans la zone **Image du panneau d'installateur/désinstallateur par défaut** de l'onglet **Additions du panneau d'installateur**. Lorsque l'option **Images** est activée, l'image par

défaut peut être remplacée en spécifiant une autre image dans l'onglet **Panneau de progression d'installation**. Les développeurs peuvent également décider de ne pas afficher d'image, ou d'utiliser la même image que le panneau précédent.

Si le bouton radio **Liste des étapes d'installateur** de l'onglet **Params d'affichage | Additions du panneau d'installateur** est activé, l'option/case à cocher située dans la zone **Image de fond des étapes d'installateur** devient disponible ; cette option permet aux développeurs de sélectionner l'image à afficher dans le rectangle vertical de gauche, dans lequel s'affichent les labels des étapes d'installation.

En mode Swing, une image transparente peut également être affichée par-dessus l'image de fond sélectionnée dans l'onglet **Params d'affichage | Paramètres généraux d'UI**.

La partie inférieure du panneau **Additions du panneau d'installateur** permet aux développeurs de consulter et de modifier les labels des étapes d'installation. Les boutons regroupés à droite de la liste des labels permettent d'ajouter et de supprimer un label, de changer l'ordre des labels dans la liste et d'éditer le texte d'un label. Le bouton **Remplissage auto** prévoit un panneau d'installateur pour chaque action par panneau ajoutée aux tâches Pré-installation et Post-installation. Le bouton **Sélectionner les icônes** permet aux développeurs de modifier les graphismes carrés affichés à gauche des labels (par défaut, ces icônes sont une double flèche pour l'étape en cours/les étapes à exécuter et une coche pour les étapes d'installation terminées).

Remarque : par défaut, la taille du panneau de progression d'installation est de 380 x 270 pixels, mais les dimensions de l'installateur peuvent varier légèrement selon la plate-forme cible, afin d'améliorer l'affichage du texte et des différentes polices.

Billboards

Un « billboard » est une image qui s'affiche dans la partie droite des panneaux de l'installateur pendant l'installation des fichiers. Si l'installation doit prendre un certain temps, il est possible de prévoir plusieurs billboards ; pour les installations moins volumineuses (par exemple, le didacticiel OfficeSuite), un seul billboard est affiché. Des billboards peuvent également être associés aux fonctions, mais ils s'affichent seulement lorsque leur fonction respective est sélectionnée pour installation.

Lorsqu'un installateur utilise plusieurs billboards, ceux-ci s'affichent dans l'ordre spécifié par **Projet | Billboards | Liste des billboards**.

Remarque : par défaut, la taille des billboards est de 587 x 312 pixels, mais les dimensions de l'installateur peuvent varier légèrement selon la plate-forme cible, afin d'améliorer l'affichage du texte et des différentes polices.

Aide

La sous-tâche **Interface d'installation | Aide** permet d'ajouter une fonction d'aide à l'installateur. L'option (bouton radio) **HTML** permet de modifier le formatage du texte des messages d'aide. Dans la fenêtre inférieure du panneau, les développeurs peuvent saisir un message standard (qui sera le

même pour tous les panneaux) ; pour associer un texte différent à chaque panneau de l'installateur, sélectionnez l'option **Utiliser un texte d'aide différent pour chaque panneau**. Le personnaliseur d'action affiché à la partie inférieure des tâches **Fichiers**, **Pré-installation**, **Post-installation**, **Pré-désinstallation** et **Post-désinstallation** comporte un onglet **Aide** qui permet d'ajouter et de modifier le message HTML ou texte associé à chaque panneau d'installateur. Lorsque l'option **Utiliser un texte d'aide différent pour chaque panneau** est activée, l'onglet **Aide** du personnaliseur d'action est disponible.

Structure

La tâche Structure permet aux développeurs d'organiser les types d'installation, les fonctions du produit, les composants et les modules de fusion. Les « types d'installation » et les « fonctions » permettent à l'utilisateur de sélectionner différentes options d'installation. Le « composant » est le plus petit élément pouvant être sélectionné par un ensemble de fonctions. Les types d'installation correspondent à des groupes de fonctions ; ils permettent aux développeurs de structurer leur installateur. Les composants peuvent être de simples fichiers ou des actions complexes qui sont nécessaires pour installer et exécuter correctement certaines applications ou fonctions.

Il existe de nombreuses interactions entre les sous-tâches **Types d'installation**, **Fonctions** et **Composants** et la tâche **Fichiers**, par exemple :

- 1) Si un type d'installation est ajouté dans la tâche **Structure | Types d'installation**, des fonctions peuvent être affectées à ce type d'installation à l'aide de **Structure | Fonctions**.
- 2) Si une fonction est ajoutée dans **Structure | Fonctions**, des composants peuvent être affectés à cette fonction à l'aide de **Structure | Composants**.
- 3) Si un composant est ajouté dans **Structure | Composants**, des fichiers et/ou des actions peuvent être associé(e)s à ce composant après avoir été ajouté(e)s dans la tâche **Fichiers**.

Pour plus de détails sur la structure d'un installateur, voir la section « Organiser les fonctions et les composants » de la rubrique « Solutions ».

Les « modules de fusion » sont des sous-projets InstallAnywhere ; ils permettent aux développeurs de créer des modules réutilisables de projets InstallAnywhere.

Types d'installation

Un type d'installation correspond à un ensemble de fonctions. La sous-tâche **Structure | Types d'installation** permet aux développeurs d'ajouter, de nommer, de supprimer et d'agencer les types d'installation d'un installateur. Dans la **Liste des types d'installation**, les développeurs définissent le ou les types d'installation qui seront proposés comme option par défaut à l'utilisateur. La sous-tâche **Structure | Fonctions** permet d'associer des fonctions à des types d'installation.

Lorsque l'installateur demande les informations des types d'installation, chaque type est représenté par un élément graphique ; le bouton **Sélectionner une image** permet aux développeurs de sélectionner l'élément graphique requis.

Des « règles » peuvent être associées à un type d'installation : cette association est définie en sélectionnant **Règles** dans le personnaliseur et en ajoutant les règles requises. Les règles associées à un type d'installation sont évaluées avant l'installation des fonctions correspondant à ce type ; si une ou plusieurs de ces règles a la valeur **False**, le type d'installation n'est pas proposé (affiché).

Fonctions

Les fonctions correspondent à des activités du produit ; elles sont présentées à l'utilisateur pour permettre à celui-ci de décider s'il souhaite les installer ou non. Les « fonctions de produit » InstallAnywhere correspondent à des groupes de composants. Pour définir le regroupement logique de certains composants en fonctions, les développeurs associent des composants aux fonctions dans la sous-tâche **Structure | Composants**.

La sous-tâche **Structure | Fonctions** permet d'ajouter, de nommer, de supprimer et de réagencer les fonctions.

Des « règles » peuvent être associées à un ensemble de fonctions : cette association est définie en sélectionnant **Règles** dans le personnaliseur et en ajoutant les règles requises. Les règles associées à un ensemble de fonctions sont évaluées avant l'installation de cet ensemble ; si une ou plusieurs de ces règles a la valeur **False**, l'ensemble de fonctions n'est pas proposé (affiché).

Composants

Le « composant » est le plus petit élément de structure pouvant intervenir dans la création d'un installateur. Contrairement aux fonctions et aux types d'installation, les composants peuvent exister sous plusieurs versions. Les composants sont associés à un identifiant unique pour permettre aux développeurs de cibler les mises à jour ou d'exécuter l'action **Rechercher un composant dans le registre** pour retrouver un composant spécifique.

Le « fichier clé » est un fichier qui doit être présent dans toutes les versions ultérieures d'un composant ; il est utilisé pour définir l'emplacement des composants lorsque l'action **Rechercher un composant dans le registre** est exécutée.

La sous-tâche **Structure | Composants** permet de manipuler les composants : ajouter, nommer, supprimer, réagencer, identifier et affecter une version.

Des « règles » peuvent être associées à un ensemble de composants : cette association est définie en sélectionnant **Règles** dans le personnaliseur et en ajoutant les règles requises. Les règles associées à un ensemble de composants sont évaluées avant l'installation de cet ensemble.

Modules de fusion

La sous-tâche **Structure | Modules** permet d'importer des modules de fusion dans un projet InstallAnywhere. Les modules de fusion sont créés sous forme d'options de sortie de construction dans **Construction du projet | Distribution**.

Fichiers

La tâche **Fichiers** est l'une des plus importantes d'InstallAnywhere : elle permet d'afficher la hiérarchie des dossiers et des fichiers du système cible et d'associer des fichiers, des dossiers et des actions aux composants ou aux fonctions du produit. On notera que cette souplesse facilite le développement des installateurs, mais qu'elle peut également entraîner une certaine confusion pour les nouveaux utilisateurs d'InstallAnywhere.

Les développeurs peuvent ajouter des actions directement dans la hiérarchie de dossiers et de fichiers affichée par la tâche **Fichiers**. Les actions étant liées aux tâches, les actions disponibles dans la tâche **Fichiers** sont différentes de celles qui sont disponibles à partir des autres tâches. Les actions disponibles dans la tâche **Fichiers** permettent de créer des dossiers et des alias, de copier et de supprimer des fichiers ou des dossiers, de développer la hiérarchie d'un ou plusieurs dossiers et d'ajouter des fonctionnalités (par exemple, créer un lanceur LaunchAnywhere, installer des clients PowerUpdate ou ajouter des actions supportées par une plateforme donnée).

Arbre graphique

L'arbre graphique affiche la hiérarchie des fichiers, des dossiers et des actions de l'installation en cours.

Pour ajouter des fichiers dans l'arbre graphique, cliquez sur **Ajouter des fichiers**. Vous pouvez également faire glisser des fichiers et des dossiers d'une fenêtre d'Explorateur (Windows), du Finder (Mac OS X) et de certains gestionnaires de fichiers UNIX/Linux à l'arborescence des fichiers.

Les objets (fichiers, dossiers, actions) affichés dans l'arbre graphique peuvent être réagencés par glisser-déplacer, à l'aide des flèches Haut, Bas, Droite ou Gauche. La flèche Droite fait remonter l'objet dans le dossier immédiatement supérieur ; la flèche Gauche fait remonter l'objet dans un niveau plus élevé de la hiérarchie.

Des règles peuvent être affectées aux objets présents dans la hiérarchie ; les règles associées à un objet sont évaluées avant installation de celui-ci.

Un clic droit dans l'arbre graphique affiche un menu contextuel dont les commandes permettent d'ajouter, de dupliquer et de supprimer des actions, d'ajouter des fichiers et de spécifier le Magic Folder dans lequel l'objet sélectionné doit être enregistré.

Associer des objets aux fonctions du produit et aux composants

L'arbre graphique montre également les associations établies entre les objets (fichiers, dossiers et actions) et les fonctions du produit ou les composants. Le menu **Affecter à** (partie supérieure de l'arbre graphique) contient les commandes **Fonctions du produit** et **Composants**.

Associer des objets aux fonctions du produit

Pour toute fonction créée à l'aide de la tâche **Structure | Fonctions**, une colonne est ajoutée dans l'affichage **Affecter aux fonctions du produit** de

l'arbre graphique. Les objets affichés (fichiers, dossiers ou actions) sont associés à une case à cocher dans la colonne de chaque fonction.

Pour associer un objet à une fonction, cochez sa case.

Si la case d'un dossier contient une coche, cela signifie que **tous les objets** de ce dossier sont associés à la fonction correspondant à cette case ; si la case contient un tiret (-), cela signifie que **certains objets** du dossier sont associés à cette fonction ; si **aucun objet** du dossier n'est associé à la fonction, la case est alors vide.

La case des fichiers et des actions associés à une fonction contient une coche ; elle est vide dans le cas contraire.

Associer des objets aux composants

La vue **Affecter aux composants** de la tâche **Fichiers** se comporte différemment de la vue **Affecter aux fonctions du produit**. Dans la vue **Affecter à Fonctions du produit** chaque fonction a des cases à cocher carrées. La vue **Affecter à Composant** utilise des boutons radio. Les fichiers ne peuvent être associés qu'à un composant à la fois. Exemple : s'il existe trois composants (Comp1, Comp 2 et Comp 3) et que vous associez un fichier à Comp 1, ce fichier ne peut être associé à aucun des deux autres composants.

Les composants sont associés automatiquement aux fichiers lors de l'ajout de ceux-ci, mais ces associations peuvent être modifiées. Si tous les objets d'un dossier sont associés au **même composant**, la case de ce dossier contient une coche ; si les objets d'un dossier sont associés à des **composants différents**, le dossier est associé à plusieurs composants ; si certains objets d'un dossier sont associés à plusieurs composants, la case de ce dossier contient un tiret (-) ; si **aucun objet** d'un dossier n'est associé à un composant, la case de ce dossier est alors vide.

Définir le classpath

Le bouton **Définir le classpath** désigne les dossiers et les archives qui doivent être déclarés dans le classpath du projet. Si le classpath a été défini auparavant, la commande **Définir le classpath** a priorité sur cette définition. Cette fonction InstallAnywhere peut s'appliquer aux fichiers ZIP/JAR et aux fichiers de classe du projet.

Ajouter un lanceur

Pour ajouter une application LaunchAnywhere, il suffit de cliquer sur le bouton **Ajouter un lanceur** de la tâche **Fichiers**. Cette fonction ajoute une application Java LaunchAnywhere qui facilite le lancement de l'installateur par les utilisateurs (la classe main doit être définie avant d'ajouter l'application Java).

Tâches Pré-installation, Post-installation et Désinstallation

Les tâches Installation et Désinstallation ont une structure identique et disposent des mêmes options. La seule différence qui existe entre ces tâches est le moment de leur exécution lors de l'installation ou de la désinstallation, soit :

- Installateur
 - o Tâche Pré-installation
 - o Installation des fichiers (selon configuration de la tâche **Fichiers**)
 - o Tâche Post-installation
- Désinstallateur
 - o Tâche Pré-désinstallation
 - o Désinstallation des fichiers (selon configuration de la tâche **Fichiers**)
 - o Tâche Post-désinstallation

Dans chaque tâche, les développeurs peuvent ajouter, supprimer ou réagencer les actions. Les actions sont exécutées dans l'ordre où elles apparaissent dans la liste des actions (du haut vers le bas). Les actions peuvent être déplacées dans la liste par glisser-déplacer ou à l'aide des flèches Bas et Haut.

Utilisez ces tâches pour ajouter les actions par panneaux, les actions par consoles et les actions générales nécessaires à l'installateur et au désinstallateur. Lorsqu'elles sont utilisées dans la tâche **Fichiers**, certaines actions générales de ces tâches (par exemple, **Modifier fichier texte**) remplissent une fonction à la fois lors de l'installation et lors de la désinstallation ; par contre, si l'action n'est pas utilisée dans la tâche **Fichiers**, la fonction de désinstallation (annuler les modifications effectuées) ne sera pas activée pour celle-ci.

Construction du projet

La tâche **Construction du projet** propose des options qui permettent de construire un installateur sur plusieurs plates-formes cible. Lorsque la construction est terminée, le sous-panneau **Emplacement de la sortie de construction** apparaît.

Cibles de la construction

La sous-tâche **Cibles de la construction** définit les plates-formes cible pour lesquelles l'installateur doit être construit. Les installateurs peuvent être créés sans VM Java intégrée ou avec une VM Java sélectionnée par les développeurs. Si la construction doit être créée pour différentes plates-formes, il suffit de spécifier celles-ci et de sélectionner les VM Java à leur associer.

Il est possible de créer un installateur pour une version d'UNIX qui ne figure pas explicitement dans la liste des plates-formes.

1. Dans le champ de texte affiché sous l'option **Solaris**, tapez le nom de la version d'UNIX.
2. Dans la liste des VM, sélectionnez la valeur à intégrer avec la nouvelle version d'UNIX (ce champ de texte peut également être utilisé pour établir une distinction entre deux versions du même installateur).

L'option **Télécharger d'autres Packs VM** affiche la zone **Packs VM** du site Web de Zero G. Les Packs VM sont stockés sous forme de fichiers .vm (ZIP/JAR) qui contiennent l'archive d'une VM Java et un fichier vm.properties. Ces Packs VM

doivent être stockés comme ressources dans le dossier
 ../resources/installer_vms. La VM sélectionnée pour intégration sera enregistrée
 en fonction des projets envisagés.

Distribution

Dans la sous-tâche **Distribution**, les développeurs peuvent construire et optimiser un installateur qui sera utilisé via le Web et lancé à partir d'un fichier HTML. Les développeurs peuvent construire et optimiser un installateur sur un ou plusieurs CD-ROM.

La sous-tâche **Distribution** permet également aux développeurs de construire et d'optimiser des modules de fusion et des modèles.

Installateurs Web

Un « installateur Web » est un fichier exécutable qui contient la logique d'installation nécessaire. La construction d'un installateur Web génère également une page HTML et une applet Java intégrée qui facilite le téléchargement de l'installateur via le Web. Sélectionnez l'option **Optimiser la taille de l'installateur selon plate-forme** pour réduire la taille de l'installateur en excluant les ressources spécifiques à la plate-forme (en fonction des règles **Vérifier la plate-forme**). Les développeurs peuvent également sélectionner la langue dans laquelle la page Web cible doit être créée.

Installateurs CD/DVD

Ces installateurs se composent de plusieurs fichiers qui sont gravés sur un ou plusieurs CD ou DVD. S'ils sont trop volumineux pour les CD/DVD, ils peuvent également être stockés sur des volumes en réseau. La sortie de ce type de construction peut être gravée directement sur disque.

Les installateurs CD/DVD peuvent être enregistrés sur plusieurs CD ou DVD. Si la taille de l'installateur dépasse 650 Mo, il est automatiquement segmenté. Pour définir la limite au-delà de laquelle InstallAnywhere doit passer sur un nouveau disque les développeurs peuvent cliquer sur **Changer nom et espace disque** pour définir la capacité et le nom de chaque disque. Le nom ainsi défini sera affiché pendant l'installation, au moment où l'installateur demande l'installation du disque suivant.

Graver un installateur CD

La structure des dossiers d'un installateur CD est la suivante :

```
Disk1/InstallerData.1/
  |- Platform1/
  |- Platform2/
Disk2/ InstallerData.2/
  |- InstallerResources
```

Si l'installateur CD est « optimisé », sa structure de dossiers se présente alors comme suit :

```
Platform1/ Disk1/
  |- InstallerData.1/
```

```
Disk2/  
  |- InstallerData.2/  
Platform2/ Disk1/  
  |- InstallerData.1/  
Disk2/  
  |- InstallerData.2/
```

Cette structure (Disk1, Disk2, etc.) doit être reproduite lors de la gravure sur CD ou DVD : si la gravure est limitée au contenu de ces dossiers, l'installateur ne fonctionnera pas correctement. Pour l'application de gravure sur disque la structure des dossiers doit se présenter comme suit :

```
<Nom ISO du CD>  
  |- Disk1  
  | -..Disk2
```

Modules de fusion

Lors de la phase de conception, InstallAnywhere peut intégrer un ou plusieurs « modules de fusion » dans un projet d'installateur. Pour importer un module, cliquez sur **Importer un module de fusion** et sélectionnez le module de fusion à importer. Vous pouvez ensuite sélectionner les actions qui seront ajoutées à la partie inférieure de la liste des actions.

Journal de construction

La fenêtre **Journal de construction** affiche le journal XML généré lorsque la construction d'un projet d'installateur est terminée. Pour afficher le contenu actuel de ce journal, cliquez sur **Rafraîchir le journal** ; pour le supprimer, cliquez sur **Effacer le journal**.

Menus InstallAnywhere

InstallAnywhere regroupe certaines commandes dans des menus. Les commandes des menus **Fichier** et **Édition** sont assez classiques, mais cette rubrique fournit quelques détails complémentaires.

Menu Fichier

Le menu **Fichier** propose les commandes standard **Ouvrir** et **Enregistrer** et d'autres commandes.

Nouveau

Nouveau permet de lancer un nouveau projet d'installateur. Pour enregistrer et nommer un nouveau projet, cliquez sur le bouton **Enregistrer sous** de la boîte de dialogue **Créer un projet**. La boîte de dialogue **Créer un projet** permet également aux développeurs d'ouvrir un modèle, en cliquant sur **Ouvrir Autre**.

Ouvrir

Ouvrir affiche la boîte de dialogue **Ouvrir un fichier de projet**.

Ouvrir récent

Ouvrir récent est associé à la liste des fichiers de projet InstallAnywhere ouverts en dernier.

Enregistrer

Enregistrer enregistre le projet (le projet étant nommé lors de sa création, cette tâche ne sollicite jamais de nom).

Enregistrer sous

Enregistrer sous permet aux développeurs de renommer le projet actuel (le projet actuel est conservé, mais dans l'état où il se trouvait lors de son dernier enregistrement). Le projet renommé contient toutes les modifications effectuées jusqu'à ce changement de nom et devient le nouveau projet ouvert.

Enregistrer une copie sous

Enregistrer une copie sous permet aux développeurs de renommer le projet : contrairement à **Enregistrer sous**, **Enregistrer une copie sous** crée une copie mais conserve le projet original comme projet ouvert.

Revenir au projet enregistré

Revenir au projet enregistré permet de rouvrir le projet actuel dans l'état où celui-ci se trouvait lors de son dernier enregistrement. Pour enregistrer une version du projet avant de « revenir au projet enregistré », choisissez

Enregistrer une copie sous, spécifiez un nom, puis choisissez **Revenir au projet enregistré**.

Quitter

Quitter ferme le projet actuel et quitte l'environnement de développement InstallAnywhere. Si des modifications ont été effectuées et non enregistrées depuis l'ouverture du projet ou depuis son dernier enregistrement, un message vous le signale et vous donne la possibilité de procéder à un enregistrement.

Menu Édition

Les actions et les règles disposent d'un Presse-papiers distinct. Les développeurs peuvent donc choisir tour à tour les commandes **Couper une règle**, **Copier une action**, **Coller une action** et **Coller une règle** sans constater d'interférence entre les opérations portant sur les actions et les opérations portant sur les règles.

Couper, copier et coller une action

Couper une action supprime l'action de sa tâche et en stocke une copie dans le Presse-papiers, alors que **Copier une action** conserve l'action dans sa tâche et en stocke une copie dans le Presse-papiers. L'action stockée dans le Presse-papiers peut ensuite être collée dans un autre endroit de la tâche active ou dans d'autres tâches à l'aide de la commande **Coller une action**.

Supprimer une action

Supprimer une action supprime définitivement l'action (même résultat qu'avec le bouton **Supprimer**).

Couper, copier et coller une règle

Les commandes **Couper une règle**, **Copier une règle** et **Supprimer une règle** permettent aux développeurs de choisir entre la règle qui est sélectionnée et toutes les règles associées à une action (un menu apparaît à cet effet lorsqu'ils choisissent la commande). **Couper une règle** supprime la ou les règles sélectionnées et en stocke une copie dans le Presse-papiers. **Copier une règle** conserve la ou les règles sélectionnées et en stocke une copie dans le Presse-papiers.

Coller une règle ajoute la ou les règles à la règle existante (ou dans la liste des règles) ou remplace la ou les règles existantes.

Supprimer une règle

Supprimer une règle supprime définitivement la règle sélectionnée ou toutes les règles associées à une action.

Préférences

La boîte de dialogue des préférences permet de personnaliser l'environnement de développement InstallAnywhere : lancer l'Assistant de projet ou Advanced Designer, sélectionner une langue pour Advanced Designer, etc.

Paramètres généraux

Les **Paramètres généraux** permettent aux développeurs de définir les valeurs par défaut d'Advanced Designer : langue d'Advanced Designer, affichage au démarrage d'Advanced Designer et affichage de l'arbre graphique pour la tâche Fichiers, soit vue **Affecter à Fonctions du produit** ou vue **Affecter à Composant**.

Pour changer la langue d'Advanced Designer : dans la liste affichée sous **Langue de création**, sélectionnez la valeur requise, puis quittez et relancez InstallAnywhere.

Les paramètres décrits ci-dessous indiquent à InstallAnywhere le comportement à adopter s'il ne trouve pas les fichiers référencés dans le projet.

Chargement du projet décrit le comportement à appliquer aux fichiers qui sont référencés dans le projet InstallAnywhere (ces fichiers référencés sont ceux qui sont ajoutés dans la tâche **Fichiers**).

Constructions exécutées en ligne de commande précise si une construction exécutée à partir de la ligne de commande doit réussir même si les fichiers référencés sont introuvables.

InstallAnywhere utilise un « fichier de manifeste » pour garder la trace des fichiers utilisés dans l'installateur. Les fichiers de manifeste sont particulièrement utiles lorsque plusieurs groupes de développeurs travaillent sur un projet dans des lieux distincts et qu'un de ces groupes souhaite dresser la liste des fichiers à inclure dans la construction. **Paramètres généraux | Fichier de manifeste** permet aux développeurs d'autoriser la construction si les fichiers référencés dans le fichier de manifeste sont introuvables.

Chemins source

La boîte de dialogue **Chemins source** permet aux développeurs de spécifier des chemins relatifs pour leurs projets d'installateur.

Pour utiliser des chemins source relatifs, cliquez sur **Activer chemins source**.

Pour ajouter un chemin source relatif :

1. Cliquez sur **Ajouter**.
2. Dans le bloc de texte affiché sous le label **Nom du chemin d'accès**, tapez le chemin d'accès (par exemple RESOURCE).
3. Dans le bloc de texte affiché sous le label **Dossier**, tapez le chemin source (par exemple c:\resources).

Mettre à jour InstallAnywhere

InstallAnywhere utilise la fonction PowerUpdate (développée par Zero G) pour rechercher automatiquement les mises à jour disponibles. Par défaut, la fonction PowerUpdate est active, mais même si elle est désactivée, il est possible de rechercher les mises à jour manuellement.

Couleurs d'Advanced Designer

La boîte de dialogue **Couleurs d'Advanced Designer** permet aux développeurs de modifier le jeu de couleurs utilisé dans l'environnement de développement InstallAnywhere.

Enregistrement

Une licence et un numéro de série sont nécessaires pour enregistrer et activer InstallAnywhere en ligne. L'activation transmet à Zero G les informations suivantes : adresse IP, numéro de série et informations d'enregistrement. Aucune autre information n'est transmise à Zero G. Si l'accès à Internet est impossible, il suffit d'enregistrer ces informations dans un fichier et de faire parvenir celui-ci à l'adresse suivante : register@ZeroG.com.

Les versions d'essai proposées par Zero G pour permettre l'évaluation d'InstallAnywhere ne sont pas limitées : elles permettent aux développeurs de tester l'ensemble des fonctionnalités du produit ; toutefois, la durée de vie des installateurs créés à l'aide d'une version d'essai non enregistrée d'InstallAnywhere est limitée à trois jours. Pour construire des installateurs sans restrictions, il suffit d'acheter une licence et d'enregistrer une copie d'InstallAnywhere (y compris versions d'essai). Pour tout renseignement d'ordre commercial, faites parvenir un e-mail à Zero G, à l'adresse sales@ZeroG.com.

Pour enregistrer une copie d'InstallAnywhere, renseignez les champs et cliquez sur **Enregistrer**.

Assistant de projet

Lancer l'Assistant

Lancer l'Assistant lance l'Assistant de projet d'InstallAnywhere et affiche l'écran **Définir les détails**. Le projet s'ouvre lorsque l'action **Lancer l'Assistant** reste ouverte.

Menu Aide

À propos

À propos affiche un écran contenant des informations sur la version du logiciel et les boutons **Préférences** et **Auteurs**.

Rechercher les mises à jour

Rechercher les mises à jour permet de rechercher manuellement (sur le site Web de Zero G) les mises à jour d'InstallAnywhere.

Enregistrer

Enregistrer affiche l'onglet InstallAnywhere **Préférences | Enregistrement** qui contient plusieurs champs d'information à renseigner pour enregistrer le logiciel. Après avoir saisi les informations demandées, il suffit de cliquer sur **Enregistrer** pour les transmettre automatiquement à Zero G. Remarque : la durée de vie des installateurs créés à l'aide d'une copie non enregistrée d'InstallAnywhere est limitée à trois jours.

Aide

La séquence **Aide | Aide InstallAnywhere** affiche le fichier d'aide InstallAnywhere (format HTML) dans le navigateur par défaut du système.

Actions InstallAnywhere

Une « action InstallAnywhere » représente une opération qui doit être exécutée par l'installateur ou par le désinstallateur. Les actions peuvent être de complexité variable : afficher un panneau, installer des fichiers, exécuter du code personnalisé pendant l'installation, etc.

Remarque : les actions doivent être ajoutées dans Advanced Designer. Les actions peuvent être associées aux tâches qui représentent des opérations exécutées en temps réel par l'installateur, soit : Pré-installation, Fichiers, Post-installation, Pré-désinstallation et Post-désinstallation. Pour ajouter une action à un installateur, cliquez sur **Ajouter une action**. Le bouton **Ajouter une action** est disponible dans tous les contextes où les actions sont disponibles ; lorsque vous cliquez sur ce bouton, la boîte de dialogue **Sélectionnez une action** apparaît.

La boîte de dialogue **Sélectionnez une action** présente uniquement les actions qui sont disponibles pour la tâche en cours. Si vous cliquez sur **Ajouter une action** à partir de la tâche **Fichiers**, le système n'affiche pas les actions qui exigent une entrée utilisateur. Dans la tâche **Fichiers**, la boîte de dialogue **Sélectionnez une action** propose un onglet pour les tâches d'installation et un onglet pour les tâches générales ; les actions regroupées dans l'onglet **Général** sont disponibles à partir de toutes les tâches. D'autres actions, liées aux tâches Pré-installation, Post-installation et Désinstallation, sont regroupées dans les onglets **Panneaux**, **Consoles** et **Plug-ins**.

Remarque : l'onglet **Plug-ins** s'affiche dès qu'un plug-in de code personnalisé a été ajouté.

Les actions sont exécutées dans l'ordre où elles apparaissent dans la tâche (du haut vers le bas) ; pour modifier cet ordre, utilisez la flèche Haut ou la flèche Bas. Les flèches Gauche et Droite déplacent les actions de/vers les dossiers.

Lorsqu'une action a été ajoutée, son personnaliseur apparaît dans la partie inférieure d'Advanced Designer ; les développeurs peuvent alors personnaliser cette action en modifiant son personnaliseur. La zone **Propriétés** permet aux développeurs d'associer un emplacement de fichier et des variables à l'action et de décrire les conditions d'exécution de l'action. La zone **Règles** permet aux développeurs d'associer des règles à l'action ; dans ce cas, la lettre **R** apparaît sur l'icône de l'action. Par ailleurs, la présence de la lettre **R** peut faciliter les opérations de test ou de modification d'un installateur comportant un grand nombre d'actions – certaines avec règles et d'autres sans.

Les actions Pré-installation et Désinstallation sont exécutées avant les actions Fichiers, qui sont elles-mêmes exécutées avant les actions Post-installation/Désinstaller. En général, les actions Pré-installation déterminent si l'installation est possible et les éléments à installer sur le système cible. Les actions ajoutées à partir de la tâche **Fichiers** définissent les opérations effectuées sur le système cible, par exemple créer des dossiers, décompresser des archives ou déplacer des fichiers. Les actions Post-installation exigent généralement l'installation de fichiers, par exemple afficher le Readme/LisezMoi ou lancer l'application qui vient d'être installé. **Les actions Pré-installation et Post-installation ne peuvent pas être désinstallées.**

Remarque : les actions Pré-installation et Post-installation sont exécutées plusieurs fois si l'utilisateur clique à plusieurs reprises sur **Précédent** puis sur **Suivant**. Cette situation peut entraîner des erreurs pour les actions qui modifient les fichiers, par exemple **Modifier fichier texte – Fichier unique** ou **Enregistrer un service Windows**. Pour éviter ces erreurs, Zero G conseille d'exécuter les actions de ce type pendant l'installation effective (tâche **Fichiers**).

Actions d'installation

Action	Versions IA	Description
Configurer le registre Windows – Entrée unique	E S	Permet de déplacer (une par une) des clés, des données ou des valeurs présentes dans le registre Windows du système de l'utilisateur.
Configurer le registre Windows – Plusieurs entrées	E	Permet de déplacer (en une seule opération) plusieurs clés, données ou valeurs présentes dans le registre Windows du système de l'utilisateur.
Copier fichier	E	Copier un fichier d'un emplacement à un autre dans le système de l'utilisateur.
Copier un dossier	E	Copier un dossier d'un emplacement à un autre dans le système de l'utilisateur.
Créer alias/lien/raccourci	E S N	Créer un alias (Mac OS X), un lien symbolique (UNIX et Linux) ou un raccourci (Windows).
Créer un désinstallateur	E S N	Créer le désinstallateur et les fichiers associés au désinstallateur (Zero G conseille d'installer le désinstallateur dans son propre dossier).
Créer un dossier	E S N	Créer un dossier dans le système de l'utilisateur. Si un dossier existant est déjà affecté du nom spécifié, il n'est pas remplacé.
Créer un lanceur LaunchAnywhere pour les applications Java	E S N	Créer un lanceur LaunchAnywhere pour exécuter l'application Java installée. Pour plus de détails, consultez la rubrique traitant de LaunchAnywhere , ci-dessous.
Décompresser une archive	E S N	Décompresser un fichier ZIP (.zip, .jar, .war, .ear) ou décoder un fichier MacBinary (.bin) et le copier dans le système de l'utilisateur.
Définir variable d'environnement système	E	Permet de définir des variables d'environnement dans le système de l'utilisateur. Limité à Windows et UNIX ; les shells UNIX Bash, sh, ksh, zsh, csh et tcsh sont supportés.

Déplacer un dossier	E	Transférer un dossier d'un emplacement à un autre (ou renommer un dossier) dans le système de l'utilisateur.
Déplacer un fichier	E	Transférer un fichier d'un emplacement à un autre (ou renommer un fichier) dans le système de l'utilisateur.
Enregistrer un service Windows	E	Démarrer, arrêter ou mettre en pause un service Windows du système de l'utilisateur.
Installer à partir d'un manifeste	E	Installer dans le système de l'utilisateur les dossiers et fichiers spécifiés par le fichier de manifeste. Pour plus de détails, consultez la section traitant des fichiers de manifeste, ci-dessous.
Installer HP-UX Depot	E	Permet d'installer et de désinstaller les fichiers HP-UX Depot. Les fichiers Depot pouvant contenir plusieurs paquetages, les développeurs doivent désigner le paquetage requis. Pour installer plusieurs paquetages dans le même Depot, il est nécessaire de prévoir une action par paquetage (l'installation de plusieurs paquetages n'augmente pas la taille de l'installateur).
Installer Linux RPM	E	Permet d'installer et de désinstaller des paquetages Linux RPM via l'action RPM de Linux. Les RPM peuvent être intégrés à l'installateur ou préexistants dans le système. Si le RPM peut être traduit et que la case Traductible est cochée dans le personnalisateur d'action, le RPM sera installé dans l'emplacement approprié de l'arborescence des fichiers. En outre, le RPM peut être configuré de manière à ignorer les dépendances (résultat semblable à celui qui est obtenu en spécifiant l'option <code>-nodeps</code> dans l'outil RPM en ligne de commande) et à forcer l'installation (cf. <code>-force</code>).
Installer SpeedFolder	E S	Au moment de la construction, récupère les fichiers requis (ces fichiers sont regroupés dans un SpeedFolder). Ces fichiers sont ensuite installés en une seule opération. Conseil – Cette action est conseillée pour les installations faisant intervenir un grand nombre de fichiers et pour les constructions qui s'exécutent automatiquement.
Installer un client PowerUpdate	E S N	Créer un client PowerUpdate qui recherche les mises à jour pour l'application installée (exige un client ZIP téléchargé à partir d'un serveur PowerUpdate).

Installer un fichier	E S N	Installer un fichier de l'installateur dans le système de l'utilisateur. Cette action s'affiche automatiquement chaque fois qu'un fichier est ajouté.
Installer un module de fusion	E	Installer un module de fusion comme s'il était exécuté sous forme d'installateur en mode silencieux.
Installer un paquetage Solaris	E	Permet d'installer et de désinstaller les fichiers d'un paquetage Solaris. Ces paquetages peuvent être intégrés à l'installateur ou préexistants dans le système. (Les développeurs doit spécifier le nom du paquetage.) InstallAnywhere peut intégrer dans l'installateur les fichiers admin/réponse du ou des paquetages sélectionnés. Pour plus de détails sur ces fichiers, consultez les pages de manuel/man pages traitant de pkgadd(1), de pkgask(1) et d'admin(4).
Installer une archive	E S N	Installer un fichier ZIP (.zip, .jar) de l'installateur dans le système de l'utilisateur. Cette action s'affiche automatiquement chaque fois qu'un fichier ZIP est ajouté.
Supprimer un dossier	E	Supprimer un dossier dans le système de l'utilisateur.
Supprimer un fichier	E	Supprimer un fichier dans le système de l'utilisateur.

Actions générales

Action	Versions IA	Description
Affichage d'une boîte de dialogue de messages	E	Affiche une boîte de dialogue modale (message) qui sollicite une entrée utilisateur. Cette boîte de dialogue de message s'affiche par-dessus le panneau affiché et peut être utilisée pour forcer l'utilisateur à réafficher le panneau précédent, à quitter l'installateur ou à saisir les informations demandées. Lorsque cette boîte de dialogue est contrôlée par des règles, elle peut être utilisée comme outil de vérification des données.
Afficher du texte sur la console	E S	Affiche sur la console de débogage le texte spécifié. Cette action est utile pour évaluer l'état d'avancement d'un installateur non interactif en mode silencieux ou en mode Console, ou l'état d'avancement d'une étape non interactive de

l'installation.		
Afficher informations de débogage	E S	Les installateurs InstallAnywhere disposent de fonctions complètes de débogage. En travaillant en « mode Débogage », les développeurs peuvent détecter la plupart des problèmes des installateurs InstallAnywhere. L'action Afficher informations de débogage permet aux développeurs de générer des informations relatives à l'installation (affichage sur la console ou enregistrement dans un fichier). Informations disponibles : table des matières du gestionnaire des variables InstallAnywhere, arborescence de l'installation, propriétés Java, etc.
Ajouter un label de saut	E	Cette action permet de créer une dérivation conditionnelle de l'installation. À l'aide de règles InstallAnywhere, les développeurs peuvent faire « sauter » l'utilisateur vers une opération ultérieure (ou antérieure) de l'installation en fonction des caractéristiques du système cible ou de l'installation envisagée. Utilisez cette action en même temps que l'action Cible du saut (action disponible pendant les tâches de pré-installation et post-installation).
Commentaire	E S N	Permet aux développeurs d'associer un bref commentaire à l'installateur.
Définir une variable InstallAnywhere – Variable unique	E S	Voir « Définir une variable InstallAnywhere – Plusieurs variables ».
Définir variables InstallAnywhere – Plusieurs variables	E S	Cette action et la suivante permettent aux développeurs de spécifier des valeurs pour les variables InstallAnywhere et/ou de créer les variables InstallAnywhere requises. Cette action peut être utilisée dans l'ensemble de l'installation ; elle peut en contrôler la plupart des aspects.
Exécuter code personnalisé	E S	Permet aux développeurs de compléter les fonctionnalités d'InstallAnywhere. L'interface API d'InstallAnywhere étant en Java pur, elle permet aux développeurs d'inclure toute action supportée par Java. L'action Exécuter code personnalisé représente l'interface non interactive de cette API. Pour plus de détails, consultez la rubrique Code personnalisé , ci-dessous.
Exécuter fichier script/batch	E	Permet aux développeurs de saisir le texte d'un fichier de script ou d'un fichier batch qui sera ensuite exécuté lors de l'installation.

Exécuter le fichier cible	E S	Lance un exécutable ou affiche un document inclus dans l'installateur. Si la cible est un document dont le type a été associé à une application, il est ouvert dans celle-ci. Remarque : cette action est disponible pendant la séquence d'installation et dans les fichiers installés.
Exécuter le script Ant	E	Permet aux développeurs d'exécuter les scripts conçus pour l'application Ant du projet Apache Jakarta. Si cette action est sélectionnée, Ant sera intégré à l'application. Cette action doit être utilisée uniquement par les développeurs qui ont l'habitude de travailler avec Ant. Pour plus de détails, consultez les pages suivantes : http://ant.apache.org/ .
Exécuter une commande	E	Permet aux développeurs d'exécuter une commande comme s'ils utilisaient un interpréteur à ligne de commande. Cette action est utile pour exécuter des applications qui sont déjà installées dans le système cible. La commande est entrée comme elle le serait dans la ligne de commande du système.
Exécuter XSL Transform	E	Permet aux développeurs de spécifier une XSL Transform et sa cible. Le dossier <InstallAnywhere>\resource\extras\presets contient des XSL Transform prédéfinies.
Exécuter XSL Transform – Dans l'archive	E	Cette action s'exécute de la même manière que l'action Exécuter XSL Transform , mais elle s'applique aux fichiers d'une archive ; elle est utile pour configurer les applications Web dans des fichiers WAR, EAR et JAR.
Lancer le navigateur par défaut	E S	Permet de lancer le navigateur Web par défaut de l'utilisateur avec les arguments spécifiés par les développeurs. L'action peut ouvrir directement une adresse URL ou un fichier du système. (Action disponible pendant les tâches de pré-installation et post-installation.)
Lecture de l'entrée dans le registre Windows	E	Permet d'accéder aux informations stockées dans le registre Windows. Cette action permet de lire une valeur (ou de vérifier l'existence d'une clé ou d'une valeur) et de la copier dans les variables InstallAnywhere qui seront utilisées lors de l'installation.
Modifier fichier texte – Dans l'archive	E	Les trois actions décrites ci-dessous permettent aux développeurs de modifier les fichiers texte présents dans le système cible de l'installation. Cette action et les deux suivantes permettent de gérer les

		fichiers de configuration ; l'action Plusieurs fichiers permet en outre de changer les fins de ligne de plusieurs fichiers en une seule opération.
Modifier fichier texte – Fichier unique	E	Voir « Modifier fichier texte – Plusieurs fichiers ».
Modifier fichier texte – Plusieurs fichiers	E	Cette action permet aux développeurs de modifier les fichiers texte présents dans le système cible de l'installation. Les trois actions Modifier fichier texte permettent de gérer les fichiers de configuration ; l'action Plusieurs fichiers permet en outre de changer les fins de ligne de plusieurs fichiers en une seule opération.
Rechercher un composant dans le registre	E	Examiner le registre multiplate-forme pour rechercher un composant donné dans le système de destination, déterminer s'il existe plusieurs instances de ce composant et détecter la ou les versions de ce composant ainsi que leur emplacement respectif.
Relancer Windows	E S N	Relance Windows. Attention – Windows redémarre dès que cette action est lancée ; elle doit donc être utilisée avec discernement et seulement en association avec des règles.
Sauter vers la cible	E	Apparentée à l'action Ajouter un label de saut (voir description au début de cette rubrique d'aide), cette action permet aux développeurs de passer directement – en amont ou en aval – à une autre étape de l'installation. Lorsque cette action est gérée par les règles InstallAnywhere, les développeurs disposent d'une méthode conditionnelle pour se déplacer en mode non linéaire dans la procédure d'installation (action disponible pendant la séquence d'installation et dans les fichiers).
Service Windows : démarrer, arrêter, mettre en pause	E	Si l'application dialogue avec un service Windows, l'installateur devra dans certains cas assurer la gestion de ce service. Lorsque l'installateur est exécuté avec des privilèges suffisants, cette action permet à l'installateur d'arrêter, de démarrer ou de mettre en pause les services Windows enregistrés.

Actions par panneaux

Action	Versions IA	Description
Afficher le mot de passe	E	Demande à l'utilisateur d'entrer un mot de passe. Les développeurs peuvent configurer ce panneau de manière à valider le mot de passe entré par l'utilisateur en le comparant à une liste de mots de passe spécifiés au préalable (associant ainsi différents mots de passe à des fonctions différentes), ou simplement en stockant dans une variable le mot de passe entré par l'utilisateur.
Afficher le numéro de série	E	L'action associée à ce panneau permet aux développeurs de doter l'installateur de fonctionnalités de création et de vérification du numéro de série. Les développeurs peuvent implémenter la génération d'un certain nombre de numéros de série pour un certain nombre de produits. Chaque numéro de série peut représenter un produit unique ou un ensemble de produits. Les développeurs peuvent associer des règles à cette action pour assurer la gestion des différents aspects de l'installation, en fonction des droits associés au numéro de série entré par l'utilisateur.
Afficher message	E S	Affiche un message de texte à l'intention de l'utilisateur pendant l'installation (texte défini par les développeurs). Ce message permet par exemple de récapituler les paramètres d'installation sélectionnés par l'utilisateur ou de faciliter la résolution des problèmes impliquant les variables InstallAnywhere.
Contrat de licence	E S	Affiche le texte du Contrat de licence. Pour pouvoir continuer l'installation, l'utilisateur est obligé d'accepter les termes de ce Contrat. Les développeurs peuvent définir l'état par défaut des boutons radio Accepter ou Refuser (activé/non activé) et spécifier le fichier à utiliser comme texte du Contrat de licence. Ce panneau peut également être associé à un ou plusieurs fichiers HTML, ce qui permet aux développeurs de modifier le formatage du texte et de créer des liens avec des documents externes.
Désinstallateur – Introduction		Affiche un texte de présentation de l'installateur.
Désinstallation terminée		Affiche un message indiquant que la désinstallation est terminée.

Entrée utilisateur (avancé)	E	L'action associée à ce panneau permet aux développeurs de solliciter différents types d'entrée de la part de l'utilisateur et de définir plusieurs variables. Ce panneau peut comporter des boutons radio, des cases à cocher, des champs de texte ou/et des menus déroulants.
Entrée utilisateur (simple)	E	Demande à l'utilisateur d'entrer certaines données.
Évaluation de l'espace disque disponible	E S N	Cette action évalue l'espace disque disponible dans le système cible de l'installation, en fonction des valeurs (emplacement d'installation et fonction) sélectionnées par l'utilisateur. Si l'espace disque disponible est insuffisant pour exécuter l'installation, un message demande à l'utilisateur de libérer une certaine valeur d'espace disque ou de sélectionner un autre emplacement d'installation. Remarque : <i>Dans cette situation, les fichiers déjà ajoutés sont installés. Cette action ne figure pas dans la liste.</i>
Installation terminée	E S N	Ce panneau d'action affiche des informations sur l'état d'avancement de l'installation. Sur les systèmes Windows, il apparaît également si un redémarrage est nécessaire. Cette action est disponible seulement lorsque l'installation des fichiers est terminée.
Introduction	E S N	Ce panneau fait partie du projet InstallAnywhere par défaut ; il affiche un texte de présentation de l'installation.
Message défilant	E	Permet aux développeurs de présenter un texte d'une certaine longueur dans un panneau doté de barres de défilement (par exemple, des instructions).
Panneau de code personnalisé	E	L'API de code personnalisé d'InstallAnywhere permet aux développeurs de créer des panneaux personnalisés selon besoin.
Récapitulatif de pré-installation	E S N	Présente les informations recueillies et analysées avant l'installation des fichiers. Permet aux développeurs de personnaliser les informations présentées. Ce panneau fait partie du projet InstallAnywhere par défaut.

Rechercher fichier/dossier	E	Ce panneau implémente une fonction de recherche dont l'objet dépend des spécifications effectuées par les développeurs (recherche limitée à certaines parties du système de fichiers, fichier correspondant à un certain modèle, etc.). L'utilisateur peut également spécifier un modèle de fichier à partir de ce panneau.
Remarque importante	E S N	Ce panneau permet aux développeurs d'afficher un fichier texte ou HTML d'information (sans les boutons radio affichés dans le panneau du Contrat de licence) : ReadMe/LisezMoi, errata, etc.
Sélectionner alias/lien/raccourci	E S N	Ce panneau fait partie du projet InstallAnywhere par défaut ; il permet à l'utilisateur de sélectionner l'emplacement d'installation des raccourcis (Windows), des alias (Mac) ou des liens (UNIX).
Sélectionner la VM Java	E S N	Lorsque ce panneau est implémenté, l'installateur demande à l'utilisateur de sélectionner la Java VM qui sera utilisée pour les lanceurs LaunchAnywhere installés. Les développeurs peuvent préciser le type de VM qui doit être sélectionné par l'utilisateur : l'action associée à ce panneau recherche alors dans le système une VM correspondant à ce type.
Sélectionner le dossier d'installation	E S N	Ce panneau fait partie du projet InstallAnywhere par défaut ; il permet à l'utilisateur de sélectionner le dossier d'installation principal. Ce panneau n'est pas obligatoire : en son absence, l'installateur sélectionne le panneau par défaut (spécifié dans la tâche Projet Plates-formes).
Sélectionner les fonctions à désinstaller		Permet à l'utilisateur de sélectionner les fonctions à désinstaller.
Sélectionner un dossier	E	Lorsque ce panneau est implémenté, l'installateur demande à l'utilisateur de sélectionner un dossier en fonction de certains critères et de le définir comme variable qui peut ensuite être utilisée lors de l'installation.
Sélectionner un fichier	E	Lorsque ce panneau est implémenté, l'installateur demande à l'utilisateur de sélectionner un fichier en fonction de certains critères et de le définir comme variable InstallAnywhere qui peut ensuite être utilisée lors de l'installation.
Sélectionnez un type d'installation	E S	Demande à l'utilisateur de sélectionner un type d'installation (ou les fonctions à installer).

Sélectionner un type de désinstallateur	Permet à l'utilisateur de demander la désinstallation totale ou partielle de l'application.
---	---

Actions par consoles

Les actions par consoles (généralement désignées « consoles ») permettent à l'installateur de solliciter les entrées utilisateur lorsque celui-ci utilise une interface à ligne de commande. Lorsque l'utilisateur sélectionne une installation par consoles, l'installateur utilise des actions par consoles et non des actions par panneaux.

Action	Versions IA	Description
Afficher boîte de dialogue de messages sur la console	E	Affiche une boîte de dialogue de messages à l'intention de l'utilisateur final.
Afficher le mot de passe	E	Demande à l'utilisateur d'entrer un mot de passe.
Afficher le numéro de série	E	Permet aux utilisateurs de générer une liste de numéros de série et de demander à l'utilisateur d'en saisir un.
Afficher message	E	Affiche un message de texte à l'intention de l'utilisateur pendant l'installation (texte défini par les développeurs).
Code personnalisé	E	L'API de code personnalisé d'InstallAnywhere permet aux développeurs de créer des consoles personnalisées selon besoin.
Contrat de licence	E	Affiche le texte du Contrat de licence.
Désinstallateur – Introduction	E	Affiche un texte de présentation de l'installateur.
Désinstallation terminée	E	Affiche un message indiquant que la désinstallation est terminée.
Entrée utilisateur	E	Demande à l'utilisateur d'entrer certaines données.
Installation impossible	E	Cette console doit être affichée lorsqu'un installateur par consoles a généré une erreur.
Installation terminée	E	Cette action affiche des informations sur l'état d'avancement de l'installation (elle est disponible seulement lorsque l'installation des fichiers est terminée).

Prêt pour installation	E	Signale à l'utilisateur que l'installateur est prêt à installer les fichiers.
Récapitulatif de pré-installation	E	Récapitule les informations recueillies et évaluées avant installation des fichiers.
Sélectionner la VM Java	E	Demande à l'utilisateur de sélectionner la VM Java qui sera exécutée avec le ou les lanceurs LaunchAnywhere installés. Les développeurs peuvent préciser le type de VM qui doit être sélectionné par l'utilisateur : le panneau recherche alors dans le système une VM correspondante.
Sélectionner le dossier d'installation	E	Sélectionne l'emplacement primaire d'installation.
Sélectionner le dossier des liens	E	Permet à l'utilisateur de spécifier l'emplacement d'installation des liens symboliques UNIX.
Sélectionner les fonctions à désinstaller	E	Cette console permet à l'utilisateur de sélectionner les fonctions qu'il souhaite désinstaller.
Sélectionner un type d'installation	E	Demande à l'utilisateur de sélectionner un type d'installation (ou les fonctions à installer).
Sélectionner un type de désinstallateur	E	Permet à l'utilisateur de demander la désinstallation totale ou partielle de l'application.

Propriétés communes

La liste ci-dessous présente les propriétés communes aux différents personnaliseurs d'action disponibles dans InstallAnywhere.

Propriété	Description
Afficher boîte de dialogue non précisée	Affiche une jauge de progression non précisée à l'intention de l'utilisateur pendant l'exécution d'un processus externe.
Chemin	Indique le chemin d'installation de l'action.
Commentaire	Indique le nom de l'action dans l'arborescence graphique.
Dans classpath	Déclare l'objet dans le classpath (pour tous les exécutables LaunchAnywhere installés).

Enregistrer le code de sortie du processus dans	Affecte à la variable InstallAnywhere la valeur du code de sortie du processus.
Enregistrer les données stderr du processus dans	Affecte à la variable InstallAnywhere la valeur des données stderr du processus.
Enregistrer les données stdout du processus dans	Affecte à la variable InstallAnywhere la valeur des données stdout du processus.
Fichier à installer/Fichier existant	Détermine si le fichier doit être installé ou s'il existe déjà dans le système de l'utilisateur.
Interrompre l'installation jusqu'à ce que le processus soit terminé	Met l'installateur en pause jusqu'à ce que l'exécution du processus lancé se termine.
Ne pas désinstaller	Demande à une action de ne pas défaire les résultats d'une autre action exécutée lors de la désinstallation.
Outrepasser les autorisations UNIX / Mac OS X par défaut	Fixe les autorisations du fichier à une valeur spécifique pour cette action.
Si le fichier existe déjà sur le système de l'utilisateur	Ignore le comportement par défaut en ce qui concerne la résolution des conflits entre les fichiers en cours d'installation et les fichiers existants.
Source	Affiche le chemin de l'élément dans le système des développeurs (si les chemins source sont utilisés, affiche le chemin source).

Actions par panneaux – Paramètres

Les actions par panneaux (généralement désignées « panneaux ») permettent de solliciter les entrées utilisateur via une interface graphique/GUI.

Les installateurs à interface graphique ont la possibilité d'indiquer les étapes d'installation en cours à l'aide de « labels » ou d'images. Lorsque l'option **Images** est sélectionnée via la séquence **Interface d'installation | Params d'affichage | Additions du panneau d'installateur | Type d'addition des panneaux de l'installateur**, le personnalisateur associé au panneau des tâches **Pré-installation** et **Post-installation** active l'onglet **Paramètres d'image**. Si l'option **Liste des étapes d'installateur** est sélectionnée, l'onglet **Paramètres des labels** est activé.

Remarque : ces paramètres ne sont pas disponibles pour les actions par panneaux du désinstallateur ; celles-ci utilisent les valeurs par défaut définies dans la tâche **Interface d'installation | Params d'affichage**.

Paramètres d'image

Les paramètres d'image permettent de sélectionner l'image à associer à un panneau donné. Dans un panneau donné, les développeurs peuvent utiliser l'image par défaut, lui associer une image spécifique ou n'afficher aucune image.

Paramètres des labels

L'onglet **Paramètres des labels** du personnalisateur permet aux développeurs de prévisualiser les labels et les icônes. Les labels sont contrastés et marqués pendant l'installation. La construction de l'installateur remplit la liste en fonction des titres des panneaux.

Remarques :

- À partir des onglets **Additions du panneau d'installateur** et **Paramètres des labels** de la tâche **Interface d'installation | Params d'affichage** – qui figurent dans le personnalisateur de chaque panneau –, les développeurs peuvent associer le même label à plusieurs panneaux, ce qui permet de simplifier l'interface (par exemple, si l'installation exige un grand nombre d'étapes, ou si l'installateur affiche plusieurs panneaux pour la même étape).
- Dans l'onglet **Additions du panneau d'installateur** de la tâche **Interface d'installation | Params d'affichage**, les éléments d'interface regroupés à droite de la liste des panneaux permettent de contrôler l'ordre des labels (flèches), d'éditer le texte d'un label (bouton **Éditer le label**), etc.

Aide

En cochant la case **Activer l'aide de l'installateur** (sous-tâche **Interface d'installation | Aide**), l'utilisateur peut disposer d'un complément d'information sur l'installateur. Dans la fenêtre inférieure, les développeurs peuvent saisir un message standard (qui sera le même pour tous les panneaux) ou un texte différent pour chaque panneau. Pour adapter les informations d'aide à chaque panneau d'installateur, activez l'option **Utiliser un texte d'aide différent pour chaque panneau**. Les textes d'aide personnalisés doivent être saisis dans l'onglet **Aide** du personnalisateur d'action (partie inférieure des tâches **Pré-installation** et **Post-installation**).

En activant l'option **HTML**, il est possible de modifier le formatage des messages d'aide (mais non leur barre de titre). Pour changer le formatage, utilisez des balises HTML standard, par exemple :

```
<B>Aide traduite par Pierre BLANC</B>
```

pour afficher cette chaîne en caractères gras :

Aide traduite par Pierre BLANC

LaunchAnywhere et fichiers LAX

Un « exécutable LaunchAnywhere » est un fichier à extension LAX qui lance une application Java sur une plate-forme supportant LaunchAnywhere (UNIX, Mac OS X et toutes versions de Windows). Avec LaunchAnywhere, les utilisateurs peuvent ainsi lancer une application Java en double-cliquant sur une icône (Windows/Mac OS X) ou en entrant une seule commande (UNIX). Le fichier LAX assure également la

configuration de l'environnement de l'application Java, en particulier : définition du classpath, redirection des données stdout et stderr et transmission des propriétés système, des variables d'environnement et des paramètres de ligne de commande.

Pour déterminer les conditions de son exécution, le lanceur examine un fichier de configuration de type `<MyLauncherName>.lax`. Ce fichier LAX est créé pendant l'installation, et il réside dans le même emplacement que le lanceur.

Propriétés du fichier LAX

Les propriétés modifiables par LaunchAnywhere sont les suivantes :

Propriété	Définition
<code>lax.application.name</code>	Nom de l'application exécutée par le lanceur
<code>lax.class.path</code>	Classpath de l'application (par défaut, <code>\$IA_CLASSPATH\$</code> , le classpath spécifié dans l'environnement Advanced Designer d'InstallAnywhere). La spécification du classpath peut utiliser des barres obliques standard (/) ou inverses (\) comme séparateurs de dossiers, et des signes deux-points (:) ou points-virgule (;) comme séparateurs de chemin : InstallAnywhere remplace automatiquement ces caractères en fonction de la plate-forme cible de l'installation. Remarque : InstallAnywhere considérant les signes deux-points (:) comme des séparateurs de chemin, les utilisateurs Mac doivent séparer les dossiers par des barres obliques standard (/) et non par des deux-points.
<code>lax.command.line.args</code>	Liste des arguments transmis à la méthode main de l'application. Ces arguments sont spécifiés exactement de la même manière que sur la ligne de commande. Par exemple, si l'application est appelée à l'aide de la commande "java myApp arg1 arg2", cette propriété doit être affectée de la valeur "arg1 arg2" (les guillemets sont obligatoires autour des arguments qui contiennent des espaces). S'il est nécessaire de transmettre un argument qui est connu seulement au moment de l'installation (par exemple, le nom du dossier d'installation), il est conseillé d'utiliser une variable InstallAnywhere.

<code>lax.java.compiler</code>	(Pour exécution uniquement – Ne peut pas être défini via la commande Éditer les propriétés) Compileur JIT utilisé pour l'exécution de cette application.
<code>lax.main.method</code>	Nom de la méthode initiale de l'application qui doit être appelée par cet exécutable LaunchAnywhere.
<code>lax.mainclass</code>	Classe lancée par cet exécutable LaunchAnywhere. Cette classe doit contenir une méthode dont le nom est défini par la propriété <code>lax.main.method</code> .
<code>lax.nl.current.vm</code>	Chemin de l'exécutable de VM utilisé par cette application.
<code>lax.nl.env.<lowercase></code>	[Windows et UNIX uniquement] – Pour consulter l'une des variables d'environnement du système (par exemple, PATH via <code>\$lax.nl.env.path\$</code>), spécifiez le nom de la propriété sous la forme d'une chaîne utilisant uniquement des minuscules. La valeur de ces propriétés est établie au moment du lancement de l'application, lorsque LaunchAnywhere s'exécute. Les variables LaunchAnywhere permettent également de consulter ces variables d'environnement.
<code>lax.nl.env.<UPPERCASE></code>	[Windows et UNIX uniquement] – Pour consulter l'une des variables d'environnement du système (par exemple, PATH via <code>\$lax.nl.env.PATH\$</code>), spécifiez le nom de la propriété sous la forme d'une chaîne utilisant uniquement des majuscules. La valeur de ces propriétés est établie au moment du lancement de l'application, lorsque LaunchAnywhere s'exécute. Les variables LaunchAnywhere permettent également de consulter ces variables d'environnement.
<code>lax.nl.env.exact_case</code>	[Windows et UNIX uniquement] – Pour consulter l'une des variables d'environnement du système (par exemple, PATH via <code>\$lax.nl.env.exact_case.Path\$</code>), spécifiez le

	nom de la propriété telle qu'elle a été définie dans l'environnement, soit en respectant la hauteur de casse de ses différents caractères. La valeur de ces propriétés est établie au moment du lancement de l'application, lorsque LaunchAnywhere s'exécute. Les variables LaunchAnywhere permettent également de consulter ces variables d'environnement.
<code>lax.nl.env.path</code>	(Pour exécution uniquement – Ne peut pas être défini via la commande Éditer les propriétés) Variable PATH de l'ordinateur sur lequel cette application s'exécute.
<code>lax.nl.java.compiler</code>	Cette propriété définit le nom du compilateur JIT qui doit être utilisé par votre application. Si vous n'affectez pas de valeur à cette option (valeur vide), le système utilise le compilateur JIT par défaut. Vous pouvez également spécifier le nom d'un compilateur JIT spécifique en le définissant dans cette propriété. Si vous n'utilisez pas de compilateur JIT, cette propriété doit être désactivée (valeur Off).
<code>lax.nl.java.launcher.main.class</code>	Classe qui contient la méthode main appelée par LaunchAnywhere.
<code>lax.nl.java.launcher.main.method</code>	Nom de la méthode main appelée par LaunchAnywhere.
<code>lax.nl.java.option.additional</code>	La valeur de cette propriété sera copiée textuellement sur la ligne de commande. Les propriétés/paramètres de la VM Java qui ne sont pas directement supportées par les propriétés du fichier de configuration LAX utilisé peuvent être incluses dans la ligne de commande utilisée pour appeler Java.
<code>lax.nl.java.option.check.source</code>	Est activée (valeur On) pour demander à la VM de vérifier le pseudocode (bytecode), ou désactivée (valeur Off) dans le cas contraire.
<code>lax.nl.java.option.debugging</code>	Active la fonction de débogage de la VM pour permettre le débogage d'une application.

<code>lax.nl.java.option.garbage.collection.background.thread</code>	Les développeurs peuvent prévoir un thread en tâche de fond à faible priorité qui assure la récupération de mémoire (garbage collection). Prend la valeur On ou Off .
<code>lax.nl.java.option.garbage.collection.extent</code>	Peut prendre l'une des valeurs suivantes : min: Récupération de mémoire intégrale (sauf classes). full: Récupération de mémoire intégrale.
<code>lax.nl.java.option.java.heap.size.initial</code>	Définit la taille initiale de la mémoire « heap » mise à la disposition de l'installateur qui sera appelé. Cette valeur est toujours spécifiée en octets, et non en ko ou en Mo ; elle est semblable au paramètre de VM -ms ou Xms. Valeur par défaut : 16777216 (16 Mo).
<code>lax.nl.java.option.java.heap.size.max</code>	Définit la taille maximale de la mémoire « heap » mise à la disposition de l'installateur qui sera appelé. Cette valeur est toujours spécifiée en octets, et non en ko ou en Mo ; elle est semblable au paramètre de VM -mx ou Xmx. Valeur par défaut : 50331648 (48 Mo).
<code>lax.nl.java.option.verbose</code>	Peut prendre l'une des valeurs suivantes : gc: Affiche les messages de récupération de mémoire (garbage collection). normal: Affiche les messages détaillés (verbose) normaux. all: Affiche les messages normaux et les messages de récupération de mémoire (garbage collection). none: N'affiche aucun message détaillé (verbose).
<code>lax.nl.java.option.verify.mode</code>	Définit la vérification effectuée par Java en matière de sécurité et d'erreurs. Valeurs possibles : À distance , Tout ou Néant .
<code>lax.nl.message.vm.not.loaded</code>	Message affiché pour indiquer à l'utilisateur qu'aucune VM n'a été détectée.

<code>lax.nl.valid.vm.list</code>	Liste des VM que cet exécutable LaunchAnywhere pourra utiliser pour l'exécution de l'application Java. Valeurs possibles (si plusieurs valeurs sont spécifiées, elles doivent être séparées par un espace) : ALL (Toute VM) JDK (Tout JDK Java) JRE (Tout JRE Java) J1 (Toute VM Java 1) J2 (Toute VM Java 2) JRE_J1 (Tout JRE Java 1) JDK_J1 (Tout JDK Java 1) JRE_J2 (Tout JRE Java 2) JDK_J2 (Tout JDK Java 2) MSJ (VM Microsoft [jview]) Par ailleurs, la valeur de cette propriété a priorité sur la valeur de VM spécifiée par <code>lax.nl.current.vm</code> si cette VM n'est pas de type correct. L'ordre de la liste des VM correctes spécifie la priorité de sélection des VM détectées dans le système si aucune VM correcte ne figure dans <code>lax.nl.current.vm</code>.
<code>lax.nl.win32.microsoftvm.min.version</code>	Version minimale de la VM Microsoft (jview) qui sera détectée par ce lanceur LaunchAnywhere.
<code>lax.resource.dir</code>	Nom de la plate-forme (en respectant la hauteur de casse exacte des différents caractères de ce nom).
<code>lax.root.install.dir</code>	Dossier racine/root de l'installation (même valeur que <code>\$USER_INSTALL_DIR\$</code>).
<code>lax.stderr .redirect</code>	Emplacement de la sortie stderr de votre application. Valeurs possibles : vide (aucun affichage), Console (affichage dans une fenêtre de console) ou nom de fichier (enregistrement des données dans un fichier). Remarque : si vous spécifiez le chemin d'un fichier Windows, les séparateurs doivent être des barres obliques inverses doubles (\\ – par exemple, <code>c:\\myfolder\\output.txt</code>).
<code>lax.stdout .redirect</code>	Emplacement de la sortie stdout de votre application. Valeurs possibles : vide (aucun affichage), Console (affichage

	dans une fenêtre de la console) ou nom de fichier (enregistrement des données dans un fichier). Remarque : si vous spécifiez le chemin d'un fichier Windows, les séparateurs doivent être des barres obliques inverses doubles (\\ – par exemple, c:\\myfolder\\output.txt).
<code>lax.user.dir</code>	Dossier utilisateur pour votre application (représenté par "."). Vous pouvez également spécifier un chemin absolu ou relatif de votre choix (les chemins relatifs doivent être spécifiés par rapport au lanceur).
<code>lax.version</code>	Numéro de version du lanceur LaunchAnywhere.
<code>LISTPROPS</code>	Cette propriété (qui peut prendre toute valeur) affiche la liste des propriétés système utilisables par l'application Java. Pour afficher les résultats de cette sortie, il est nécessaire de rediriger les données Stdout et stderr.

Format des fichiers de manifeste

Le « fichier de manifeste » est un fichier texte qui contient une liste de dossiers et de fichiers et dont la spécification doit respecter un certain format (voir ci-dessous). Cette spécification précise la source du fichier et sa destination (chemin relatif par rapport à l'emplacement de l'action dans l'Arbre graphique de la tâche Fichiers) ; le cas échéant, elle définit les autorisations UNIX requises et elle précise si le fichier doit être déclaré dans le classpath. Au moment de la construction, ce fichier est analysé et son contenu est importé dans l'installateur.

Format des fichiers de manifeste

Spécification des fichiers :

F,[chemin_source]chemin_relatif_fichier_source,./chemin_relatif_fichier_destination

F,chemin_absolu_fichier_source,./chemin_relatif_fichier_destination

Pour déclarer un fichier dans le classpath :

F,chemin_absolu_fichier_source,./chemin_relatif_fichier_destination,cp

Pour définir les autorisations d'un fichier sous UNIX :

F,[chemin_source]chemin_relatif_fichier_source,./chemin_relatif_fichier_destination,755

Spécification des dossiers :

D,[chemin_source]chemin_relatif_dossier_source[/],./chemin_relatif_dossier_destination[/]

D,chemin_absolu_dossier_source[/],./chemin_relatif_dossier_destination[/]

Exemples :

F,\$IA_HOME\$/chemin/vers/fichier_source.txt,./chemin/destination/Ce_fichier.txt

F,/chemin/absolu/vers/fichier_source.txt,./chemin/destination/Ce_fichier.txt,cp,655

D,\$IA_HOME\$/chemin/vers/dossier,./chemin/destination/dossier

D,/chemin/absolu/vers/dossier,./chemin/destination/dossier

Règles

Les règles peuvent être configurées et associées à la plupart des éléments de l'installateur : actions, composants, fonctions – ou l'installateur lui-même. Les règles renvoient la valeur « true » ou « false », en fonction de l'état de l'ordinateur sur lequel l'installateur s'exécute. Exemple : la règle **Vérifier la plate-forme** peut être configurée de manière à renvoyer la valeur « true » seulement si l'installateur est exécuté sur plate-forme Solaris. Cette règle peut ensuite être associée à un fichier binaire supporté par Solaris et non par Windows ou par Mac OS X.

Les règles peuvent être évaluées à l'aide de « AND/ET » ou de « OR/OU ». **AND** évalue toutes les règles associées à l'action considérée. Si vous souhaitez que les règles soient conditionnelles entre elles, en fonction de la valeur qu'elles renvoient, choisissez **OR** ; dans ce cas, il est impératif de ranger les règles dans l'ordre approprié.

Conseil – Lorsqu'une règle est associée à une action, elle s'exécute avant celle-ci ; Par conséquent, la vérification des valeurs renvoyées dans une action doit être effectuée avec une action qui s'exécute ultérieurement.

Règle	Versions IA	Description
Comparer avec l'horodate de modification de fichier	E	Cette règle permet aux développeurs de comparer l'horodate d'un fichier cible existant avec celle d'un fichier à installer afin de déterminer si le fichier existant doit être écrasé.
Comparer les variables InstallAnywhere	E S	Cette règle permet aux développeurs de procéder à une comparaison de chaînes entre plusieurs variables InstallAnywhere, soit déterminer si une variable est égale/non égale à la valeur ou contient/ne contient pas celle-ci.
Correspond à l'expression régulière	E	Cette règle permet aux développeurs de comparer une chaîne ou une variable InstallAnywhere à une expression régulière de leur choix. Les expressions régulières (regex) constituent une méthode standard pour exprimer une chaîne variable. Le Web permet de consulter des informations sur les expressions régulières, y compris archives et applications permettant de vérifier la validité d'une expression.
Évaluer une règle personnalisée	E	Les règles personnalisées sont définies à partir des spécifications de l'API InstallAnywhere et peuvent être adaptées aux besoins de l'installation.
Vérifier la langue sélectionnée par l'utilisateur	E S	Cette règle permet aux développeurs de prendre des décisions d'installation en fonction de la locale sélectionnée par l'utilisateur lors de l'installation.

Vérifier la plate-forme	E S N	Cette règle permet aux développeurs de spécifier l'action à exécuter ou les fichiers à installer seulement sur une certaine plate-forme. La plate-forme est déterminée par la VM Java et signalée à l'installateur.
Vérifier les attributs des fichiers/dossiers	E	Cette règle permet aux développeurs de vérifier les attributs d'un fichier ou d'un dossier qui existe déjà sur le système cible. Elle permet de déterminer si l'objet existe, s'il s'agit d'un fichier ou d'un dossier et s'il peut être consulté en lecture et/ou modifié.
Vérifier si le fichier/dossier existe	E S	Cette règle s'applique à des actions d'installation portant sur un fichier/dossier à la fois. Elle vérifie si le fichier/dossier auquel elle est associée existe dans l'emplacement d'installation spécifié. Les développeurs peuvent ensuite décider de procéder à l'installation même si le fichier/dossier n'existe pas dans cet emplacement.

Variables InstallAnywhere

À la fin de l'installation, la plupart des variables InstallAnywhere sont enregistrées dans le désinstallateur. Ces variables sont disponibles pour faciliter les actions du désinstallateur : elles décrivent l'état dans lequel se trouve l'installateur à la fin de son exécution.

Syntaxe des variables InstallAnywhere

Les variables sont désignées par un nom (NAME). Par exemple, `USER_INSTALL_DIR` est la variable qui contient le dossier sélectionné par l'utilisateur pour l'installation des fichiers. Toutefois, les développeurs peuvent définir `USER_INSTALL_DIR` différemment s'ils souhaitent changer l'emplacement par défaut d'installation des fichiers. Les variables InstallAnywhere se présentent sous la forme `$NAME$`. Pour déterminer la valeur d'une variable, utilisez la notation `$NAME$`, par exemple `$USER_INSTALL_DIR$` (le caractère \$ signifie « Insérer la valeur de la variable ici. »). Pour simplifier les descriptions, la notation \$ sera utilisée dans l'ensemble de cette documentation.

Définir les variables InstallAnywhere

Les variables InstallAnywhere peuvent être définies de différentes manières :

- Variables fournies par la VM Java – Les installateurs InstallAnywhere ont accès à l'ensemble des propriétés de la VM Java, qui sont stockées dans `$prop.PROPERTY_NAME$`, `PROPERTY_NAME` étant le nom de la propriété Java.
- Variables importées de l'environnement – Dans les systèmes Windows et UNIX, LaunchAnywhere importe les variables d'environnement. Les valeurs de ces variables sont stockées dans `$lax.nl.env.VARIABLE_NAME$` (par exemple, dans les systèmes UNIX, `$lax.nl.env.PATH$` aurait la même valeur que la variable d'environnement `PATH`). Ces variables sont importées au lancement de l'installateur, et elles ne sont pas modifiables. Pour définir des variables d'environnement, utilisez l'action **Définir variable d'environnement système**.
- Variables définies par des actions InstallAnywhere – Les actions **Définir variable – Variable unique** et **Définir variable – Plusieurs variables** affectent des valeurs spécifiques aux variables InstallAnywhere.
- Variables définies par entrée utilisateur – La plupart des actions par panneaux et des actions par consoles définissent des variables InstallAnywhere.
- Variables définies pendant l'exécution de l'installateur ou du désinstallateur – Pendant l'exécution de l'installateur et du désinstallateur, certaines valeurs sont définies ou mises à jour (par exemple, `$INSTALL_STATUS$`).
- Variables définies par code personnalisé

Vérifier la valeur d'une variable InstallAnywhere

La valeur de certaines variables peut changer pendant l'exécution de l'installateur. En ce qui concerne les variables prédéfinies, le point de changement de leur valeur est indiqué dans le tableau « Variables standard ». Pour déterminer la valeur de ces variables pendant l'installation, la meilleure solution consiste à exécuter les actions **Afficher du texte sur la console** et **Afficher informations de débogage** et à diriger les informations de débogage vers la console ou vers un fichier texte. Pour afficher une variable et sa valeur, les développeurs disposent également du panneau **Afficher message**.

Variables InstallAnywhere standard

InstallAnywhere dispose de plusieurs variables par défaut (ou « variables standard ») qui contiennent les valeurs nécessaires à l'installation.

Les variables InstallAnywhere standard sont les suivantes :

Variable	Définition	État
\$/\$ ou \\$\\$	Ces formes de variable sont utilisées comme séparateurs de fichiers (selon plate-forme) ; elles permettent de spécifier des chemins sans tenir compte de la plate-forme. Elles prennent la même valeur que le séparateur utilisé pour les fichiers de propriétés Java.	Ces variables sont en lecture seule.
;\$ ou \$:\$	Séparateur de chemin (selon plate-forme).	Cette variable est en lecture seule.
\$CHOSEN_DIALOG_BUTTON\$	Cette variable est affectée de la valeur sélectionnée par l'utilisateur dans l'action Affichage des messages : si l'utilisateur clique sur le bouton 1, cette variable aura la valeur 1.	
\$CHOSEN_INSTALL_BUNDLE_<#>\$_<#>\$	Si l'option Sélectionnez les fonctions requises est activée, il existera autant de variables que de fonctions – le nombre de fonctions étant spécifié par la variable \$CHOSEN_INSTALL_BUNDLE_NUM\$. Cette variable contient le nom court d'une fonction à installer. Exemple : si \$CHOSEN_INSTALL_BUNDLE_NUM\$ a la valeur 2, il existera deux variables – respectivement \$CHOSEN_INSTALL_BUNDLE_1\$ et \$CHOSEN_INSTALL_BUNDLE_2\$.	Cette variable est en lecture seule.

\$CHOSEN_INSTALL_BUNDLE_LIST\$	Cette variable contient la liste (noms courts, séparés par une virgule) des fonctions sélectionnées pour l'installation.	Cette variable est en lecture seule.
\$CHOSEN_INSTALL_BUNDLE_NUM\$	Si l'option Sélectionnez les fonctions requises est activée, cette variable contiendra le nombre total de composants (sous forme de chaîne) sélectionnés pour installation par l'utilisateur.	Cette variable est en lecture seule.
\$CHOSEN_INSTALL_SET\$	Si l'option Sélectionnez les fonctions requises est activée, cette variable contiendra le nom court de la fonction sélectionnée par l'utilisateur. Exemple : si l'utilisateur décide de personnaliser l'installation, cette variable contiendra la chaîne CUSTOM.	Cette variable peut être définie avant installation.
\$CMD_LINE_ARGUMENTS\$	Variable InstallAnywhere spéciale dont la valeur est déterminée par le lanceur et non par l'installateur. Si cette variable est présente dans la propriété LAX lax.command.line.args, elle prend la valeur des arguments transmis à l'exécutable LaunchAnywhere.	Cette variable est utilisée par LaunchAnywhere.
\$COMMA\$	Cette variable prend la valeur de la virgule (,).	Cette variable est en lecture seule.
\$DEVELOPER_DISK_SPACE_ADDITIONAL\$	Cette variable spécifie une valeur complémentaire arbitraire sous la forme d'une chaîne représentant les octets ajoutés par l'action Évaluation de l'espace disque disponible à l'espace disque calculé comme nécessaire à l'installation. Par défaut, cette variable a la valeur zéro.	Les développeurs peuvent affecter une valeur à cette variable.
\$DOLLAR\$	Cette variable prend la valeur \$.	Cette variable est en lecture seule.
\$EMPTY_STRING\$	Cette variable contient la valeur vide (null) ; elle permet de déterminer si des variables ont été initialisées. Les variables qui n'ont pas encore été initialisées contiennent la valeur vide (null).	Cette variable est en lecture seule.

\$EXTRACTOR_DIR\$	Chemin complet du dossier contenant l'exécutable à auto-extraction (dossier à partir duquel l'installateur est lancé).	Cette variable est en lecture seule.
\$EXTRACTOR_EXECUTABLE\$	Chemin complet du dossier contenant l'exécutable à auto-extraction (dossier à partir duquel l'installateur est lancé).	Cette variable est en lecture seule.
\$FREE_DISK_SPACE_BYTES\$	Chaîne indiquant l'espace disponible en octets sur le volume de destination de l'installation (soit le résultat de l'action Évaluation de l'espace disque disponible). Cette variable reçoit sa valeur immédiatement avant l'installation des fichiers ou/et des dossiers listés dans la tâche Installation .	Cette variable est en lecture seule.
\$IA_CLASSPATH\$	Classpath spécifié dans Advanced Designer.	Les développeurs peuvent définir cette variable dans Advanced Designer.
\$IA_INSTALL_LOG\$	Lorsque cette variable reçoit la valeur appropriée, elle génère le journal d'installation (fichier XML) et l'enregistre dans le dossier \$USER_INSTALL_DIR\$. Ce journal contient des informations sur l'installation et des messages d'erreur ou d'avertissement.	Cette variable est en lecture seule.
\$INSTALL_LOG_DESTINATION\$	La génération du journal étant la dernière action de toute installation, cette variable peut être définie à tout moment pendant les phases de pré-installation, installation ou post-installation. L'utilisateur peut spécifier l'emplacement du journal d'installation.	Les développeurs peuvent affecter une valeur à cette variable.
\$INSTALL_SUCCESS\$	Cette variable indique à l'utilisateur si l'installation a réussi ou échoué ; elle peut prendre l'une des valeurs suivantes : SUCCESS, WARNING, NONFATAL_ERROR ou FATAL_ERROR.	Cette variable est en lecture seule.
\$INSTALLER_LAUNCH_DIR\$	Chemin complet de l'exécutable à auto-extraction de l'installateur.	Cette variable est en lecture seule.

<code>\$INSTALLER_LOCALE\$</code>	Locale sous forme de chaîne (voir <code>java.util.Locale.toString()</code>) sélectionnée par l'utilisateur au début de l'installation.	Cette variable est en lecture seule.
<code>\$INSTALLER_TEMP_DIR\$</code>	Chemin du dossier Temp qui sera utilisé par l'installateur (ce dossier est supprimé lorsque son utilisation est terminée – à condition qu'aucun de ses objets ne soit en service au moment de cette suppression).	Cette variable est en lecture seule.
<code>\$INSTALLER_TITLE\$</code>	Nom de l'installateur (affiché dans la barre de titre).	Les développeurs peuvent affecter une valeur à cette variable.
<code>\$INSTALLER_UI\$</code>	Cette variable prend la valeur de mode d'UI de l'installateur.	Cette variable est en lecture seule, mais les développeurs peuvent lui affecter une valeur au début de l'installation.
<code>\$JAVA_DOT_HOME\$</code>	Cette variable contient la valeur renvoyée par la propriété Java <code>java.home</code> .	Cette variable est en lecture seule.
<code>\$JAVA_EXECUTABLE\$</code>	Chemin de l'exécutable Java sélectionné.	Cette variable est en lecture seule.
<code>\$JAVA_HOME\$</code>	Dossier racine/root d'un installateur Java.	Cette variable est en lecture seule.
<code>\$JDK_HOME\$</code>	Chemin du dossier racine/root d'un installateur à JDK. Cette variable est définie seulement si la VM sélectionnée est un JDK ; sinon, la variable est vide.	Cette variable est en lecture seule.
<code>\$lax.nl.env.[ENVIRONNEMENT VARIABLE NAME]\$</code>	[Windows et UNIX uniquement] – Permet de consulter la valeur des variables d'environnement du système (par exemple, <code>PATH</code> via <code>\$lax.nl.env.PATH\$</code>), spécifiez le nom de la variable sous la forme d'une chaîne utilisant uniquement des majuscules. La valeur de ces propriétés est établie au moment du lancement de l'application, lorsque LaunchAnywhere s'exécute. Les développeurs peuvent également	Cette variable est en lecture seule.

	accéder aux variables d'environnement système via les propriétés LaunchAnywhere.	
\$lax.nl.env.exact_case [Nom_variable_environnement]\$	[Windows et UNIX uniquement] – Permet de consulter la valeur des variables d'environnement du système (par exemple, PATH via \$lax.nl.env.exact_case.Path\$), spécifiez le nom de la variable telle qu'elle a été définie dans l'environnement, soit en respectant la hauteur de casse de ses différents caractères. La valeur de ces propriétés est établie au moment du lancement de l'application, lorsque LaunchAnywhere s'exécute. Les développeurs peuvent également accéder aux variables d'environnement système via les propriétés LaunchAnywhere.	Cette variable est en lecture seule.
\$NULL\$	Dans cette version d'IA, même valeur que \$EMPTY_STRING\$.	Cette variable est en lecture seule.
\$PRODUCT_ID\$	Prend la valeur de l'ID produit spécifié dans la tâche Projet Description .	Cette variable est en lecture seule.
\$PRODUCT_NAME\$	Nom du produit.	Les développeurs peuvent affecter une valeur à cette variable.
\$PRODUCT_VERSION_NUMBER\$	Prend la valeur de la version de produit spécifiée dans la tâche Projet Description .	Cette variable est en lecture seule.
\$PROMPT_USER_CHOSEN_OPTION\$	Cette variable est affectée de la valeur sélectionnée par l'utilisateur dans l'action Afficher la boîte de dialogue des messages : si l'utilisateur sélectionne l'option 1, cette variable aura la valeur 1.	Cette variable est en lecture seule.
\$prop.[JAVA PROPERTY]\$	Pour consulter la valeur d'une propriété Java via une variable InstallAnywhere. Exemple : \$prop.os.name\$ renvoie la valeur de la propriété os.name.	Cette variable est en lecture seule.

\$REQUIRED_DISK_SPACE_BYTES\$	Chaîne indiquant l'espace nécessaire à l'installateur en octets (soit le résultat de l'action Évaluation de l'espace disque disponible). Cette variable reçoit sa valeur immédiatement avant l'installation des fichiers ou/et des dossiers listés dans la tâche Installation .	Cette variable est en lecture seule.
\$RESTART_NEEDED\$	Indique à l'installateur ou au désinstallateur si le système doit être redémarré pour terminer l'opération.	Cette variable est en lecture seule.
\$SHORTCUT_NAME\$	Cette variable prend la valeur « Shortcut/Raccourci » sur les systèmes Win32, « Alias » sur les systèmes Mac OS X et Classic, et « Link/Lien » sur tout autre système.	Cette variable est en lecture seule.
\$UNINSTALL_STATUS\$	Même valeur que \$INSTALL_STATUS\$, mais pour le désinstallateur.	Cette variable est en lecture seule.

Magic Folders et variables InstallAnywhere

Un « Magic Folder » correspond à un emplacement spécifique, par exemple le dossier d'installation sélectionné par l'utilisateur, le bureau Windows ou l'emplacement des fichiers de bibliothèque. Lors de l'installation, l'installateur détecte le système d'exploitation utilisé par l'ordinateur et il associe les Magic Folders aux chemins absolus appropriés. En général, les Magic Folders dépendent de la plate-forme considérée ; la plupart d'entre eux sont implantés automatiquement par InstallAnywhere dans des emplacements standard des différentes plates-formes supportées.

Chaque Magic Folder est associé à une variable InstallAnywhere. La première initialisation de ces variables a lieu au lancement de l'installateur. Chaque fois que la valeur d'une variable de Magic Folder est modifiée, le Magic Folder correspondant est installé dans une destination différente. Tout changement de valeur de \$USER_INSTALL_DIR\$ (dans InstallAnywhere) modifie la destination d'installation des fichiers.

À trois exceptions près (voir ci-dessous), ces variables sont initialisées lors de l'installation et ne changent pas – sauf par utilisation de code personnalisé ou par exécution de l'action « Définir une variable ». Les trois exceptions sont les suivantes :

- \$USER_INSTALL_DIR\$ – Cette variable est initialisée à la valeur par défaut déterminée dans la tâche **Plates-formes** d'Advanced Designer. Sa valeur peut changer lors de l'étape **Sélection du dossier d'installation**, si l'utilisateur sélectionne un dossier différent.
- \$USER_SHORTCUTS\$ – Cette variable est initialisée à la valeur par défaut déterminée dans la tâche **Plates-formes** d'Advanced Designer. Sa valeur peut changer lors de l'étape **Créer alias/liens/raccourcis**, si l'utilisateur sélectionne un emplacement différent.
- \$JAVA_HOME\$ – Pour un installateur sans VM, prend par défaut la valeur de la propriété Java java.home. Cette valeur peut changer lors de l'exécution de l'étape **Sélection de la VM Java**, si l'utilisateur sélectionne une VM. Pour un installateur avec VM, prend par défaut la valeur spécifiée dans la tâche **Projet | Java**. Cette valeur peut changer si \$USER_INSTALL_DIR\$ change, ou lors de l'étape **Sélection de la VM Java**, si l'utilisateur sélectionne une VM déjà présente sur son système.

Remarque : la valeur d'une variable ne peut pas être cette variable – pas même en partie. Exemple : pour compléter/tester USER_MAGIC_FOLDER_1, il n'est pas possible de déclarer USER_MAGIC_FOLDER_1 = USER_MAGIC_FOLDER_1\$/\$test. InstallAnywhere n'autorise pas la récursion directe ou indirecte avec les variables InstallAnywhere (cette situation déclenche une erreur).

Nom du dossier	Variable InstallAnywhere	Destination
Dossier d'installation sélectionné par l'utilisateur	\$USER_INSTALL_DIR\$	Dossier d'installation désigné par l'utilisateur. Les développeurs peuvent spécifier la valeur par défaut de cette variable dans l'écran Infos sur le projet

		d'Advanced Designer, et sélectionner un emplacement dans la zone Dossier d'installation par défaut.
Dossier des programmes (valeur par défaut de la plate-forme cible)	\$PROGRAMS_DIR\$	Dossier du système de destination dans lequel les applications sont installées par défaut, soit Windows = dossier \Program Files\ ; Mac OS = dossier Applications ; UNIX = compte Home de l'utilisateur ayant ouvert la session.
Raccourcis	\$USER_SHORTCUTS\$	Dossier spécifié par l'utilisateur comme emplacement pour les raccourcis/liens/alias. L'utilisateur peut changer la valeur de cet emplacement en activant l'action Sélectionner le dossier des alias/liens/raccourcis dans l'installateur. Les développeurs peuvent spécifier une valeur par défaut pour cette variable (en fonction de la plate-forme cible) en sélectionnant la tâche Plates-formes d'Advanced Designer.
Système	\$SYSTEM\$	Cette variable représente le dossier contenant les fichiers système de la machine cible. Sous Windows 95/98, elle prend la valeur <WINDOWS>\System. Sous Windows NT/2000, elle prend la valeur <WINDOWS>\System32. Sous Mac OS, sa valeur est : dossier System. Sous UNIX, sa valeur est : /usr/local/bin.
Bureau	\$DESKTOP\$	Cette variable représente le dossier « Bureau » de la machine cible (limité aux systèmes sous Windows, Linux et Mac OS).

Dossier temporaire	\$TEMP_DIR\$	Cette variable représente le dossier temporaire (Temp) de la machine cible. Si l'installateur exécuté est de type Pure Java sous Windows, \$TEMP_DIR\$ prend la valeur du dossier Home de l'utilisateur.
Démarrage	\$STARTUP\$	Dossier contenant les applications qui sont lancées automatiquement au démarrage du système d'exploitation (limité aux systèmes sous Windows et Mac OS).
Racine/root de l'unité d'installation	\$INSTALL_DRIVE_ROOT\$	Dossier root/racine du volume sélectionné pour l'installation.
Dossier Home	\$USER_HOME\$	Dossier Home de l'utilisateur qui exécute l'installateur. Cette variable est disponible avec toutes les plates-formes sauf Mac OS. Si la variable \$UNIX_USER_HOME\$ a déjà été spécifiée, elle utilisera la même définition que \$USER_HOME\$.
Racine/root de l'unité système	\$SYSTEM_DRIVE_ROOT\$	Dossier racine/root de l'unité contenant les fichiers système.
Dossier Home pour Java	\$JAVA_HOME\$	Dossier Home de la VM Java qui sera utilisée.
Windows	\$WIN_WINDOWS\$	Dossier Windows (uniquement pour systèmes sous Windows 95/98/Me/XP/NT/2000).
Menu Démarrer	\$WIN_START_MENU\$	Dossier contenant les applications déclarées dans le menu Démarrer de Windows (uniquement pour systèmes sous Windows 95/98/Me/XP/NT/2000).
Barre de lancement rapide	\$WIN_QUICK_LAUNCH_BAR\$	Barre de lancement rapide (Quick Launch) sous

(Quick Launch)		Windows. Sous Windows 2000 et XP, l'emplacement de la Quick Launch dépend de la valeur de la variable d'environnement UserProfile. Sous Windows 98/Me, l'emplacement de la barre Quick Launch est relatif par rapport au dossier Windows.
Ne pas installer	\$DO_NOT_INSTALL\$	Indique que le fichier ne doit pas être installé sur la plateforme cible. Utilisé pour les fichiers qui sont utilisés pendant l'installation mais qui ne sont pas conservés dans le système cible (par exemple, Contrat de licence localisé et éléments graphiques).
USER_MAGIC_FOLDER_#	\$USER_MAGIC_FOLDER_#\$	Ces variables représentent des Magic Folders de destination d'installation définies par l'utilisateur. Elles correspondent aux dossiers indiqués par leur nom. Pour définir ces variables, utilisez l'action Définir variable . Remarque : pour UNIX, si la barre oblique de début (/) n'est pas incluse dans le chemin du Magic Folder, le système interprète la spécification comme <le dossier correspondant au double-clic>/USER_MAGIC_FOLDER_#. En effet, le système considère par défaut que tout emplacement spécifié par un chemin non précédé de cette barre oblique réside sous le dossier actuel.
Menu Programmes	\$WIN_PROGRAMS_MENU\$	Menu Programmes du menu Démarrer de Windows (95/98/Me/XP/NT/2000 uniquement).
Menu Démarrer/All	\$WIN_COMMON_START	Menu Programmes/All Users du menu

Users	MENU\$	Démarrer de Windows (NT/2000 uniquement). Sous Win 9x, même valeur que \$WIN_START_MENU\$.
Menu All Users/Programmes	\$WIN_COMMON_PROGRAMS_MENU\$	Menu Programmes/All Users du menu Démarrer de Windows (NT/2000/XP uniquement). Sous Win 9x, même valeur que \$WIN_PROGRAMS_MENU\$.
Menu Démarrer/All Users	\$WIN_COMMON_STARTUP\$	Dossier All Users/Démarrage de Windows (dans le menu Démarrer) (Windows NT/2000/XP uniquement). Sous Win 9x, même valeur que \$STARTUP\$.
Bureau/All Users	\$WIN_COMMON_DESKTOP\$	Dossier « Bureau commun » de Windows (Windows NT/2000/XP uniquement). Sous Win 9x, même valeur que \$DESKTOP\$.
Polices	\$FONTS\$	Dossiers contenant les polices (Windows et Mac OS uniquement). \$WIN_WINDOWS\$\Fonts sous Windows; \$SYSTEM\$:Fonts sous Mac OS
Articles de menu Apple	\$MAC_APPLE_MENU\$	Dossier des articles de menu Apple (Mac OS uniquement).
Tableaux de bord	\$MAC_CONTROL_PANELS\$	Dossier des tableaux de bord Apple (Mac OS uniquement).
Extensions	\$MAC_EXTENSIONS\$	Dossier des extensions (Mac OS uniquement).

Préférences	\$MAC_PREFERENCES\$	Dossier des préférences (Mac OS uniquement). \$SYSTEM\$:Préférences
Nettoyage au démarrage	\$MAC_CHEWABLE\$	Dossier de nettoyage au démarrage (Mac OS uniquement). \$SYSTEM_DRIVE_ROOT\$:Nettoyage au démarrage
Applications de l'utilisateur	\$MACX_USER_APPLICATIONS\$	Dossier Applications de l'utilisateur de l'utilisateur qui exécute l'installateur (Mac OS X uniquement).
Dock	\$MACX_DOCK\$	Dock Mac OS X – Pour raccourcis uniquement. Les fichiers ne peuvent pas être installés dans le Dock.
/usr/local/bin	\$UNIX_USR_LOCAL_BIN\$	Répertoire /usr/local/bin (UNIX uniquement).
/opt	\$UNIX_OPT\$	Répertoire /opt directory (systèmes UNIX uniquement).
/usr/bin	\$UNIX_USR_BIN\$	Répertoire /usr/bin (systèmes UNIX uniquement).

Remarque : dans la version Standard Edition d'InstallAnywhere, les développeurs ne peuvent pas définir de Magic Folders.

Code personnalisé

La plupart des tâches nécessaires au déploiement d'une application peuvent être exécutées à l'aide des actions et des panneaux InstallAnywhere. Lorsqu'une fonctionnalité n'est pas assurée par les actions et panneaux standard, les développeurs peuvent créer les composants requis à l'aide de l'API de code personnalisé d'InstallAnywhere.

Cette API ouverte permet aux développeurs d'écrire du code Java exécutable dans l'architecture InstallAnywhere. Elle permet également d'accéder aux fonctionnalités complémentaires d'InstallAnywhere, par exemple le chargement des variables et des ressources. Les développeurs peuvent utiliser cette API pour définir des actions spéciales et des éléments d'interface graphique (GUI) qui dialoguent avec l'environnement InstallAnywhere et complètent celui-ci.

Tout code personnalisé inséré dans l'environnement InstallAnywhere doit être écrit en Java. Les principales formes de code personnalisé sont les actions, les panneaux, les consoles et les règles :

- **Actions en code personnalisé** – Ces actions s'exécutent dans le même environnement que les actions InstallAnywhere par défaut.
- **Panneaux en code personnalisé** – Ces panneaux s'affichent dans l'interface graphique d'InstallAnywhere (GUI) pendant l'installation. Ce mécanisme permet aux développeurs d'enrichir leur installateur de panneaux qui n'existent pas dans l'environnement par défaut d'InstallAnywhere.
- **Consoles en code personnalisé** – Ces actions s'exécutent dans l'interface à consoles d'InstallAnywhere, pendant l'installation. Ce mécanisme permet aux développeurs d'enrichir leur installateur de consoles personnalisées.
- **Règles en code personnalisé** – Ces règles sont analysées juste avant l'exécution de l'action à laquelle elles sont associées. Les règles en code personnalisé doivent renvoyer une valeur booléenne précisant si l'action doit être exécutée.

Zero G propose des exemples de code personnalisé pour les actions, pour les actions par panneaux et pour les actions par consoles. Ces échantillons résident dans le sous-dossier CustomCode (dossier d'installation racine/root d'InstallAnywhere). Ils illustrent des implémentations de code personnalisé réalisées via l'API d'InstallAnywhere prévue à cet effet. Par ailleurs, de nombreux exemples de code personnalisé (actions, panneaux et règles) sont proposés dans les pages **Support** du site Web de Zero G.

Code personnalisé et variables InstallAnywhere

Pour plus de détails sur les variables, consultez la rubrique **Variables InstallAnywhere standard**. Le code personnalisé peut lire et définir les variables InstallAnywhere, ce qui peut être utile dans les cas suivants :

- Un projet peut définir des variables InstallAnywhere, qui seront ensuite utilisées comme paramètres du code personnalisé. Exemple : si une action en code personnalisé détermine le dossier d'installation sélectionné par l'utilisateur, utilisez l'action **Définir variables InstallAnywhere** pour donner à PARAM_1 la valeur \$USER_INSTALL_DIR\$. Appelez ensuite (dans

l'action en code personnalisé)

`InstallerProxy.substitute("$PARAM_1$")` pour déterminer sa valeur.

- Les variables InstallAnywhere peuvent également recevoir les valeurs renvoyées par les actions en code personnalisé. Ces variables peuvent être interrogées par d'autres actions ou par d'autres règles ; il suffit pour cela d'appeler la méthode `InstallerProxy.setVariable("PARAM_1", <OBJECT>)` pour définir la valeur de la variable InstallAnywhere. N'oubliez pas que ces variables sont traitées en tant que chaînes, et qu'elles sont soumises à la méthode `java.lang.Object.toString()` pour déterminer leur valeur dans InstallAnywhere. Un fois définie, la valeur de `PARAM_1` peut être consultée normalement, par exemple en utilisant `$PARAM_1$` dans une règle ou dans une action InstallAnywhere ultérieure.

Lors de la consultation des valeurs des variables InstallAnywhere à partir d'un composant en code personnalisé, on constate parfois une utilisation incorrecte des méthodes `getVariable()` et `substitute()`, en particulier l'application de `getVariable()` à un Magic Folder tel que `$USER_INSTALL_DIR$` suivie de la conversion de cet objet en chaîne. Au lieu de convertir l'objet, il est conseillé d'utiliser `substitute()`, qui traduit en chaîne le contenu de la variable.

Créer et exécuter du code personnalisé dans un installateur

Pour créer et utiliser du code personnalisé dans un installateur :

1. Déclarez `IAClasses.zip` dans le classpath.

Lors de la compilation du code, le compilateur Java doit avoir accès aux classes présentes dans ce fichier. Remarque : `IAClasses.zip` réside dans le dossier racine/root d'installation d'InstallAnywhere.

2. Créez le composant de code personnalisé requis (action, panneau, console ou règle), par exemple en utilisant les modèles de fichier Java source stockés dans le sous-dossier `CustomCode/Templates` (dossier d'installation d'InstallAnywhere).
3. Compilez les fichiers source de ce composant.
4. Déterminez les fichiers et les ressources complémentaires requis (images, fichiers texte, etc.).
5. Créez une archive Java (JAR) non signée contenant les fichiers de classe/de ressources compilés (les archives signées ne fonctionnent pas correctement avec l'API de code personnalisé d'InstallAnywhere).

Remarque : vérifiez que l'archive sélectionnée contient des informations de chemin complet (chaque fichier de l'archive doit être stocké avec son chemin de paquetage, et celui-ci doit être correct). En l'absence de cette archive JAR, Java ne pourra pas retrouver le paquetage des fichiers de classe.

6. Ajoutez une action ou une règle qui exécute du code personnalisé (par exemple, **Exécuter code personnalisé**).
7. Sélectionnez le fichier JAR créé dans l'étape précédente.

8. Entrez le nom complet du paquetage et de la classe, par exemple :
`com.acme.MyAction`
9. Testez et déboguez l'action ou le panneau.

InstallAnywhere ne peut pas être exécuté à partir d'un environnement de développement intégré (IDE) ; par conséquent, la meilleure solution pour déboguer le code personnalisé consiste à utiliser l'action **Afficher informations de débogage**. Vous pouvez également utiliser l'instruction `System.out.println()` pour imprimer la sortie de débogage sur la console pendant les tests.

Remarque essentielle : le JAR de code personnalisé ne doit pas inclure les classes et les ressources contenues dans `IAClasses.zip` ; sinon, le comportement de l'installateur et du désinstallateur sera fortement perturbé.

Internationaliser le code personnalisé

L'API d'InstallAnywhere propose des fonctions complètes pour la localisation (traduction) des composants de code personnalisé (actions, panneaux et consoles). Exemple : localisation du label `java.awt.Label`, présent dans un panneau de code personnalisé. La méthode de définition de l'interface se présente comme suit :

```
public boolean setupUI(CustomCodePanelProxy ccpp)
{
    Label myLabel = new Label();
    myLabel.setText(ccpp.getValue(MyCustomCodePanel.myLabel));
}
```

`CustomCodePanelProxy`, `InstallerProxy`, `CustomCodeConsoleProxy` et `UninstallerProxy` permettent d'accéder à la méthode `getValue`. Cette méthode considère une chaîne comme un paramètre qui représente la partie « clé » de la paire clé/valeur telle que définie dans les fichiers de ressources internationaux d'InstallAnywhere. Les développeurs peuvent affecter le nom de leur choix aux clés, à condition que ces noms ne soient pas déjà utilisés par des clés définies auparavant ; ils peuvent même utiliser une clé existante pour consulter une chaîne qui a déjà été traduite dans les fichiers de ressources InstallAnywhere.

Chaque clé doit impérativement commencer par le préfixe `_ia`. Exemple : si le code personnalisé utilise la clé `MyCode.foo.flotsam`, celle-ci doit être ajoutée sous la forme `_ia.MyCode.foo.flotsam`. Ce préfixage doit être répété pour toute chaîne du code personnalisé qui est localisée (et pour chaque locale/langue supportée).

Remarque : pour tenir compte du code personnalisé existant, les installateurs recherchent d'abord les touches commençant par `_ia.`, puis ils effectuent une recherche complémentaire qui ignore le préfixe `_ia.` et porte uniquement sur la partie principale du nom de la clé.

Les paires clé/valeur des locales sont définies directement dans les fichiers de ressources des locales du projet (cette règle n'était pas appliquée dans les versions antérieures). Le nom de ces fichiers suit la syntaxe `custom_xx`, où `xx` est le code standard de deux lettres de la locale. Par exemple, le fichier de ressources de la locale « Anglais » s'appelle `custom_en`, le fichier de ressources de la locale « Japonais » s'appelle `custom_ja`, etc. Ces fichiers sont au format UTF-8 (pour plus de détails, consultez la section « Localisation » de la documentation).

Plug-ins

Avec le module Advanced Designer d'InstallAnywhere, les développeurs peuvent enregistrer leur code personnalisé sous forme de plug-ins, ce qui permet à ce code d'être intégré dans l'environnement de conception et d'apparaître dans la palette des actions d'InstallAnywhere, dans l'onglet **Plug-ins**. Les plug-ins résident dans le dossier <InstallAnywhere>/plugins. L'enregistrement du code personnalisé sous forme de plug-ins présente les avantages suivants :

Les développeurs peuvent ajouter ces plug-ins comme s'il s'agissait d'actions standard, sans avoir à spécifier de JAR ou de classe.

En général, le code personnalisé utilise des variables InstallAnywhere pour les paramètres et les valeurs renvoyées. Les variables InstallAnywhere ayant une portée (scope) globale, si les développeurs prévoient d'exécuter le même code personnalisé plusieurs fois dans un projet, ils doivent être très prudents avec les paramètres et les valeurs renvoyées ; par contre, les plug-ins ont seulement une portée locale pour les paramètres.

Les plug-ins peuvent facilement être partagés par les équipes de développement.

Remarque : Zero G peut se charger de la distribution des plug-ins créés par les développeurs. Si vous avez écrit un plug-in qui pourrait être utile à d'autres développeurs et que vous souhaitez le partager, contactez notre équipe de support technique : supportteam@zerog.com.

Pour enregistrer du code personnalisé sous forme de plug-in :

1. Enregistrez le code personnalisé et ses ressources dans un fichier JAR (même procédure que pour du code personnalisé standard).
2. Créez le fichier de propriétés customCode.properties (ce fichier contiendra les informations nécessaires à InstallAnywhere pour intégrer le plug-in à l'aide d'Advanced Designer). Placez ce fichier de propriétés dans le JAR (au niveau racine/root), **sans** informations de chemin.

Ce fichier doit impérativement contenir les propriétés suivantes :

- plugin.main.class=<classname> – Cette classe implémente l'élément de l'API InstallAnywhere nécessaire à l'action, au panneau ou à la console en code personnalisé considérée (par exemple, une classe qui implémente com.zerog.ia.api.pub.CustomCodeAction).
- plugin.name=<plug-in name> – Nom sous lequel le plug-in sera désigné dans la palette des actions d'Advanced Designer (onglet **Plug-ins**).
- plugin.type=<action | panel | console> – Précise si le composant en code personnalisé est une action, un panneau ou une console. Cette propriété permet à InstallAnywhere de déterminer le moment où le plug-in peut être utilisé et l'icône à afficher dans Advanced Designer pour représenter le composant en code personnalisé correspondant.

Les propriétés suivantes peuvent également être utilisées :

- property.<proptername>=<propertydefault> – Cette propriété demande au plug-in de renseigner le personnaliseur de l'action à l'aide de la propriété <nom_propriété> dont la valeur par défaut est <val_défaut_propriété>.

- `plugin.icon.path=<chemin relatif du fichier .png ou .jpg du JAR>` – Cette propriété définit l'icône personnalisée (32 x 32 pixels) qui représente le plug-in de code personnalisé dans Advanced Designer.
- `plugin.available=<preinstall | install | postinstall | preuninstall | postuninstall>` – Cette propriété se présente sous la forme d'une liste dont les valeurs (séparées par une virgule) indiquent à Advanced Designer les tâches pour lesquelles le plug-in doit être disponible.

Exemple de fichier de propriétés pour plug-in :

```
plugin.main.class=com.zerog.ia.customcode.util.fileutils.ExtractToFile
plugin.name=Extract to File (Extraire données et enregistrer dans fichier)
plugin.type=action
plugin.icon.path=myicon.gif
plugin.available=preinstall,install,postinstall
property.ExtractToFile_Source=path/to/file/in.zip
property.ExtractToFile_Destination=$USER_INSTALL_DIR$$/myfile.txt
```

3. Les plug-ins peuvent afficher de l'aide à l'attention des utilisateurs InstallAnywhere ; cette aide est au format HTML et s'affiche lorsque l'utilisateur clique sur le bouton approprié dans le personnaliseur de plug-in. Remarque : ce bouton apparaît seulement si un fichier d'aide a été inclus dans le plug-in. Pour créer et afficher cette aide, procédez comme suit :
 - a. Créez le fichier `help.htm` contenant les informations requises.
 - b. Enregistrez ce fichier d'aide dans le JAR du plug-in, **sans** informations de chemin.
4. Enregistrez le JAR (non signé, contenant le code personnalisé et ses ressources et le fichier de propriétés) dans le dossier `<InstallAnywhere>/plugins`.
5. Lancez InstallAnywhere : le plug-in apparaît dans la palette des actions InstallAnywhere.

Désinstallateur

InstallAnywhere crée automatiquement un désinstallateur qui peut être supprimé manuellement. Le désinstallateur InstallAnywhere supprime les fichiers et actions exécutées par la tâche **Fichiers** lors de l'installation. Les actions ajoutées dans d'autres phases de l'installation ne peuvent pas être défaites à l'aide du désinstallateur ; il est nécessaire de tenir compte de ce fait en cas de désinstallation (et éventuellement de les défaire manuellement).

De structure semblable à celle de l'installateur, le désinstallateur se compose de panneaux, de consoles et d'actions. Il consigne les détails des actions exécutées par l'installateur et des actions exécutées lors de l'installation. Le désinstallateur exécute d'abord les panneaux, les actions et les consoles de pré-désinstallation ; il désinstalle ensuite les actions installées par la tâche Fichiers ; enfin, il exécute les actions de post-désinstallation.

Code personnalisé

Comme c'est le cas avec l'installateur, il est possible d'ajouter du code personnalisé au désinstallateur pour enrichir ses fonctionnalités. Pour inclure du code personnalisé dans le désinstallateur, il existe plusieurs solutions : (1) Inclure des actions, des panneaux ou/et des consoles en code personnalisé dans la tâche de pré-désinstallation ou de post-désinstallation. La méthode `uninstall()` est appelée pour les actions en code personnalisé. (2) Insérer une action en code personnalisé dans la tâche Fichiers. Dans ce cas, la méthode `install()` de l'action est appelée lors de l'installation et la méthode `uninstall()` de cette même action est appelée lors de la désinstallation.

Désinstaller certaines fonctions

Chaque projet d'installateur comporte un désinstallateur. Toutes les fonctions installées sont enregistrées par le désinstallateur (dans un registre local). Si le panneau « Sélectionner les fonctions à désinstaller » est inclus dans le désinstallateur, l'utilisateur a la possibilité de désinstaller uniquement les fonctions de son choix.

Remarque : cette nouvelle configuration remplace celle qui était disponible à partir d'InstallAnywhere 5.0 et 5.5.

Cette fonction permet aux utilisateurs de sélectionner les fonctions à désinstaller. Si l'utilisateur demande la désinstallation d'une fonction qui partage un composant avec une fonction qu'il n'envisage pas de désinstaller, le désinstallateur détecte ce conflit et ne désinstalle pas le composant partagé.

Désinstallateur pour plusieurs produits

Chaque désinstallateur est lié à un produit donné, déterminé par l’ID produit stocké dans la tâche **Projet | Description**. Si vous souhaitez qu’un désinstallateur fonctionne pour un groupe de produits distincts, chaque produit doit être affecté du même ID. Les produits doivent ensuite être déclarés comme fonctions. Les désinstallateurs des différents produits doivent également partager le même nom et le même emplacement d’installateur. Exemple :

Produit : OfficeSuite Acme (ID : 97338341-1ec9-11b2-90e2-a43171489d33)

Installateur 1 : Traitement de texte Acme et Tableur Acme

ID produit : 97338341-1ec9-11b2-90e2-a43171489d33

Fonctions : Traitement de texte Acme et Tableur Acme

Installateur 2 : Diaporama Acme

ID produit : 97338341-1ec9-11b2-90e2-a43171489d33

Fonctions : Diaporama Acme

Résultat final : Un désinstallateur pour ces trois fonctions

Modules de fusion et modèles

Modules de fusion

Les « modules de fusion » sont essentiellement des sous-projets d'installateur qui peuvent être créés de manière indépendante, puis fusionnés. Comme un installateur, un module de fusion est un ensemble réutilisable de fonctionnalités d'installation avec ses propres fonctions, composants, panneaux, actions et fichiers. Toutefois, un module de fusion ne peut pas être installé de manière autonome : les développeurs utilisent un module de fusion lorsqu'ils souhaitent inclure la fonctionnalité d'un installateur dans un autre installateur.

Les modules de fusion apportent des solutions pour les installations complexes, par exemple :

- Combiner plusieurs modules de fusion à partir de différents produits.
- Permettre à plusieurs équipes de développement indépendantes et situées dans des lieux différents de créer des modules de fusion pour différents composants logiciels et de combiner ces modules en un seul installateur.
- Créer des fonctionnalités d'installateur autonomes en vue de les réutiliser dans des projets futurs d'installateur. Par exemple, si le même composant logiciel doit être utilisé dans plusieurs installateurs, vous pouvez le construire dans un module de fusion et le distribuer aux développeurs concernés.
- Enregistrer des éléments communs à plusieurs installateurs (par exemple, panneau du Contrat de licence et panneaux de saisie utilisateur) dans des modules de fusion pour simplifier les projets futurs d'installateur.
- Combiner des modules de fusion à partir de paquetages de sociétés tierces pour construire des solutions logicielles complexes sans avoir à déterminer les modalités d'installation de chaque paquetage.

Remarque : Zero G propose des modules de fusion de sociétés tierces, qui peuvent être téléchargés à partir de son site Web : <http://www.zerog.com/>.

- Utilisez un module de fusion comme point de départ de vos projets d'installateur (généralement désignés « modèles », ces modules de fusion sont traités en détail dans une autre rubrique d'aide).
- Tout projet d'installateur peut être construit sous la forme d'un module de fusion, et tout module de fusion peut être utilisé dans tout projet d'installateur.
- Les modules de fusion sont créés comme option tout au long du processus de construction de l'installateur. Dans la mesure où un module de fusion contient toutes les ressources nécessaires à un projet, sa construction est semblable à celle d'un installateur. Les modules de fusion peuvent être construits automatiquement lors de la construction de l'installateur, ou être construits explicitement dans Advanced Designer (cochez la case **Construire module de fusion** de la tâche **Construction du projet**, sous l'onglet **Distribution**) ou à partir de la ligne de commande (option +merge).

Pour intégrer des modules de fusion à un installateur existant, il existe deux solutions :

- Dans la tâche **Structure | Modules**, cliquez sur **Importer un module de fusion**. Tous les éléments du module de fusion (fonctions, composants, fichiers, actions et, sur option, panneaux) sont combinés dans le projet actuel, ce qui permet aux développeurs de personnaliser l'ensemble des paramètres existants.
- Les modules de fusion peuvent également être installés sous forme de « sous-installateurs » autonomes, sans être intégrés au projet actuel. Cette solution est utile si les développeurs ne connaissent pas le contenu du module de fusion, ou s'ils n'envisagent pas de modifier les paramètres ; les modules de fusion ajoutés de cette manière sont exécutés sous la forme de sous-installateurs « silencieux ».

Pour intégrer des modules de fusion au projet actuel, il existe deux solutions :

- Exécutez l'action **Installer le module de fusion** et sélectionnez **Intégrer un module de fusion lors de la construction**, si le module de fusion est disponible lorsque vous êtes prêt à construire l'installateur. Le module de fusion sera inclus dans l'installateur.
- Utilisez l'action **Installer le module de fusion** et activez l'option **Sélectionner le module de fusion à l'installation** : l'installateur installera un module de fusion qui sera disponible lors de l'installation tout en restant externe à l'installateur. Le module de fusion peut résider dans le système de l'utilisateur ou être stocké sur CD. Si l'emplacement est un dossier qui contient plusieurs modules de fusion, ceux-ci sont tous installés.

Remarques importantes sur les modules de fusion

- Option **Lecture seule** – Les modules de fusion peuvent être verrouillés, ce qui empêche de les ouvrir, de les utiliser comme modèles ou de les fusionner dans un installateur. Les modules de fusion en lecture seule peuvent être installés uniquement sous forme d'éléments d'installateur autonomes.
- **Optimiser la taille du module de fusion selon plate-forme** – Les modules de fusion créés pour chaque plate-forme présentent des caractéristiques différentes. Chaque module contient uniquement les ressources nécessaires à la plate-forme cible. N'utilisez pas cette option si vous importez des modules de fusion dans un autre installateur. Lorsque vous importez un module de fusion, celui-ci ne doit pas être optimisé.
- **Variables publiées** – Variables InstallAnywhere qui doivent être définies avant d'installer un module de fusion via l'action **Installer le module de fusion**. Dans l'onglet **Distribution** de la tâche **Construction du projet**, cliquez sur **Éditer les variables publiées** : vous pourrez ensuite déclarer des variables, définir leur valeur par défaut et ajouter des commentaires. Les variables publiées communiquent aux installateurs les valeurs nécessaires à la configuration des modules de fusion.

- Les variables InstallAnywhere peuvent être transmises au module de fusion via l'action **Installer le module de fusion**. Seules les variables sélectionnés sont transmises au module de fusion. Par défaut, toute variable publiée définie par le module de fusion (les variables publiées sont définies lors de la construction du module) est automatiquement transmise. Des variables spécifiques peuvent également être transmises via le personnaliseur du module de fusion. Exemples : si la variable de Magic Folder \$IA_PROJECT_DIR\$ était publiée par le module de fusion, elle serait transmise. Si la variable \$OTHER_VARIABLE\$ n'était pas publiée mais simplement déclarée dans le personnaliseur de l'action **Installer le module de fusion**, elle serait également transmise.

Modèles

Le « modèle » est le point de départ de tout projet d'installateur. Un modèle peut être un projet vide, ou contenir les mêmes éléments qu'un projet standard (contrat de licence, éléments graphique, billboards et même fichiers).

Un modèle est en fait un module de fusion qui a été enregistré dans le sous-dossier iatemplates du dossier d'installation d'InstallAnywhere. Lorsque vous créez un projet, vous avez la possibilité d'utiliser un modèle comme point de départ ; dans ce cas, une copie du modèle est créée et enregistrée.

Les modèles peuvent être intéressants pour les équipes composées d'un grand nombre de développeurs et dans lesquelles vous souhaitez que chaque personne utilise le même modèle comme point de départ, ou pour commencer un nouveau projet à partir d'un ancien modèle.

Installateurs en mode Console ou en mode Silencieux

Installateurs en mode Silencieux

Le mode Silencieux, qui permet d'exécuter un installateur sans aucune interaction utilisateur, est supporté par toutes les plates-formes UNIX. Un mode « quasi silencieux » est également possible sous Windows et Mac OS X. Les variables InstallAnywhere et les variables utilisateur peuvent être définies à l'aide de paramètres en ligne de commande et/ou d'un fichier de propriétés.

Pour lancer un installateur en mode Silencieux à partir de la ligne de commande, entrez l'instruction suivante :

```
installername -i silent
```

Vous pouvez également appeler un fichier de propriétés à partir de la ligne de commande :

```
installername -f <properties file>
```

Vous pouvez spécifier le chemin absolu ou relatif du fichier de propriétés.

Remarque : Il est possible d'incorporer des variables InstallAnywhere dans ces valeurs : elles seront affectées d'une valeur lors de l'installation.

Fichiers de réponse et installateurs en mode Silencieux

Le mode Silencieux est particulièrement utile en entreprise, lorsque l'installation porte sur un grand nombre de postes de travail. En mode Silencieux, InstallAnywhere n'exige aucune interaction utilisateur : il utilise les valeurs par défaut fournies par les développeurs, ou il récupère dans un « fichier de réponse » les valeurs des variables InstallAnywhere qui contrôlent l'installation.

L'installateur recherche dans son dossier un fichier appelé `installer.properties` ou `[installername].properties`. Vous pouvez également désigner le fichier de propriétés que l'installateur doit utiliser en entrant le commutateur/switch suivant en ligne de commande :

```
-f /path/to/properties file
```

Ce fichier utilise un simple format `key=value`.

Par exemple, l'installateur à consoles construit précédemment dispose d'une seule option véritable – le dossier d'installation.

Exemple de fichier de propriétés :

```
INSTALLER_UI=silent
```

```
USER_INSTALL_DIR=C:\Program Files\OfficeSuite\
```

La variable `INSTALLER_UI` permet de spécifier le mode de l'installateur dans le fichier de propriétés ; dans ce cas, il n'est pas nécessaire de spécifier le commutateur de silence `-i` en ligne de commande.

Installeurs par consoles

Le mode Console reproduit les étapes d'une installation en mode graphique/GUI et utilise une entrée et une sortie standard. Le principal avantage du mode Console pour les développeurs UNIX est qu'ils n'ont plus besoin de l'environnement X Window (X11) pour exécuter leurs installeurs.

Le mode Console permet d'afficher le texte ligne par ligne sur la console, mais il ne supporte aucun formatage et ne permet pas l'effacement de l'écran ni le déplacement manuel du curseur.

```
+-----+
| Sélectionnez le dossier des alias, des liens ou des raccourcis |
| |
| Dans quel emplacement voulez-vous créer les raccourcis de
| l'application ? |
| |
| 1) Dans le menu Démarrer |
| 2) Sur le bureau |
| 3) Ne pas créer de raccourcis |
|-----|
| Sélectionnez 1, 2 ou 3 et |
| appuyez sur Entrée. |
|-----|
```

Pour lancer un installateur par consoles à partir de la ligne de commande, entrez l'instruction suivante :

```
installername -i console
```

Remarques spécifiques aux différentes plates-formes

UNIX

Lorsque l'installateur utilise une VM Java 1.1.x, l'action par panneau « Afficher mot de passe » n'affiche pas les caractères du mot de passe sous forme d'astérisques. Un bogue présent dans certaines implémentations de Java 1.1.x peut même entraîner le blocage (crash) de l'installateur si le champ de texte est configuré avec écho des caractères. Si l'installateur UNIX est utilisé avec une VM Java 1.1.x, ce panneau affiche la forme visible du mot de passe saisi (et non des astérisques).

L'action par console « Sélection de la VM Java » établit des liens symboliques pour les données décrivant le chemin d'accès à une VM, ce qui peut être déroutant lorsque ce chemin est visualisé.

Sous Linux RedHat 9, InstallAnywhere ne supporte pas les VM antérieures à Java 2. Il s'agit d'un problème connu ; causé par le passage à un nouveau modèle de threading dans ce système d'exploitation.

Mac OS X

Tableau des Magic Folders pour Mac OS X

Le tableau suivant décrit le comportement appliqué sous Mac OS X aux utilisateurs authentifiés et aux utilisateurs standard.

Magic Folders	Utilisateur sans privilèges d'administrateur	Utilisateur sans privilèges d'administrateur, après authentification
Applications de l'utilisateur	Applications de l'utilisateur	Applications de l'utilisateur
Bureau	Bureau	Bureau
Dock	Dock utilisateur	Dock utilisateur
Dossier des programmes	Applications	Applications
Dossier Home de l'utilisateur	Dossier Home de l'utilisateur	Dossier Home de l'utilisateur
Groupe	Groupe de l'utilisateur	Groupe administrateur
JavaHome	JavaHome	JavaHome
Nettoyage au démarrage	Corbeille	Corbeille

Polices	Polices utilisateur	Polices système
Préférences	Préférences de l'utilisateur	Préférences de l'utilisateur
Propriétaire	Utilisateur	Root/racine
Propriétaires/Autorisations	Utilisateur sans privilèges d'administrateur	Utilisateur sans privilèges d'administrateur, après authentification
Racine/root de l'unité d'installation	/, /Volumes/*	/, /Volumes/*
Racine/root de l'unité système	/	/
Système	Système	Système

Remarques :

- la modification des éléments en italiques doit être réservée aux utilisateurs expérimentés.
- L'astérisque (*) correspond à l'ensemble des volumes (partitions) disponibles, à l'exception de Startup.
- Lorsqu'un installateur est authentifié, vous pouvez l'exécuter en tant qu'utilisateur root/racine. Les panneaux Mac OS X « Sélectionner un dossier » sont associés à des alias qui pointent sur des emplacements tels que le Bureau ou Home/Départ. Les utilisateurs qui sélectionnent ces panneaux à partir d'une zone de dialogue Mac OS X native constateront qu'ils sont associés aux emplacements Bureau, Home/Départ, etc. de l'utilisateur root/racine. Ils ne pourront donc pas toujours accéder aux fichiers stockés dans ces emplacements. Dans ce cas, nous suggérons aux développeurs d'analyser les chemins à l'aide de la règle « Correspond à l'expression régulière » et d'afficher un message pour avertir les utilisateurs de cette situation.

z/OS et OS/390

z/OS et OS/390 utilisent le codage EBCDIC et non ASCII. Les installateurs InstallAnywhere fonctionnent correctement sur ces plates-formes, mais certaines actions présentent un comportement différent.

L'installation des fichiers est assurée par un mécanisme de copie binaire (les fichiers texte sont installés de la même manière). Si vous construisez l'installateur sur un système Windows destiné à être installé sous z/OS, les fichiers texte seront installés au format ASCII. Pour installer les fichiers texte au format EBCDIC, convertissez-les à l'aide d'un outil tel que « native2ascii » (inclus dans le SDK J2SE), puis ajoutez-les séparément dans l'installateur. Appliquez les règles « Vérifier la plate-forme » aux fichiers texte ASCII et aux fichiers texte EBCDIC pour spécifier leur plate-forme d'installation respective.

Remarques spécifiques aux différentes plates-formes

Certaines actions, par exemple « Modification du fichier texte », modifient l'encodage du système de destination. Si vous souhaitez que « Modification du fichier texte » s'applique à des fichiers ASCII, exécutez un utilitaire tel que « native2ascii » pour convertir le fichier en ASCII lorsque cette action est terminée.

Outils de construction

Exemples d'utilisation de la ligne de commande lors de la construction d'un installateur :

- Pour construire le ou les installateurs du projet « MonProduit » en utilisant (par défaut) les paramètres de construction enregistrés auparavant :
`build MonProduit.iap_xml`
- Pour construire le ou les installateurs du projet « MonProduit » en ignorant les paramètres de construction enregistrés auparavant et en spécifiant un fichier de propriétés Build.properties :
`build MonProduit.iap_xml -p Build.properties`
- Pour construire les installateurs du projet « MonProduit » seulement pour les plates-formes Mac OS X et Linux (signe +), en excluant les autres (signe -) :
`build MonProduit.iap_xml +x +l -s -j -w -u`

Options des outils de construction

Un exemple de fichier de propriétés de construction (Build.properties) est proposé dans le dossier d'application d'InstallAnywhere (sous resource/build).

-v	Cet argument permet d'afficher le numéro de version d'InstallAnywhere, par exemple : C: \Program Files\InstallAnywhere 6 Enterprise>build.exe -v InstallAnywhere 6 Enterprise
-p	Cet argument permet de désigner le fichier de propriétés de construction (Build.properties) à utiliser.
+	Cet argument permet d'ajouter une plate-forme (dont le nom suit l'argument – voir tableau ci-dessous) comme cible de la construction.
-	Cet argument permet de retirer une plate-forme (dont le nom – voir tableau ci-dessous – suit l'argument) comme cible de la construction.

Les arguments suivants modifient les arguments d'ajout (+) et de suppression (-) de plate-forme :

a, A	AIX sans VM
av, AV	AIX avec VM
h, H	HP-UX sans VM
hv, HV	HP-UX avec VM
j, J, o, O	Pure Java
l, L	Linux sans VM

lv, LV	Linux avec VM
s, S	Solaris sans VM
sv, SV	Solaris avec VM
u, U	UNIX générique sans VM
n, N	UNIX spécifique sans VM
nv, NV	UNIX spécifique avec VM
w, W	Windows sans VM
wv, WV	Windows avec VM
x, X	Mac OS X sans VM
web	Construire des installateurs Web
cd	Construire des installateurs CD
opt	Optimiser la taille de l'installateur selon plate-forme
merge	Construire des modules de fusion

Codes en sortie

Pour indiquer le résultat de la construction, l'outil en ligne de commande renvoie l'un des codes suivants :

Code	Résultat
Codes relatifs aux ressources	
cancelled1	Ressources manquantes, abandon de la construction annulé par les préférences
Codes relatifs au fichier de projet	
101	Impossible de charger le projet
102	Impossible de copier le projet
103	Fichier de projet introuvable
104	Fichier de projet en lecture seule
199	Fichier de projet inconnu

Codes relatifs aux options en ligne de commande	
200	Drapeau/flag de construction interdit
201	Drapeau/flag de construction incorrect
Codes relatifs au Pack VM	
300	Pack VM remplacé
301	Pack VM introuvable
302	Pack VM de format interdit
399	Pack VM inconnu
Erreurs d'écriture dans fichier cible	
400	Fichier cible introuvable
401	Fichier cible occupé
402	Fichier cible protégé en écriture
403	Écriture impossible
499	Erreur inconnue
Erreurs de lecture du fichier source	
500	Fichier source introuvable
501	Fichier source occupé
502	Fichier protégé en lecture
503	Lecture impossible
599	Erreur inconnue
Autres codes	
-1	Autres erreurs/erreurs inconnues
0	Aucune erreur : la construction a été exécutée avec succès, sans erreur ni avertissement.
666	Droits insuffisants pour ce dossier
799	Erreur inconnue

Construire avec l'outil Ant

Développé en Java dans le cadre du projet Apache Jakarta, Ant est un outil de construction très puissant. Il peut être utilisé pour contrôler des tâches de construction complexes sous Java et dans d'autres environnements de développement. Ant gère les actions spécifiques par l'intermédiaire de « tâches » qui peuvent faire partie de la distribution Ant ou être disponibles sous forme d'extensions.

InstallAnywhere est livré avec une tâche Ant pour permettre la construction d'installateurs à l'aide de l'outil Ant. La tâche Ant réside dans le dossier d'application d'InstallAnywhere, dans resource puis dans /build. Pour accéder à la tâche Ant d'InstallAnywhere, il suffit d'ajouter `iaant.jar` au classpath lors de l'exécution de Ant.

Ant utilise un fichier XML pour préciser l'ordre d'exécution des tâches de la construction. Pour plus de détails sur Ant, visitez les pages consacrées au projet Jakarta sur le site Web d'Apache Foundation.

Nom et classe de tâche

Dans InstallAnywhere, la tâche Ant est spécifiée comme suit :

```
<taskdef name="buildinstaller"
classname="com.zerog.ia.integration.ant.InstallAnywhereAntTask"/>
```

Cette tâche est stockée dans le fichier `iaant.jar` (dossier d'application d'InstallAnywhere, sous-dossier resource, puis build).

Paramètres

Après avoir défini la tâche, spécifiez les paramètres nécessaires à la construction :

```
<buildinstaller
IAlocation="C:\Program Files\InstallAnywhere 6 Enterprise"
IAProjectFile="C:\Projects\myproject.iap.xml"
additionalparameter=value
/>
```

Remplacez `IAlocation` par le chemin absolu du dossier d'application d'InstallAnywhere. Si InstallAnywhere est introuvable, Ant le recherche dans l'un des emplacements par défaut ou dans le registre.

Dans le paramètre `IAProjectFile`, spécifiez le chemin et le nom de fichier du projet à construire.

Les autres propriétés sont facultatives. Les paramètres correspondent étroitement aux propriétés présentes dans le fichier des propriétés de construction décrit précédemment. Le tableau suivant présente les paramètres disponibles.

Attribut	Description	Obligatoire
IAProjectFile	Emplacement du projet InstallAnywhere à construire	Oui
IALocation	Emplacement d'installation d'InstallAnywhere	Non*
propertiesfile	Emplacement d'un fichier build.properties que vous pouvez utiliser (si ce fichier est utilisé, les autres attributs sont ignorés)	Non
failonerror	Abandonne la construction si la commande est abandonnée et renvoie un code autre que 0 (valeur par défaut = « false »)	Non

Remarques :

- Si IALocation n'est pas spécifié, la tâche recherche une copie d'InstallAnywhere. Si InstallAnywhere n'est pas installé dans l'un des emplacements par défaut, la tâche recherche un emplacement dans le registre produit d'InstallAnywhere.
- Rappel – Pour pouvoir utiliser la tâche Ant d'InstallAnywhere, vous devez ajouter iaant.jar à \$ANT_HOME/lib/ext.

Paramètres de la plate-forme cible

IAProjectFile	Emplacement du projet InstallAnywhere à construire
IALocation	Emplacement d'installation d'InstallAnywhere sur le système cible
Failonerror	Abandonne la construction si la commande est abandonnée et renvoie un code autre que 0 (valeur par défaut = « false »)
Propertiesfile	Emplacement d'un fichier build.properties que vous pouvez utiliser (si ce fichier est utilisé, les autres attributs sont ignorés)

Options de la plate-forme

BuildLinuxWithVM	true/false
BuildLinuxWithoutVM	true/false
LinuxVMPackLocation	Chemin
BuildHPUXWithVM	true/false
BuildHPUXWithoutVM	true/false

HPUXVMPackLocation	Chemin
BuildAIXWithVM	true/false
BuildAIXWithoutVM	true/false
AIXVMPackLocation	Chemin
BuildSolarisWithVM	true/false
BuildSolarisWithoutVM	true/false
	Chemin
BuildNamedUNIXWithVM	true/false
BuildNamedUNIXWithoutVM	true/false
NamedUNIXVMPackLocation	Chemin
NamedUNIXTitle	Chemin
BuildWindowsWithVM	true/false
BuildWindowsWithoutVM	true/false
WindowsVMPackLocation	Chemin
BuildMacOSX	true/false
BuildPureJava	true/false

Options de construction

BuildCDROMInstaller	true/false
BuildWebInstaller	true/false
BuildMergeModule	true/false
BuildReadOnlyMergeModule	true/false
BuildOutputLocation	Chemin
OptimizeCDROMInstaller	true/false
OptimizeWebInstaller	true/false
OptimizeMergeModule	true/false
AutoPopulateLabels	true/false
AutoCleanComponents	true/false

Options de l'installateur

InstallerStdErrRedirect	Chemin
InstallerStdoutRedirect	Chemin
InstallerValidVMList	Chaîne
InstallerInitialHeapSize	Entier (Int)
InstallerMaxHeapSize	Entier (Int)
UNIXDefaultUI	silent/console/gui

Exemple

```
<taskdef name="buildinstaller"
classname="com.zerog.ia.integration.ant.InstallAnywhereAntTask"/>
<buildinstaller
```

```
IAlocation="C:\Program Files\InstallAnywhere 6 Enterprise"
```

```
IAProjectFile="C:\Projects\myproject.iap.xml"
```

```
BuildLinuxWithVM="true"
BuildMacOSX="true"
BuildWebInstallers="true"
OptimizeWebInstallers="true"
InstallerStdErrRedirect="C:\console.txt"
/>
```


Localisation

Localisation (installateurs en langues étrangères)

La version Enterprise d'InstallAnywhere permet aux développeurs de construire des installateurs dans 29 langues (la version Standard supporte neuf langues), ou « locales ». Pour générer un installateur en plusieurs langues, choisissez la tâche **Projet | Locales d'Advanced Designer**, puis cochez la case des langues requises.

Lors de la construction d'un projet d'installateur, le dossier <projectname>locales est automatiquement créé dans le même dossier que le fichier de projet. Pour chaque locale sélectionnée dans Advanced Designer, un fichier est automatiquement créé dans ce dossier <projectname>locales. Les fichiers de locale sont générés sous la forme custom_<localecode> ; par exemple, pour le français (code de locale fr), le nom du fichier de locale sera custom_fr. Ces fichiers contiennent les clés et les valeurs nécessaires aux chaînes dynamiques du projet. Les clés sont générées par le nom de l'action ; elles sont affectées d'une valeur unique qui représente une instance unique de l'action et d'un paramètre complémentaire indiquant qu'une clé pointe sur la valeur dynamique de l'action. Exemple :

```
InstallSet.9733839b90f6.installSetName=Typical
InstallSet.9733839b90f6.description= Les fonctions de base de
l'application seront installées. Cette option est recommandée pour
la plupart des utilisateurs.
```

Le fichier ProjectLocalizationInfo.txt contient le mappage (correspondances) entre les actions du projet et leurs clés dans les fichiers de locale. Pour déterminer les relations définies entre les actions et les clés, consultez le fichier ProjectLocalizationInfo.txt.

Texte statique et texte dynamique

Les chaînes qui peuvent être entrées dans le module Advanced Designer d'InstallAnywhere sont en « texte dynamique » (texte dont la valeur peut être modifiée par les développeurs). Le texte qui ne peut pas être modifié par Advanced Designer est du « texte statique ». Les éléments standard tels que les sélecteurs de fichier ou le contenu des actions et des panneaux que les développeurs n'envisagent pas d'éditer utilisent du texte statique. Le texte dynamique est enregistré dans les fichiers de locale (qui résident à côté du fichier de projet). Les traductions de texte statique sont regroupées dans le dossier <InstallAnywhere>\resource\i18nresources. Bien que les développeurs n'aient généralement pas besoin de modifier le texte statique, InstallAnywhere dispose des fonctions nécessaires.

La plupart des chaînes de texte dynamique disposent d'une valeur par défaut dans Advanced Designer ; par ailleurs, ces valeurs dynamiques sont associées à une traduction par défaut. Toute chaîne modifiée dans Advanced Designer doit être localisée par les développeurs s'ils souhaitent que la traduction corresponde. Par ailleurs, certaines actions ne disposent pas de valeur par défaut, par exemple les composants en code personnalisé et les différents panneaux **Entrée utilisateur**. Les développeurs doivent également prévoir la localisation de ces chaînes.

Localiser les installateurs et Advanced Designer

Avec cette version d'InstallAnywhere, les développeurs peuvent non seulement localiser les installateurs construits dans l'environnement de développement InstallAnywhere, mais ils peuvent aussi localiser l'environnement de développement lui-même.

Les modifications de valeur dynamique sont enregistrées dans le fichier de locale de la langue dans laquelle Advanced Designer s'exécute. Exemple : si vous sélectionnez la langue française, les modifications dynamiques sont enregistrées dans le fichier `custom_fr`. Pour modifier ce fichier de locale, vous devez utiliser Advanced Designer : les modifications effectuées à l'extérieur d'Advanced Designer seront écrasées.

Conseils de localisation

Localiser les ressources

Lorsqu'un développeur envisage de commercialiser son produit dans plusieurs pays, il peut en localiser les ressources (Contrat de licence, panneaux, billboards, icônes, etc.). Cette opération est possible dans InstallAnywhere. La plupart des actions (y compris le Contrat de licence et LaunchAnywhere) disposent de liens entre chemins/noms de fichier et ressources, et entre chaînes dynamiques et fichiers de locale. Il suffit donc de modifier les chemins et les noms de fichier selon besoin. Par exemple, pour localiser le Contrat de licence :

Vérifiez que toutes les ressources requises sont incluses dans la tâche Fichiers (les installateurs ne peuvent pas accéder aux ressources qui ne sont pas spécifiées dans cette tâche).

Dans le fichier de locale, retrouvez la ligne qui contient le texte `LicenseAgr.#.FileName`. Spécifiez le nom du fichier qui contient la version localisée du Contrat de licence (par exemple, `License_fr.html`). Ne tapez pas le chemin absolu du fichier, mais simplement son nom.

Retrouvez la ligne qui contient le texte `LicenseAgr.#.Path`. Spécifiez le chemin du fichier qui contient (dans le système local) la version localisée du Contrat de licence.

Localiser les labels

Les labels ajoutés par les développeurs dans les panneaux des installateurs ne sont pas automatiquement localisés. Pour localiser les labels :

1. Construisez l'installateur comme vous en avez l'habitude.
2. Affichez le contenu du dossier « Locale » de l'installateur.
3. Ouvrez le fichier `custom_en` dans un éditeur de texte.
4. Recherchez la variable `Installer.1.installLabelsAsCommaSeparatedString`.
Cette variable contient les labels actuels de l'installateur.
5. Copiez-collez cette variable dans les autres fichiers de locale.
6. Insérez la traduction des labels dans leur fichier respectif.

Localiser le code personnalisé

L'API d'InstallAnywhere permet de localiser les actions, panneaux et consoles écrits en code personnalisé. Vous trouverez ci-dessous un exemple de localisation portant sur le label `java.awt.Label`, qui s'affiche dans un panneau en code personnalisé. La méthode de définition de l'interface (`setupUI`) se présente comme suit :

```
public boolean setupUI(CustomCodePanelProxy ccpp)
{
    Label myLabel = new Label();
    myLabel.setText(ccpp.getValue(MyCustomCodePanel.myLabel));
}
```

Chaque élément `CustomCodePanelProxy`, `InstallerProxy`, `CustomCodeConsoleProxy` et `UninstallerProxy` donnent accès à la méthode `getValue`. Cette méthode considère une chaîne comme un paramètre qui représente la partie « clé » de la paire clé/valeur telle que définie dans les fichiers de ressources internationaux d'InstallAnywhere. Les développeurs peuvent affecter le nom de leur choix aux clés, à condition que ces noms ne soient pas déjà utilisés par des clés définies auparavant ; ils peuvent même utiliser une clé existante pour consulter une chaîne qui a déjà été traduite dans les fichiers de ressources InstallAnywhere.

Pour disposer des clés de la nouvelle locale dans chaque projet d'installateur, il suffit de mettre à jour le texte statique. Pour disposer de ces clés uniquement dans le projet actuel, il suffit de mettre à jour le texte dynamique. Le texte dynamique est actualisé à chaque enregistrement du projet ; il est donc nécessaire de mettre à jour les fichiers chaque fois que le projet est modifié. Cette remarque justifie également de planifier la conception de l'installateur avant de lancer le processus de localisation.

Recommandations générales pour la localisation

Préparez la conception de l'installateur avant de traduire les fichiers de locale. Les modifications apportées ultérieurement à la conception de l'installateur risquent d'affecter le format et le contenu des fichiers de locale et d'entraîner des travaux supplémentaires de traduction.

Dans la mesure du possible, conservez le texte par défaut qui est également livré sous sa forme localisée.

Testez les installateurs sur des systèmes utilisant la locale cible pour éliminer les cas où certaines chaînes ne sont pas traduites dans les fichiers de locale.

Vérifiez que chaque ressource référencée dans les fichiers de locale est incluse dans l'installateur (y compris Contrat de licence, ReadMe/LisezMoi, etc.).

Codes de langue

Langue	Code de pays	Groupe de langues
Allemand	DE	Occidentales
Anglais	EN	Occidentales
Basque	EU	Occidentales
Catalan	CA	Occidentales
Chinois (simplifié)	ZH_CN	Orientales
Chinois (traditionnel)	ZH_TW	Orientales
Coréen	KO	Orientales
Danois	DA	Occidentales
Espagnol	ES	Occidentales
Finois	FI	Occidentales
Français (Canada)	FR_CA	Occidentales
Français (Standard)	FR	Occidentales
Grec	EL	Orientales
Hongrois	HU	Orientales
Indonésien	ID	Orientales
Italien	IT	Occidentales
Japonais	JA	Orientales
Néerlandais	NL	Occidentales
Norvégien	NO	Occidentales
Polonais	PL	Occidentales
Portugais (Brésil)	PT_BR	Occidentales
Portugais (Standard)	PT	Occidentales
Russe	RU	Orientales
Slovaque	SK	Occidentales

Slovène	SL	Occidentales
Suédois	SV	Occidentales
Tchèque	CS	Orientales
Thaï	TH	Orientales
Turc	TR	Orientales

Éléments localisables

Les installateurs déployés sur des systèmes non latins exigent une VM Java internationale.

Localiser les éléments de l'installateur

La liste ci-dessous indique les propriétés à modifier dans les fichiers de ressources linguistiques. Toutefois, cette liste ne présente pas toutes les propriétés des nouvelles actions ; pour obtenir la liste complète, contactez le Support technique Zero G.

Propriété	Définition
Installer.#.ProductName	Nom de produit affiché dans la barre de titre de l'installateur.
Installer.#.RulesFailedMessage	Message affiché si les règles spécifiées empêchent le fonctionnement de l'installateur.
Installer.#.ShortcutDestinationPathMacOS	Chemin du dossier dans lequel sont créés les alias lors d'une installation sous Mac OS (chemin relatif par rapport au dossier sélectionné par l'utilisateur lors de l'étape « Sélectionner le dossier des alias »).
Installer.#.ShortcutDestinationPathWin32	Chemin du dossier dans lequel sont créés les raccourcis lors d'une installation sous Windows (chemin relatif par rapport au dossier sélectionné par l'utilisateur lors de l'étape « Sélectionner un dossier d'alias/liens/raccourci »).
Installer.#.ShortcutDestinationPathSolaris	Chemin du dossier dans lequel sont créés les liens symboliques lors d'une installation sous UNIX (chemin relatif par rapport au dossier sélectionné par l'utilisateur).

	lors de l'étape « Sélectionner le dossier des liens »).
Installer.##.End-userSplashName	Nom du fichier image à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un simple nom de fichier et non d'un chemin d'accès absolu).
Installer.##.End-userSplashPath	Chemin d'accès du fichier image à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un chemin d'accès qui ne précise pas le nom du fichier).
InstallSet.##.Description	Description d'une des fonctions de l'installateur.
InstallSet.##.InstallSetName	Nom d'une des fonctions de l'installateur.
Intro.##.DisplayText	Texte à afficher pour présenter l'installateur (Introduction). Intro.##.Title : Titre à afficher pour présenter l'installateur (Introduction).
LicenseAgr.##.FileName	Nom du fichier contenant le Contrat de licence localisé à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un simple nom de fichier et non d'un chemin d'accès absolu).
LicenseAgr.##.Path	Chemin d'accès du fichier contenant le Contrat de licence localisé à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un chemin d'accès qui ne précise pas le nom du fichier).
LicenseAgr.##.Title	Titre associé au Contrat de licence de l'installateur.
MakeExecutable.##.DestinationName	Nom du fichier exécutable LaunchAnywhere à créer sur le système de destination. MakeRegEntry.##.Value : Valeur à écrire dans le registre Win32.

ShortcutLoc.#.MacOSTitle	Titre de l'étape Mac OS « Sélectionnez le dossier des alias » de l'installateur.
ShortcutLoc.#.SolarisTitle	Titre de l'étape UNIX « Sélectionner le dossier des alias/liens/raccourcis » de l'installateur.
ShortcutLoc.#.Win32Title	Titre de l'étape Windows « Sélectionner le dossier des alias/liens/raccourcis » de l'installateur.
Billboard.#.ImageName	Nom du fichier image de billboard à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un simple nom de fichier et non d'un chemin d'accès absolu).
Billboard.#.ImagePath	Chemin d'accès du fichier image de billboard à afficher lors de la préparation de l'installateur (remarque : il s'agit d'un chemin d'accès qui ne précise pas le nom du fichier).
ChooseInstallSet.#.Title	Titre associé à l'étape « Sélectionnez une type d'installation » de l'installateur.
ChooseJavaVM.#.Title	Titre associé à l'étape « Sélectionner une VM Java » de l'installateur.
CreateShortcut.#.DestinationName	Nom du raccourci/alias/lien à créer sur le système de destination. human.readable.language.name : Nom de la langue des données présentes dans ce fichier de ressources (français, allemand, espagnol, etc.).
ImportantNote.#.FileName	Nom du fichier texte à afficher pour l'étape « Remarque importante » de l'installateur (remarque : il s'agit d'un simple nom de fichier et non d'un chemin d'accès absolu).
ImportantNote.#.Path	Chemin du fichier texte à afficher pour l'étape « Remarque importante » de l'installateur

	(remarque : il s'agit d'un chemin d'accès qui ne précise pas le nom du fichier).
ImportantNote.#.Title	Titre de l'étape « Remarque importante » de l'installateur.
InstallBundle.#.BundleName	Nom du composant.
InstallBundle.#.Description	Texte décrivant le composant.
InstallComplete.#.DisplayText	Texte à afficher lors de l'étape « Installation terminée » de l'installateur.
InstallComplete.#.Title	Titre de l'étape « Installation terminée » de l'installateur.
InstallDir.#.Title	Titre de l'étape « Sélection du dossier d'installation » de l'installateur.
Installer.#.InstallerName	Nom de l'installateur.

Index

\$

\$INSTALL_STATUS\$35

A

Actions d'installation..... 17, 20, 62

Actions générales

 Actions 12, 64

Actions générales.....12

Actions par consoles

 Actions 12, 71

Actions par consoles.....12

Actions par panneaux

 Actions 12, 68, 73

Actions par panneaux12

Additions du panneau d'installateur 46,
73

Advanced Designer..... 7, 17, 20, 102

Affichage des messages64

Afficher du texte sur la console64

Afficher informations de débogage...64

Afficher le mot de passe..... 68, 71

Afficher le numéro de série..... 68, 71

Afficher les messages sur la console 71

Afficher message 68, 71

Aide73

Ajouter des fichiers 17, 20, 39, 50

Ajouter un label de saut.....64

Ant 64, 120

Arbre graphique.....50

Arguments d'installateur (facultatif).41

Assistant de projet

 Mode de l'installateur..... 8

Assistant de projet 7, 17, 20, 40

Associer des fichiers aux composants
.....30

B

Billboards..... 9, 46

C

Chemins source 33, 56

Choisir la classe main40

Cible du saut64

Classpath.....7, 17, 20, 50, 72

Code personnalisé 64, 68, 71, 99, 100,
101, 105, 126

Codes de langue128

Codes en sortie.....118

Commentaire64

Composant..... 30, 41

Composants 9, 30, 48, 50

Configuration..... 35, 41

Console Windows29

Construction du projet . 17, 20, 35, 40,
52, 117, 120, 126

Construire avec l'outil Ant120

Contrat de licence 68, 71

Convertir des projets IA antérieurs à
5.5..... 5

Copier fichier62

Copier un dossier62

Créer un désinstallateur62

Créer un dossier62

Créer un lanceur29

Créer un lanceur LaunchAnywhere
pour les applications Java62

Créer un lanceur pour une application
Java29

D

Débogage35, 41, 64, 100

Décompresser une archive62

Définir le classpath.....40

Définir le classpath automatiquement
..... 17, 40

Définir une variable d'environnement
système.....62

Définir une variable InstallAnywhere 64

Démarrer	
Arrêter	
Mise en pause du service Windows	64
Démarrer.....	64
Déplacer un dossier.....	62
Déplacer un fichier.....	62
Désinstallateur	
Code personnalisé.....	105
Fonctions.....	105
Désinstallateur 12, 17, 20, 46, 62, 68,	71, 73, 106
Désinstallateur – Introduction... 68, 71	
Désinstallation terminée.....	68, 71
Développement d’installateurs	
personnalisés par Zero G.....	3
Didacticiel avancé.....	20
Didacticiel d’introduction.....	17
Didacticiels.....	17
Distribution.....	20, 32, 52
Dock	
Mac OS X.....	113
Dock.....	113
DTS – Services techniques Zero G	
pour développeurs	
DTS.....	3
DTS – Services techniques Zero G	
pour développeurs.....	3
DVD.....	8, 52
E	
Écran d’accueil.....	9
Éléments localisables.....	129
Enregistrement.....	4, 56
Enregistrer un service Windows.....	62
Entrée utilisateur.....	68, 71
Environnement de développement	
d’installateurs.....	1
Environnement de l’installateur.....	1
Environnement requis.....	1
Environnements de développement	
d’installateurs.....	7
Équipe de développement.....	33
Évaluation de l’espace disque	
disponible.....	68
Exécutables LaunchAnywhere.....	35
Exécuter le fichier cible.....	20, 33, 64
Exécuter le script Ant.....	64
Exécuter un script.....	64
Exécuter une commande.....	64
Exécuter XSL Transform.....	64
F	
Fichier de manifeste.....	80
Fichier de projet.....	15
Fichiers.....	50
Fichiers de réponse.....	111
Fonctions.....	9, 30, 48
H	
Horodate.....	41
I	
Image de fond.....	9
Images.....	46
InstallAnywhere Enterprise Edition2, 9,	35, 39
InstallAnywhere Now!.....	2
InstallAnywhere Standard Edition..2, 9	
Installateur pour HP.....	62
Installateur pour Mac OS X.....	35
Installateur pour Windows.....	35
Installateur Pure Java.....	35
Installateurs CD.....	8, 52
Installateurs en mode Silencieux...111	
Installateurs par consoles.....	112
Installateurs Web.....	8, 52
Installation impossible.....	71
Installation terminée.....	68, 71
Installer à partir d’un manifeste.....	62
Installer Linux RPM.....	62
Installer un client PowerUpdate. 13, 62	
Installer un fichier.....	62
Installer un module de fusion.. 62, 107	

- Installer un paquetage Solaris.....62
- Installer un SpeedFolder62
- Installer une archive.....62
- Intégrer un environnement d'exécution
Java (JRE) à l'installateur29
- Interface d'installation 17, 46, 73
- J**
- Journal d'installation.....41
- L**
- Lancer le navigateur par défaut.....64
- LaunchAnywhere 6, 13, 17, 20, 29, 35,
41, 50, 74
- LAX
 - Exécutable LaunchAnywhere . 20, 29,
35
- LAX.....20
- Lecture de l'entrée dans le registre
Windows64
- Libre-service 3
- Liste des étapes d'installateur ... 46, 73
- Locales..... 9, 35, 41, 101, 125, 126
- Localisation
 - Recommandations générales..... 126
- Localisation 9, 125, 126
- M**
- Mac OS X
 - Débogage35
 - Dépannage.....35
- Mac OS X.....13, 17, 20, 113
- Mac OS X – Conditions requises 1
- Magic Folders 13, 113
- Message défilant68
- Mise à jour du registre Windows.....62
- Modèles100, 109
- Modification du fichier texte.....64
- Modules de fusion 15, 107
- N**
- Nettoyer les composants30
- O**
- Optimiser un installateur UNIX32
- Options d'écrasement41
- Options de construction120
- Options de l'installateur120
- Outils de construction117
- P**
- Panneau de progression d'installation
.....46
- Paramètres d'image.....73
- Paramètres des labels.....73
- Personnaliseur
 - Propriétés communes.....72
- Personnaliseur 20, 72
- Plate-forme32, 52, 114, 120
- Plates-formes41
- Plug-in12
- Plug-ins102
- Post 12, 20
- PowerUpdate 6, 13, 56, 62
- Pré..... 12, 20, 68, 71
- Premium Support 3
- Prêt pour installation71
- Projet
 - Tâche41
- Projet..7, 15, 17, 35, 39, 41, 118, 120
- R**
- Rechercher un composant dans le
registre..... 30, 64
- Rechercher un fichier.....68
- Recommandations générales pour la
création de composants
 - Composants30
- Recommandations générales pour la
création de composants.....30
- Réduire la taille de l'installateur32
- Règles 20, 41, 48, 50, 99
- Relancer Windows64
- Remarque importante.....68
- Remplissage automatique.....46
- S**
- Sélectionner la VM Java 68, 71

Sélectionner le dossier d'installation	20, 68, 71
Sélectionner le dossier des liens	71
Sélectionner les fonctions à désinstaller	68, 71
Sélectionner les icônes	46
Sélectionner un alias	
Dossier des raccourcis	20
Lien	
Raccourci	68
Sélectionner un alias	68
Sélectionner un dossier	68
Sélectionner un fichier	68
Sélectionner un type de désinstallateur	68, 71
Sélectionner une classe main	17
Sélectionnez un type d'installation	68
Services de formation Zero G	3
Services professionnels	3
SpeedFolders	14, 62
stderr	29, 35, 41, 72
stdout	29, 35, 41, 72
Structure	9, 30, 48
Succès de l'installation	35
Support technique	3
Supprimer un dossier	62
Supprimer un fichier	62
T	
Tâche	
Aide	46
Billboards	46
Cibles de la construction	52
Configuration	41
Construction du projet	52
Description	41
Distribution	52
Infos	41
Interface d'installation	46
Java	41
Journal de construction	52
Locales	41
Modules de fusion	52
Params d'affichage	46
Params de fichier	41
Plates-formes	41
Projet	41
Règles	41
Structure	48
Tâche	46
Taille mémoire (heap) pour VM de l'installateur	41
Tester l'installateur	40
Type d'installation	48, 68
Types d'installation	9, 48, 71
U	
UNIX	
Débogage	35
UNIX	17, 113
V	
Variables InstallAnywhere	14, 35, 99, 102
Variables publiées	107
Versions d'InstallAnywhere	2
VM Java – Conditions requises	1
W	
Windows	
Comportement des fichiers en service	41
Windows	13, 17, 20
Windows – Conditions requises	1
Windows NT	
Dépannage	35
Windows NT	35