

HP Application Lifecycle Management – Pratiques conseillées

à l'attention des professionnels d'ALM

Flux de travail - Pratiques conseillées

Date de publication de la documentation : Décembre 2014
Date de sortie du logiciel : Décembre 2014



Mentions légales

Garantie

Les seules garanties relatives aux produits et services HP sont décrites dans les déclarations de garantie expresses accompagnant lesdits produits et services. Rien dans le présent document ne doit être considéré comme constituant une garantie supplémentaire. HP ne saurait être tenu pour responsable des erreurs techniques ou éditoriales ni des omissions que pourrait comporter le présent document.

Les informations du présent document peuvent être modifiées sans aucun préavis.

Légende relative aux droits restreints

Logiciel informatique confidentiel. La possession, l'utilisation et la copie sont interdites sans licence valide délivrée par HP. Conformément à FAR 12.211 et 12.212, les logiciels commerciaux, la documentation des logiciels et les données techniques relatives aux articles commerciaux disposent d'une licence accordée au gouvernement des États-Unis conformément aux conditions de licence commerciale standard HP.

Avis de copyright

© Copyright 2002 - 2012 Hewlett-Packard Development Company, L.P.

Mentions relatives aux marques

Microsoft® et Windows® sont des marques de Microsoft Corporation déposées aux États-Unis.

Oracle® est une marque déposée d'Oracle et/ou de ses filiales aux États-Unis.

Mises à jour de la documentation

La page de titre de ce document contient les informations d'identification suivantes :

- le numéro de version du logiciel, qui indique la version du logiciel ;
 - le numéro avant le point correspond au numéro de la mise à jour majeure ;
 - le premier numéro après le point correspond au numéro de la mise à jour mineure ;
 - le second numéro après le point représente le numéro de la mise à jour mineure-mineure ;
- la date de version du document, qui change à chaque mise à jour du document ;
- la date de sortie du logiciel, qui indique la date de sortie de cette version du logiciel.

Pour vérifier l'existence de nouvelles mises à jour ou vérifier que vous utilisez l'édition la plus récente, rendez-vous à l'adresse URL suivante :

<http://h20230.www2.hp.com/selfsolve/manuals>

Pour accéder à ce site, vous devez créer un compte HP Passport et vous connecter comme tel. Pour obtenir un identifiant HP Passport, accédez à l'adresse :

<http://h20229.www2.hp.com/passport-registration.html>

Vous pouvez également cliquer sur le lien New users - please register (Nouveaux utilisateurs - Inscrivez-vous) de la page de connexion à HP.

Vous recevrez également des versions récentes ou mises à jour si vous vous abonnez au service d'assistance du produit approprié. Pour plus d'informations, contactez votre représentant HP.

Assistance

Vous pouvez visiter le site Web en ligne de l'assistance HP Software à l'adresse :

www.hp.com/go/hpsoftwaresupport

Ce site Web fournit des informations de contact et des détails sur les produits, les services et l'assistance fournis par HP Software.

L'assistance logicielle en ligne de HP Software propose des fonctions de résolution autonome. Elle offre un moyen rapide et efficace d'accéder aux outils interactifs de support technique nécessaires à la gestion de votre entreprise. En tant que client bénéficiant de l'assistance HP, vous pouvez effectuer les opérations suivantes :

- effectuer des recherches dans les documents qui vous intéressent ;
- soumettre des incidents et suivre leur résolution, ainsi que des demandes d'amélioration ;
- télécharger des correctifs logiciels ;
- gérer vos contrats d'assistance ;
- rechercher des contacts d'assistance HP ;
- consulter les informations sur les services disponibles ;
- entamer des discussions avec d'autres utilisateurs du logiciel ;
- rechercher des formations et vous y inscrire.

La plupart des domaines de l'assistance nécessitent que vous soyez enregistré en tant qu'utilisateur HP Passport. Ils peuvent également nécessiter un contrat d'assistance actif. Pour de plus amples informations sur les niveaux d'accès à l'assistance, rendez-vous à l'adresse URL suivante :

http://h20230.www2.hp.com/new_access_levels.jsp

Pour obtenir un identifiant HP Passport, accédez à l'adresse URL suivante :

<http://h20229.www2.hp.com/passport-registration.html>

Table des matières

1 Flux de travail - Présentation	9
Importance du flux de travail	9
Étapes communes	9
2 Consignes pour la personnalisation du flux de travail.....	12
Options de personnalisation du projet	12
Règles de flux de travail générales.....	14
Environnement de test	15
Débogage du code du flux de travail.....	16
Flux de travail - À faire et À ne pas faire	17
Utilisation d'une API dans le flux de travail	25
3 Événements de flux de travail	29
Général	29
Modules communs	31
Modules d'entité.....	41
4 Conclusions	56

Bienvenue dans ce manuel

Bienvenue dans ce manuel des pratiques conseillées pour le flux de travail HP.

Ce manuel fournit des concepts, des consignes et des exemples pratiques pour la mise en œuvre des flux de travail la mieux adaptée dans différentes organisations.

À propos des flux de travail

De nouvelles technologies, architectures, tendances métier et attentes de l'utilisateur sont en train de changer la nature même des applications. Par conséquent, les applications elles-mêmes doivent évoluer. De nouveaux outils et de nouvelles architectures sont apparus pour accélérer et simplifier le développement et la production d'applications composites, d'applications Internet riches et de services Web 2.0 interactifs. De nouveaux processus, comme le développement Agile, sont mis en œuvre dans l'espoir de rendre plus facile et d'accélérer la création d'applications flexibles.

HP Application Lifecycle Management (ALM) constitue une solution de gestion du cycle de vie des applications complète, couvrant toutes les phases de la vie du logiciel. La mise en œuvre de pratiques et solutions ALM permet aux équipes logiciel de répondre aux grandes attentes et exigences des entreprises. La suite HP ALM permet à différentes entreprises de répondre à leurs besoins spécifiques selon le secteur d'activité, l'orientation et les processus de l'entreprise, le nombre d'applications et le type, etc. Aucune entreprise ne fonctionne de manière identique, y compris dans le même domaine et dans des circonstances semblables. La possibilité de personnaliser les projets HP ALM de diverses manières afin de répondre aux besoins en processus métier de votre organisation est, par conséquent, un important aspect de chaque mise en œuvre.

L'un des outils les plus importants fournis par HP ALM est la fonction de création de scripts intégrée qui permet de définir, contrôler et gérer les flux de travail exécutés dans le projet. L'administrateur du projet peut rédiger des scripts de flux de travail pour personnaliser l'interface utilisateur d'HP ALM et pour contrôler les actions que les utilisateurs peuvent effectuer.

L'objectif de ce document est d'aider les clients HP ALM à évaluer leurs pratiques actuelles en matière de personnalisation, et de créer et de mettre à jour avec succès des scripts de flux de travail à l'aide des fonctions avancées fournies par HP ALM. Tous les aspects de ce processus ont été étudiés en utilisant les données de pratiques conseillées et l'expertise de sources diverses, y compris des administrateurs de système d'exploitation de HP, services professionnels de HP, documentations techniques, livres d'experts de l'industrie et de l'expérience personnelle des organismes de test clients. Ces consignes permettent de réduire le temps initial de création et de tirer au maximum profit de l'utilisation d'HP ALM.

Public visé

Ce manuel est destiné aux personnes suivantes :

- Administrateurs de projet
- Administrateurs de modèle
- Spécialistes de la personnalisation

Conditions préalables

Pour utiliser ce manuel, vous devez bien connaître les principales phases du cycle de vie du développement des logiciels. Les processus métier d'organisations informatiques réelles doivent également vous être familiers.

Des connaissances opérationnelles et des privilèges administratifs d'HP ALM sont essentiels pour la mise en œuvre de ces pratiques conseillées.

Structure

Ce manuel est organisé comme suit :

- *Flux de travail - Présentation*
- *Consignes pour la personnalisation du flux de travail*
- *Événements de flux de travail*
- *Conclusions*

Commentaires

Si vous avez des questions, des commentaires ou des informations utiles sur les pratiques conseillées que vous souhaitez partager, envoyez un message à l'adresse e-mail suivante :

alm_cust_feedback@hp.com

1 Flux de travail - Présentation

Importance du flux de travail

Aucune entreprise n'est la même, les différents processus métier, les différentes affiliations industrielles, méthodologies de développement, technologies existantes et nouvelles utilisées imposent la nécessité d'appliquer la mise en œuvre unique de HP ALM. La philosophie d'une solution uniforme n'existe guère dans les organisations informatiques réelles. Par conséquent, chaque client HP ALM tire finalement avantage de la flexibilité fournie par les scripts de flux de travail.

Toutefois, les scripts de flux de travail peuvent avoir un impact significatif sur les performances du projet et du site en général. Par conséquent, il est essentiel de développer un code de flux de travail logique et organisé. Il est également essentiel de mettre en œuvre un processus solide pour développer et gérer le code de flux de travail.

Les sections suivantes expliquent en détail les étapes correctes d'un flux de travail.

Étapes communes

Comprendre les besoins du projet

Avant de créer ou de modifier un code de flux de travail, il est important de comprendre la structure du projet, la méthodologie d'utilisation du projet, les processus de l'organisation et les différentes personnes concernées.

Pour garantir la réussite du code du flux de travail, saisissez d'abord les exigences de chaque groupe ou personne, et déterminez un flux de travail prenant en compte l'ensemble des groupes. Identifiez les dénominateurs communs pour créer un processus combiné qui réponde aux besoins globaux.

Créer un document sur les exigences des flux de travail

Avant de vous lancer dans la rédaction du code, commencez par le document sur les exigences, qui doit contenir les personnalisations du flux de travail planifiées. Ce document vise à définir une personnalisation initiale. Une fois le flux de travail dans l'environnement de production, vous devez mettre à jour régulièrement ce document pour inclure toutes les modifications mises en œuvre.

Ce document doit inclure les éléments suivants :

- Le processus complet du flux de travail.
- La fonctionnalité requise du flux de travail.

Voici certains exemples :

- Processus de vérification des exigences : votre organisation peut demander que chaque exigence soit vérifiée et approuvée avant d'être associée à un test.
 - Actions que les utilisateurs ou les groupes peuvent effectuer en fonction de leurs autorisations.
 - Envoyer une notification par e-mail lorsqu'une modification spécifique est apportée à un champ.
- La présentation et le format.

Par exemple, vous pouvez souhaiter déterminer les champs disponibles lors de la création d'une anomalie, en définissant différents ensembles de champs par groupe d'utilisateurs, emplacements de champ dans différents onglets, etc.

Le document doit ensuite être approuvé par toutes les parties prenantes concernées.

Rédiger le code du flux de travail

Une fois le document de personnalisation du flux de travail approuvé, commencez à rédiger le code du flux de travail dans un environnement de test. Pour plus d'informations sur l'environnement de test, voir la section [Environnement de test](#).

Tester le code du flux de travail

Invitez les utilisateurs à valider vos modifications dans l'environnement de test. Assurez-vous que l'application du flux de travail correspond à leurs besoins.

Transmettre le flux de travail à l'environnement de production

Le flux de travail est appliqué côté client. Pendant la connexion, les fichiers de personnalisation et du flux de travail sont téléchargés sur l'ordinateur client local dans le répertoire suivant : %temp%\TD_80.

Lorsque vous transmettez le code du flux de travail à l'environnement de production, vous devez vous déconnecter, puis vous reconnecter pour accéder aux dernières modifications apportées à la personnalisation et au flux de travail.

Gérer les demandes de flux de travail

Pour bien contrôler votre code de flux de travail, notamment lorsque de nombreuses personnes sont impliquées, vous devez définir un système de gestion des nouvelles demandes de flux de travail.

Ce système va vous permettre d'effectuer le suivi des demandes, de comprendre les besoins métier associés, les conséquences de la modification, son importance, l'étendue de la demande (combien d'utilisateurs requièrent la fonction), etc. Il vous permet également d'envoyer des notifications et le statut de progression des demandes.

Une solution possible consiste à définir un projet ALM, dédié à gérer les nouvelles demandes de flux de travail.

2 Consignes pour la personnalisation du flux de travail

Options de personnalisation du projet

Les fonctions de scripts de flux de travail de HP ALM reposent sur différentes sections de personnalisation, décrites ci-dessous. Avant de rédiger le code, identifiez tous les autres besoins de personnalisation dans Personnalisation du projet. Ils permettront d'appliquer le flux de travail.

Groupes d'autorisations

Pour protéger un projet contre tout accès non autorisé, ALM vous permet d'affecter chaque utilisateur à un ou plusieurs groupes d'utilisateurs. ALM comprend des groupes prédéfinis avec des privilèges par défaut. Chaque groupe peut accéder à des fonctions ALM données.

Vous pouvez créer un groupe basé sur les privilèges d'un groupe existant. Si vous choisissez un groupe existant qui possède des privilèges d'accès semblables à ceux du nouveau groupe d'utilisateurs que vous souhaitez créer, vous réduisez le degré de personnalisation à effectuer.

Notez que la définition d'autorisations basées sur un groupe d'utilisateurs permet non seulement d'imposer l'accès, mais aussi d'envoyer des e-mails, des notifications, etc.

HP **ne recommande pas** d'affecter un utilisateur à plusieurs groupes d'utilisateurs.

Listes de projets

Un projet ALM contient un ensemble de listes prédéfinies qui sont utilisées pour la personnalisation du projet par défaut, par exemple, les statuts d'anomalie et les listes Oui, Non. Certaines listes peuvent être personnalisées pour prendre en charge des processus particuliers utilisés dans votre entreprise. D'autres listes ne peuvent pas être personnalisées, car la logique

interne système ALM utilise les valeurs de la liste. Vous pouvez également créer des listes définies par l'utilisateur, contenant des valeurs que vous pouvez entrer dans un champ de liste de recherche.

Entités de projet

Les entités constituent les blocs de construction des projets ALM. Les entités contiennent des données entrées par les utilisateurs pour un processus de gestion d'application donné, et ces données sont stockées dans des tables. Une entité peut représenter des objets de travail quelconques, par exemple, des exigences, tests, étapes de conception, pièces jointes ou anomalies.

La personnalisation du projet permet de définir des attributs et des propriétés pour les entités ALM, par exemple, les champs requis, en lecture seule et du type Vérifier la valeur. Chaque entité contient des champs ALM par défaut, appelés *champs système*. Elles peuvent également inclure des *champs utilisateur* que vous pouvez créer. Un champ utilisateur peut avoir l'un des types suivants :

- Liste d'utilisateurs (liste de tous les utilisateurs dans le projet)
- Liste
- Nombre
- Date
- Chaîne

Dans Personnalisation du projet, vous pouvez définir des propriétés pour chaque entité du projet, comme définir les champs qui doivent être remplis par les utilisateurs, et pour lesquels l'historique des données de champ est consigné. Certaines propriétés de ce type peuvent être définies également à l'aide du flux de travail. Il est recommandé de définir le comportement par défaut dans Personnalisation du projet et de le modifier uniquement pour des cas spécifiques à l'aide des scripts de flux de travail.

Chaque entité est limitée à 99 champs définis par l'utilisateur. C'est pour cette raison que HP **recommande** de travailler ensemble avec toutes les parties prenantes pour inclure les champs qui répondent aux besoins de la plupart de ces parties prenantes et qui ne deviendront pas obsolètes après une courte période.

Types d'exigence

Vous pouvez définir des types d'exigence pour votre projet. Un type d'exigence définit les champs facultatifs et les champs définis par l'utilisateur disponibles. Cela permet de créer des champs définis par l'utilisateur qui sont disponibles uniquement pour les exigences d'un type spécifique.

Règles de flux de travail générales

À l'aide du code du flux de travail, vous pouvez personnaliser davantage votre projet. Vous pouvez définir les paramètres suivants :

- Champs visibles et requis
- Ordre d'affichage des champs dans une boîte de dialogue
- Champs à afficher dans chaque onglet de la boîte de dialogue
- Listes à affecter à des champs spécifiques
- Valeurs par défaut pour des champs spécifiques
- Dépendances entre des valeurs de champs

Vous pouvez définir ces paramètres en fonction d'un groupe d'utilisateurs.

Remarques importantes :

- Le code du flux de travail passe outre les paramètres définis dans la catégorie de personnalisation donnée dans Personnalisation du projet.
- Vous pouvez effectuer certains types de personnalisation, comme la définition de règles de transition pour des groupes d'utilisateurs ou la définition de propriétés de champ pour des types d'exigence, soit en utilisant la page Personnalisation du projet spécifique, soit par l'intermédiaire du code de flux de travail. Il est **recommandé** de choisir une seule méthode pour cette personnalisation et de ne pas combiner les deux.
- À l'aide de la fonction Message auto., ALM vous permet d'informer les utilisateurs automatiquement par e-mail dès que des changements sont apportés aux champs d'anomalie spécifiés. À l'aide de la fonction `SendMail_AfterPost`, le flux de travail vous permet de définir une notification automatique pour toutes les entités de projet, d'ajouter des conditions complexes ou de l'utiliser pour des utilisateurs ou des groupes d'utilisateurs spécifiques. Il est **recommandé** de vous assurer que vous ne

créez pas de conflits entre la fonction Message auto. et la fonction du flux de travail.

- Les scripts de flux de travail permettent de contrôler les actions d'entrée et de sortie des modules, ainsi que de limiter les autorisations d'accès des modules. Pour empêcher un groupe d'utilisateurs d'accéder à un module spécifique, HP **recommande** d'utiliser la page Accès au module dans Personnalisation du projet. N'empêchez pas l'accès aux modules à l'aide des scripts du flux de travail, car il existe un conflit avec la fonction Accès au module.

Environnement de test

Avant de procéder à la personnalisation du flux de travail dans votre environnement de production, HP **recommande** de valider la fonction de personnalisation dans un environnement de test qui reflète votre configuration spécifique.

Description de l'environnement de test

L'environnement de test est différent de l'environnement de production et le reflète exactement. Il simule les configurations et les applications installées sur le système de production, y compris le serveur de base de données, les logiciels et les projets de production. En testant le flux de travail dans votre environnement de test, vous pouvez obtenir un meilleur aperçu des résultats que vous pouvez atteindre, tout en identifiant et en empêchant toute incidence négative sur votre environnement de production.

Raisons pour la définition d'un environnement de test

Le flux de travail influe énormément sur le fonctionnement de votre projet. HP **recommande** de définir un environnement de test pour les raisons exposées ci-après :

- Il est recommandé de tester le flux de travail avant application réelle.
- Si le flux de travail échoue, aucun impact négatif n'est réellement à déplorer étant donné que l'environnement de test est indépendant de l'environnement de production. L'impact négatif éventuel dans l'environnement de production peut se traduire par une perte de

données, un blocage de fonction en raison d'erreurs de flux de travail, etc.

- Identification et détection précoces des problèmes.
- Vérification par les parties prenantes du processus planifié.

Débogage du code du flux de travail

Vous pouvez déboguer le code du flux de travail de différentes façons :

Option	Fonctionnalité de l'outil	Utilisé pour
Ajout de MsgBox via le flux de travail ALM	<p>Fonction intégrée du code de flux de travail ALM.</p> <p>Vous pouvez ajouter <code>Msgbox</code> (boîte de message) n'importe où dans le flux de travail pour afficher la valeur du champ, le nom de l'action, l'emplacement dans le code, etc.</p> <p>Si vous utilisez cette méthode, il est recommandé de commenter la ligne de traitement d'erreur dans le code, comme suit :</p> <pre>On Error Resume Next (Sur erreur, reprendre au suivant)</pre>	Débogage d'un emplacement spécifique dans le code
Dbgview	<p>Ce moniteur libre effectue le débogage de la sortie sur votre système local et imprime tous les événements de l'application.</p> <p>Pour n'afficher que les événements du flux de travail, il suffit de filtrer à l'aide du mot flux.</p> <p>Notez que cette option vous permet de voir uniquement les événements intégrés et non les fonctions supplémentaires que vous avez ajoutées.</p> <p>Fournisseur : Microsoft (SysInternals)</p>	Affichage des procédures du flux de travail appelées par ALM

	<p>URL :</p> <p>http://technet.microsoft.com/fr-fr/sysinternals/bb896647</p>	
Microsoft Visual Studio	<p>Produit commercial permettant de valider votre code à l'aide de points d'arrêt, de variables d'inspection, etc.</p> <p>Pour joindre le code du flux de travail à Visual Studio, générez une erreur dans le code du flux de travail au point d'attache souhaité pour Visual Studio. Vous pouvez pour cela appeler une sous-routine qui n'existe pas. Lorsque Visual Studio est installé, cette erreur dans le script apparaît sous forme de boîte de dialogue pour vous permettre de joindre le code au script.</p> <p>Notez que vous devez ajouter le commentaire <code>On Error Resume Next</code> (Sur erreur, reprendre au suivant) dans votre code de flux de travail pour utiliser cette option.</p>	<ul style="list-style-type: none"> • Débogage d'un emplacement spécifique dans le code • Affichage du flux du code que vous avez rédigé. • Observation du code en exécution

Outre les outils ci-dessus, vous pouvez appliquer votre propre enregistreur d'événements. HP **recommande** d'appliquer l'option d'activation et désactivation selon vos besoins de l'enregistreur d'événements, si vous utilisez le vôtre, pour éviter tout effet négatif sur les performances.

Flux de travail - À faire et À ne pas faire

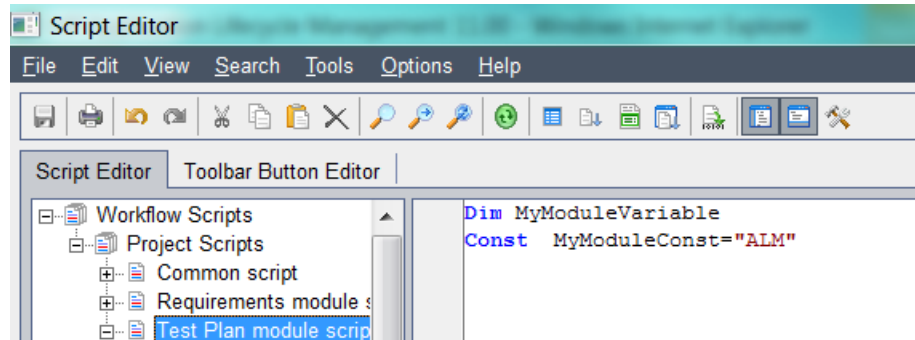
Les pratiques communes **recommandées** ci-après peuvent vous aider à maîtriser les flux de travail.

À faire

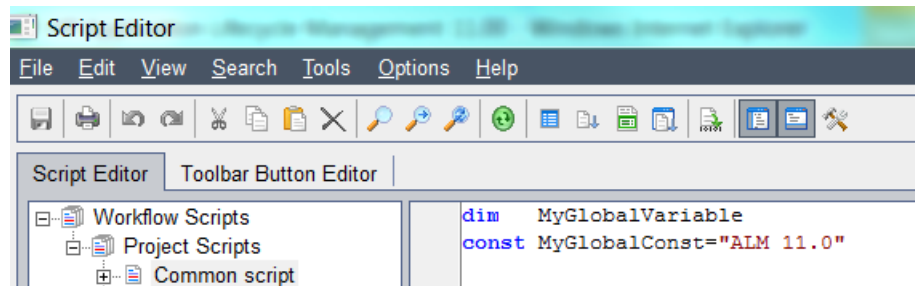
Utiliser des variables ou constantes globales

Pour transmettre des valeurs entre les différents événements, il est nécessaire d'utiliser des variables globales ou des constantes globales.

La variable peut exister au niveau du module. Elle peut être utilisée dans les événements du module.



Pour transmettre des valeurs entre les différents événements de module, il est nécessaire de définir la variable globale dans le module commun.



Optimiser le code

Lors de la programmation en VBScript, il vous arrive de devoir reprogrammer le même code. En général, cela indique que vous devez utiliser une fonction pour réduire la répétition de code :

- Utilisez des procédures et des fonctions au lieu du code obsolète.

Les exemples de tâches communes suivants constituent de bons sujets pour appliquer des fonctions ou procédures séparées :

- Définition de propriétés de champ

- Définition de champs sur le formulaire
 - Définition de dépendances de liste
 - Toute procédure fonctionnant avec l'API OTA
 - Utilisez une instruction de commutation au lieu d'instructions ElseIf répétitives.
- En règle générale, s'il existe deux conditions ElseIf ou plus, vous devez utiliser une instruction de commutation.

Améliorer la lisibilité du code

Outre la réutilisation de code répétitif, les fonctions permettent également d'augmenter la lisibilité du code. Les pratiques conseillées suivantes sont **recommandées** pour vous aider à obtenir une meilleure lisibilité du code :

- Utilisez des lignes vides pour séparer logiquement les blocs de code associés.
- Utilisez les commentaires d'introduction (en-tête) de la première instruction de variable et la dernière variable déclarée du code lui-même.
- Tous les commentaires doivent être précédés d'une ligne vide.
- Indentez le code et les commentaires dans une procédure à l'aide de deux à quatre espaces de tabulation. (L'éditeur Visual Basic utilise quatre espaces de tabulation par défaut.) Tout comme les espaces blanches, les indentations permettent d'organiser le code de manière logique et de le rendre visuellement agréable. La liste suivante contient des consignes générales pour l'utilisation correcte de l'indentation afin d'obtenir un code plus facile à lire et à maintenir :
 - Indentez le code et les commentaires dans une procédure à l'aide d'au moins une tabulation. Les lignes de code à ne pas indenter sont celles de début et de fin de la procédure, ainsi que les étiquettes de ligne utilisées avec votre gestionnaire d'erreurs.
 - Si vous utilisez des sauts de ligne pour organiser la liste d'arguments d'une procédure, utilisez des tabulations pour indenter les arguments et leurs instructions de type données, et les aligner au premier argument de la liste.
 - Indentez les variables déclarées à l'aide d'une tabulation. Déclarez une variable par ligne.

- Indentez les structures de contrôle à l'aide d'au moins une tabulation. Si une structure de contrôle est intégrée dans une autre, indentez la structure intégrée à l'aide d'une tabulation. Indentez le code dans une structure de contrôle à l'aide d'une tabulation supplémentaire.
- Si vous utilisez un caractère de continuation de ligne pour rompre une ligne de code, indentez la nouvelle ligne à l'aide d'une tabulation supplémentaire. Cela permet de voir que les deux lignes (ou plus) sont associées. Si la ligne qui suit la ligne de continuation est indentée au même niveau que cette dernière, ajoutez une tabulation ou plus à cette ligne pour la distinguer de la ligne suivante.
- Indentez les commentaires au même niveau que le code auquel ils appartiennent.

Commenter le code

Utilisez des modèles de commentaire comme celui ci-dessous pour documenter votre code :

```
'#####
'#Date:
'#Designer:
'#Purpose:
'#####
```

Accéder aux champs par nom

L'accès à un champ ou à plusieurs dans une entité donnée est une tâche commune :

- Utilisez l'instruction `<Entity>_Fields.Field("<Field_Name>")` pour accéder à un champ par son nom.
- Utilisez la répétition sur `<Entity>_Fields.FieldById(i)` pour accéder à tous les champs dans l'ensemble. Cela permet, par exemple, de réinitialiser l'ordre des champs.

Les deux méthodes conviennent pour travailler avec les champs de l'entité en cours.

L'entité en cours peut être définie comme suit :

- Entité en cours

Entité pour laquelle l'événement actuel est déclenché. La plupart des événements pointent vers le type d'entité pour laquelle les champs peuvent être récupérés. Par exemple, dans les événements `Defects_Bug_`, vous ne pouvez manipuler que le champ `Bug_Fields`, et dans les événements `TestPlan_DesignStep_` vous ne pouvez accéder qu'au champ `DesignStep_Fields`.

— Élément sélectionné

De toutes les entités de l'ensemble défini par l'événement, uniquement les champs de l'entité actuellement sélectionnée peuvent être récupérés. Par exemple, le test sur lequel se trouve le curseur ou l'exécution en cours dans l'exécuteur manuel.

Pour récupérer les champs d'autres objets du même type ou d'un autre type, utilisez l'API OTA.

Réinitialiser avant de définir une nouvelle disposition

Assurez-vous de réinitialiser la disposition de *tous les champs* avant de définir la disposition des *champs spécifiques*, tels que `PageNo` et `ViewOrder`.

Les champs présentant un ordre prédéfini par défaut, il est important de réinitialiser cet ordre avant d'en définir un nouveau, personnalisé. Sinon, un champ non souhaité peut apparaître au même emplacement et fausser l'ordre des autres champs.

Voir [Exemple de flux de travail](#) plus loin dans ce document.

Sauvegarder le code

HP **recommande** de sauvegarder régulièrement votre code de flux de travail. Copiez et collez l'intégrité ou une partie de votre script de flux de travail dans un fichier texte pour l'enregistrer dans le système de fichiers.

Utiliser des constantes globales au lieu de noms de champ

Utilisez des constantes globales au lieu de noms de champ pour améliorer la lisibilité du code.

Vous devez déclarer une constante globale pour chaque nom de champ (voir la section [Utiliser des variables ou constantes globales](#) plus haut).

Exemple :

```
If Bug_Fields.Field(Bug_Status).Value="Closed" then
```

```
        Bug_Fields.Field(Bug_Closed_In_Version).isRequired = true
    End if
```

Nettoyer les objets

Assurez-vous de nettoyer les objets à la fin de leur étendue. Pour chaque instance d'objet, il est essentiel de nettoyer les objets non utilisés. Vous améliorez ainsi les performances d'ALM et évitez les erreurs.

```
set myTDConnection = TDConnection
set myTDConnection = nothing
```

Normaliser

HP **recommande** d'appliquer la normalisation à tous les projets. L'administrateur du projet ALM, responsable de plusieurs projets, doit utiliser des conventions communes aux projets. Cela contribue à la lisibilité et maintenance du code, ainsi qu'à l'utilisation de la fonction interprojets.

Traiter les erreurs

L'un des facteurs les plus importants susceptible d'affecter la qualité du script du flux de travail est l'application correcte du traitement des erreurs.

Voici quelques pratiques simples **recommandées** pour le traitement des erreurs, permettant de mieux contrôler le comportement de l'application :

- Utiliser une instruction `On Error Resume Next` (Sur erreur, reprendre au suivant) au début de chaque procédure et fonction
- Utiliser l'instruction `On Error GoTo 0` (Sur erreur, accéder à) à la fin de chaque procédure et fonction
- Afficher les erreurs pour l'utilisateur dans une boîte de message standard

Le code doit être ajouté une fois à chaque script de flux de travail (Anomalies, Plan de test, etc.) comme dans la fonction `PrintError` ci-dessous.

Utilisez l'objet Visual Basic `Err` contenant les informations sur les erreurs d'exécution pour cela.

```
Sub GetBug1
    On Error Resume Next (Sur erreur, reprendre au suivant)
    Set Bug1 = TDConnection.BugFactory.item(1)
```

```

    PrintError("GetBug1")
End Sub
Sub PrintError(strFunctionName)
    If Err.Number <> 0 Then
        MsgBox "Error #" & Err.Number & ": " & Err.Description, _
            vbOKOnly, "Workflow Error in Function " & strFunctionName
    End If
End Sub

```

À ne pas faire

Dupliquer

Ne dupliquez pas le code dans l'événement `Entity_CanChange` et l'événement `EntityChange`.

Le code basé sur une modification de champ peut être rédigé dans l'événement `Entity_CanChange` ou l'événement `Entity_Change`. Assurez-vous de bien comprendre la différence entre ces événements et appliquez les simples règles ci-dessous dans votre code de flux de travail :

- Le code portant sur les autorisations (statut Autoriser la modification) doit être rédigé dans l'événement `Entity_CanChange`.
- Le code portant sur le traitement des valeurs de dépendance ou les listes de dépendances doit être rédigé dans l'événement `EntityChange`.

Utiliser trop de code

Trop de code dans l'événement `CanLogin` ou l'événement `EnterModule` a un impact sur les performances. Une erreur courante consiste à mettre à jour des entités pendant l'événement `CanLogin`.

Définir d'autres propriétés avant de définir la propriété Visible

Définissez la propriété `IsVisible` avant de définir la propriété `IsRequired` ou `IsReadOnly` du champ.

La définition de la propriété obligatoire ou en lecture seule du champ non visible à l'écran est inutile et ignorée par ALM. Il est donc important de vérifier qu'un champ est visible avant de définir ces propriétés.

Voir *Exemple de flux de travail* plus loin dans ce document.

Complicquer le code du flux de travail

Même si un projet ALM est utilisé pendant plusieurs années, le code du flux de travail doit toujours être clair et extensible.

- Choisir la sélection au lieu de la condition Si
- Utiliser des fonctions

Mettre à jour les paramètres

Ne mettez pas à jour les valeurs de paramètre à partir d'une fonction de flux de travail. Plus spécifiquement, ne mettez pas à jour le paramètre `NewValue` dans l'événement `Entity_CanChange`.

Apporter des modifications dans l'événement de nouvelle entité

Dans l'événement de nouvelle entité, il **n'est pas recommandé** de modifier des actions, car cet événement est appelé lorsque l'entité est *créée* et non lorsque la boîte de dialogue de la nouvelle entité est ouverte.

Un cas d'utilisation fréquent est lorsqu'un utilisateur ouvre pour la deuxième fois la boîte de dialogue de la nouvelle entité. L'événement de nouvelle entité n'est pas appelé car l'entité a été déjà créée lors de la première ouverture de la boîte de dialogue.

Affecter des valeurs à un champ dans l'événement MoveTo

N'affectez pas de valeurs à un champ dans l'événement `MoveTo`. Cette pratique n'est pas recommandée car l'événement `MoveTo` verrouille l'entité.

Apporter des modifications dans l'événement After_Post

N'apportez pas de modification d'objet dans l'événement `After_Post`.

Utiliser trop d'appels API

N'utilisez pas trop d'appels API, car cela augmente le niveau de communication entre le serveur et la base de données. Chaque appel d'API génère une action de communication réseau, ce qui ralentit l'exécution du script.

Par exemple, pour traiter 100 entités, regroupez-les dans un même filtre, au lieu de les récupérer une à une.

Utilisation d'une API dans le flux de travail

Utiliser une API ALM pour apporter des modifications

L'API ALM fournit une couche de séparation entre l'interface utilisateur (ou une application qui l'utilise) et la logique serveur. HP **recommande** de suivre ces règles lors de l'utilisation d'appels API :

- Utiliser un objet `TDConnection` prédéfini pour accéder à la session en cours

Lorsque vous utilisez l'API OTA à partir d'applications externes, telles que Visual Basic ou Excel, la première étape pour chaque application utilisant OTA consiste à créer l'instance de l'objet `TDConnection`, initialiser la connexion au serveur et établir la connexion à la base de données. Toutefois, le flux de travail contient l'objet `TDConnection` prédéfini (dans ce cas, `TDConnection` n'est pas seulement un nom de classe, mais aussi la variable globale contenant l'instance de `TDConnection`), qui pointe vers la même session où travaille l'utilisateur actuel. Cela signifie que l'accès aux collections et objets ALM est toujours disponible depuis n'importe quel point dans le flux de travail.

- Il est recommandé d'éviter la mise à jour directe de la base de données à l'aide de l'objet `Command`, en raison des problèmes suivants susceptibles de produire :
 - Le contournement du mécanisme serveur entraîne les événements suivants :
 - Verrouillage d'entité
 - Perte d'historique
 - Autre fonctionnalité non souhaitée (configuration e-mail)

- Haute maintenance des requêtes
- Possible corruption ou incohérence des données
- Utilisez les méthodes de messagerie disponibles dans OTA pour envoyer des messages personnalisés aux utilisateurs.

OTA autorise l'accès à la messagerie ALM pour vous permettre d'effectuer les tâches suivantes :

- Créer des conditions personnalisées non applicables à l'aide du système de notification automatique d'ALM
- Modifier l'objet ou le texte de l'e-mail
- Envoyer un e-mail aux groupes ou utilisateurs ALM spécifiques
- Envoyer l'e-mail de la part d'un utilisateur spécifique et non de la part d'admin comme dans la notification de courrier automatique

Les méthodes de messagerie sont disponibles à partir de tout objet ALM, par exemple, Anomalies, Test, etc., ou directement à partir de l'objet `TDConnection`. La méthode `Mail`, à partir de l'objet `TestDirector`, vous permet d'envoyer l'e-mail contenant cet objet, ainsi que votre sujet et texte personnalisés.

Exemple - Utiliser des objets de flux de travail au lieu d'une instruction SQL

N'utilisez pas les commandes suivantes :

```
Com.CommandText = "UPDATE TESTCYCL SET TC_TESTER_NAME =
'" & Cstr(ASSIGNED_TESTER) & "' " &
"Where TC_CYCLE_ID = " & iTestSetId & " and TC_TESTER_NAME is
NULL"
```

```
Set UpdateRecSet = Com.Execute
```

Utiliser à la place cet extrait de code:

```
Set tstestF = currentTestSet.tstestFactory
Set tsFilter = tstestF.Filter
tsFilter("TC_TESTER_NAME")= ""
Set tsTestList = filter.newList
For each tsTest in tsTestList
    tsTest.Field("TC_TESTER_NAME") = "admin"
Next
```

Minimiser l'activité sur le client

Lorsque vous récupérez des données du serveur, il est **recommandé** de filtrer les informations côté serveur au lieu de côté client. L'impact sur les performances est très élevé lors du filtrage sur le client. Trop d'enregistrements chargés peuvent également avoir un impact sur les performances du serveur.

Exemple - Utiliser un filtre lors de l'utilisation de l'historique

Lors de l'utilisation de l'objet `Command` pour passer au tableau `HISTORY`, il est recommandé de créer un filtre dans SQL en appliquant la condition `WHERE`, pour éviter de charger tous les jeux d'enregistrements sur le client.

Exemple - Calcul des étapes de conception

L'étape de conception comporte un champ défini par l'utilisateur indiquant la durée de l'étape. L'objectif recherché consiste à obtenir le nombre d'étapes de conception avec une durée de plus de 30 minutes.

Le code ci-dessous représente une mauvaise pratique :

```
For Each Test In TestLists
    Set DesStepF = Test.DesignStepFactory
    Set DSList = DesStepF.NewList("")
    For Each DStep In DSList
        If DStep.Field("DS_USER_01")>30 Then
            HowManyFound = HowManyFound + 1
        End If
    Next
Next
```

Essayez plutôt d'utiliser le bloc de code suivant :

```
Set TestF = TDConnection.TestFactory
Set TestList = TestF.NewList("")
For Each Test In TestList
    Set DesStepF = Test.DesignStepFactory
    Set DSList = DesStepF.NewList("select * from DESSTEPS WHERE
DS_USER_01>30 ")
    HowManyFound = HowManyFound + DSList.count
```

Next

3 Événements de flux de travail

Lors d'une session utilisateur ALM, à mesure que l'utilisateur lance des actions, ALM déclenche des procédures d'événement. Vous pouvez placer du code dans ces procédures pour personnaliser l'exécution des actions utilisateur associées. Des fonctions ou des sous-routines peuvent représenter des procédures d'événement.

Général

La section suivante présente des informations générales sur les fonctions et les sous-routines d'événement, ainsi que sur les conventions de dénomination utilisées dans HP ALM.

Fonctions d'événement

Ces procédures sont déclenchées par ALM pour vérifier si l'action de l'utilisateur doit être effectuée. Vous pouvez placer du code dans ces fonctions pour déterminer si ALM peut exécuter la demande de l'utilisateur. Si le code renvoie une valeur faux, ALM ne poursuit pas l'action.

Par exemple, lorsqu'un utilisateur clique sur le bouton Soumettre dans la boîte de dialogue *Ajouter une anomalie*, ALM appelle la fonction `Bug_CanPost` avant de publier l'anomalie dans la base de données sur le serveur. Vous pouvez ajouter du code à la fonction `Bug_CanPost` pour contrôler si ALM publie l'anomalie. Par exemple, vous pouvez vérifier que l'utilisateur ne peut pas rejeter une anomalie sans ajouter un commentaire.

Sous-routines d'événement

Ces procédures sont déclenchées pour effectuer des actions lorsqu'un événement se produit.

Par exemple, lorsqu'un utilisateur ouvre la boîte de dialogue *Ajouter une anomalie*, ALM appelle la sous-routine `Bug_New`. Vous pouvez ajouter du code à la sous-routine `Bug_New` pour effectuer des actions qui doivent être

effectuées lorsqu'un utilisateur ouvre la boîte de dialogue. Par exemple, vous pouvez modifier la valeur du champ de *mode de détection* en BTW si l'utilisateur ne fait pas partie du groupe des testeurs de contrôle qualité.

Conventions d'affectation de noms

La convention d'affectation de noms dans HP ALM pour une procédure d'événement est la suivante :

<entity>_<event> (entité_événement)

Remarque : Certains noms de procédure d'événement, par exemple, `GetDetailsPageName`, n'incluent pas un nom d'entité.

Entités

Entité	Description
AnalysisItem	Données de rapports et de graphiques
AnalysisItemFolder	Données de dossier de rapports et de graphiques
Bug	Données d'anomalie
BusinessModel	Données de modèle métier
BusinessModelActivity	Données d'activités de modèle métier
BusinessModelFolder	Données de dossier de modèles métier
BusinessModelPath	Données de chemin de modèle métier
Composant	Données de composant métier
ComponentFolder	Données de dossier de composants métier
ComponentStep	Données d'étape de composant métier
DashboardFolder	Données de dossier de tableau de

	bord
DashboardPage	Données de page de tableau de bord
DesignStep	Données d'étape de conception
Ressource	Données de ressource de test
Dossier de ressources	Données de dossier de ressources de test
Exécuter	Données d'exécution de tests
Étape	Données d'étape d'exécution de tests
TestSet	Données de campagne de test
TestSetTests	Données d'instance de test

Le flux de travail peut également prendre en charge certaines extensions.

Modules communs

CanLogin

Cet événement est déclenché pour vérifier si l'utilisateur spécifié peut se connecter au projet spécifié. Il permet d'autoriser ou d'interdire la connexion à un projet. Cet événement peut être utilisé pour mettre à jour les utilisateurs.

Rubrique	Description
Syntaxe	CanLogin (DomainName, ProjectName, UserName) où DomainName est le nom du domaine, ProjectName est le nom du projet et UserName est le nom de l'utilisateur.
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	CanLogin (tous les modules)

Exemple - Notifier l'utilisateur lors de la connexion

```
Function CanLogin(DomainName, ProjectName, UserName)
CanLogin = false
Call MsgBox("Hi " & User.UserName & ", " _
           & vbCrLf & " " _
           & vbCrLf & "Your project " & TDConnection.ProjectName
           & " was upgraded to ALM 11.0" _
           & vbCrLf & " " _
           & vbCrLf & "The Project was moved to the server :
http://ALM:port/qcbin" _
           & vbCrLf & " " _
           & vbCrLf & "QC Admin" _
           , vbExclamation, "Important Message")
Exit function
End function
```

CanLogout

Cet événement est déclenché pour vérifier si l'utilisateur actuel peut se déconnecter du projet en cours.

Rubrique	Description
Syntaxe	CanLogout
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	CanLogout (tous les modules)

Exemple - Notifier l'utilisateur avant la déconnexion

```
Function CanLogout
Call MsgBox("Hi " & User.UserName & ", " _
           & vbCrLf & " " _
           & vbCrLf & "Your project " & TDConnection.ProjectName
           & " will be upgraded to ALM 11.0 on 01/01" _
```



```

        & vbCrLf & "" _
        & vbCrLf & "The Project will be moved to the server:"
http://ALM:port/qcbin" _
        & vbCrLf & "" _
        & vbCrLf & "QC Admin" _
        , vbExclamation, "Important Message")
End Function

```

ActionCanExecute

Cet événement est déclenché avant qu'ALM n'effectue une action initiée par l'utilisateur, pour vérifier que l'action peut être exécutée. Vous pouvez ajouter du code à cette procédure d'événement pour effectuer des actions lorsque l'utilisateur a initié une action donnée ou pour empêcher l'exécution de l'action dans certains cas.

Rubrique	Description
Syntaxe	<p>ActionCanExecute (ActionName) où ActionName est l'action initiée par l'utilisateur.</p> <p>Les actions suivent le format <code>contexte.action</code>.</p> <p>Les actions définies par l'utilisateur commencent par le préfixe <code>UserDefinedActions</code>.</p>
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	ActionCanExecute (tous les modules)

Exemple - Empêcher la suppression des anomalies

```

Function ActionCanExecute (ActionName)
On Error Resume Next (Sur erreur, reprendre au suivant)
if ActionName= "Defects.DeleteDefect" then
    if Bug_Fields.Field("BG_STATUS").value ="Closed" then
        ActionCanExecute = true

```

```

        Else
Msgbox "You don't have enough credentials to perform Delete"
        ActionCanExecute = false
        Exit function
    End if
End if
\.....
End function

```

Exemple - Rechercher les noms d'action

```

Function ActionCanExecute(ActionName)
On Error Resume Next (Sur erreur, reprendre au suivant)
if user.Username="Project_admin" then
    MsgBox actionname
End if
End function

```

EnterModule

Cet événement est déclenché lorsque l'utilisateur passe à un module ALM.

Vous pouvez ajouter du code à cette procédure d'événement pour effectuer une action lorsque l'utilisateur passe au module spécifié.

Rubrique	Description
Syntaxe	EnterModule
Type	Sous-type
Renvoie	
Disponibilité	EnterModule (tous les modules)

Exemple - Masquer un bouton

```

Sub EnterModule
'hides the button Send Mail in the Defects grid
    On Error Resume Next (Sur erreur, reprendre au suivant)

```

```

        Actions.action("Defects.SendByEmail").Visible= false
    On Error GoTo 0
End Sub

Sub DialogBox(DialogBoxName, IsOpen)
'hide the button Send Mail in the Defect details Dialog
'Use ActiveModule and ActiveDialogName to get the current context
On Error Resume Next (Sur erreur, reprendre au suivant)
if (DialogBoxName="actBugDetails" or DialogBoxName="Details" or
DialogBoxName="Bug Details") and IsOpen=true then
        Actions.action("BugDetails.SendByEmail").Visible= false
End if
    On Error GoTo 0
End Sub

```

ExitModule

Cet événement est déclenché lorsque l'utilisateur quitte le module spécifié.

Rubrique	Description
Syntaxe	ExitModule
Type	Sous-type
Renvoie	
Disponibilité	ExitModule (tous les modules)

DialogBox

Cet événement est déclenché lorsqu'une boîte de dialogue est ouverte ou fermée.

Rubrique	Description
Syntaxe	DialogBox (DialogBoxName, IsOpen) où DialogBoxName est le nom de la boîte de dialogue, et IsOpen indique

	si la boîte de dialogue est ouverte.
Type	Sous-type
Renvoie	
Disponibilité	DialogBox (tous les modules)

Exemple - Identifier un type de vue

Cet exemple aide à identifier le type de vue actuel : grille, détails, nouvelle entité. Le type est géré dans une variable globale dans le module commun, appelée DialogIsOpen.

```

Sub DialogBox(DialogBoxName, IsOpen)
On error resume next
If DialogBoxName="New Bug" and IsOpen=true then
    DialogIsOpen = "NEW"
Else
    DialogIsOpen ="OTHER" `Details Or Grid
End if
On Error GoTo 0
End sub

```

CanCustomize

Cet événement est déclenché lorsque l'utilisateur tente d'ouvrir la fenêtre Personnalisation, pour vérifier si l'utilisateur peut personnaliser le projet spécifié.

Rubrique	Description
Syntaxe	CanCustomize (DomainName, ProjectName, UserName) où DomainName est le nom du domaine, ProjectName est le nom du projet et UserName est le nom de l'utilisateur.
Type	Fonction
Renvoie	Vrai ou Faux

Disponibilité	CanCustomize (tous les modules)
----------------------	---------------------------------

Exemple - Entrée refusée dans la fenêtre de personnalisation

Cet exemple refuse l'entrée dans la fenêtre de personnalisation aux utilisateurs non autorisés.

```

Function CanCustomize(DomainName, ProjectName, UserName)
on error resume next
if User.IsInGroup("TDAdmin") then
    CanCustomize = true
else
    MsgBox User.FullName & vbcrLf & vbcrLf & "You don't have
enough privileges" & vbcrLf & vbcrLf & "Please Open a SR in Project
Center Admin", vbExclamation, "Not Allowed"
    CanCustomize = false
end if
On Error GoTo 0
End Function

```

Attachment_New

Cet événement est déclenché lorsqu'une pièce jointe est ajoutée à ALM.

Rubrique	Description
Syntaxe	Attachment_New (Attachment) où Attachment est l'interface IAttachment.
Type	Sous-type
Renvoie	
Disponibilité	Attachment_New (tous les modules)

Attachment_CanOpen

Cet événement est déclenché avant qu'ALM n'ouvre une pièce jointe à partir du serveur, pour vérifier si la pièce jointe peut être ouverte.

Rubrique	Description
Syntaxe	Attachment_CanOpen (Attachment) où Attachment est l'interface lAttachment.
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	Attachment_CanOpen (tous les modules)

Attachment_CanPost

Cet événement est déclenché avant qu'ALM ne mette à jour une pièce jointe existante sur le serveur, pour vérifier si la pièce jointe peut être mise à jour.

Rubrique	Description
Syntaxe	Attachment_CanPost (Attachment) où Attachment est l'interface lAttachment.
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	Attachment_CanPost (tous les modules)

Attachment_CanDelete

Cet événement est déclenché avant qu'ALM ne supprime une pièce jointe du serveur, pour vérifier si la pièce jointe peut être supprimée.

Rubrique	Description
Syntaxe	Attachment_CanDelete (Attachment) où Attachment est l'interface lAttachment.
Type	Fonction

Renvoie	Vrai ou Faux
Disponibilité	Attachment_CanDelete (tous les modules)

GetDetailsPageName

Cet événement est déclenché par ALM pour récupérer le nom de la page (onglet) dont le numéro d'index est spécifié sous PageNum dans les boîtes de dialogue suivantes :

- Boîte de dialogue Détails de l'entité
- Boîte de dialogue Nouvelle <entity> (entité)

Rubrique	Description
Syntaxe	GetDetailsPageName (PageName, PageNum) où PageName est le nom de page par défaut (par exemple, Page 1) et PageNum est le numéro de page. Remarque : Le numéro de page représente le numéro de page absolu, quelle que soit la position relative de la page par rapport aux autres pages affichées dans la boîte de dialogue.
Type	Fonction
Renvoie	Chaîne contenant le numéro de page
Disponibilité	GetDetailsPageName (tous les modules)

Exemple - Mettre à jour un onglet

```
Function GetDetailsPageName (PageName, PageNum)
    On Error Resume Next (Sur erreur, reprendre au suivant)
    Select Case activemodule
        Case "Requirements"
```

```

        Select Case PageNum
            Case 1
                GetDetailsPageName="Req_Details-First Tab"
            Case 2
                GetDetailsPageName="Req_Details-Second Tab"
            Case 3
                GetDetailsPageName="Req_Details-Third Tab"
        End select
    Case "Defects"
        Select Case PageNum
            Case 1
                GetDetailsPageName="Def_Details-First Tab"
            Case 2
                GetDetailsPageName="Def_Details-Second Tab"
            Case 3
                GetDetailsPageName="Def_Details-Third Tab"
        End select
    End select
    On Error GoTo 0
End Function

Function GetNewBugPageName (PageName , PageNum)
On Error Resume Next (Sur erreur, reprendre au suivant)
    Select Case PageNum
        Case 1
            GetNewBugPageName="Def_Details-First Tab"
        Case 2
            GetNewBugPageName="Def_Details-Second Tab"
        Case 3
            GetNewBugPageName="Def_Details-Third Tab"
    End select
    On Error GoTo 0
End Function

```


Modules d'entité

Entity_New

Cet événement est déclenché lorsqu'un objet est ajouté à ALM. Vous pouvez ajouter du code à cette procédure d'événement pour effectuer une action lorsqu'un nouvel objet est ajouté.

Rubrique	Description
Syntaxe	<entity>_New
Type	Sous-type
Renvoie	
Disponibilité	AnalysisItem_New AnalysisItemFolder_New Baseline_New Bug_New BusinessModelFolder_New BusinessModelPath_New Component_New ComponentFolder_New ComponentStep_New Cycle_New DashboardFolder_New DashboardPage_New DesignStep_New Library_New LibraryFolder_New Release_New ReleaseFolder_New Req_New Resource_New ResourceFolder_New Step_New Test_New TestConfiguration_New TestFolder_New

	TestSet_New TestSetFolder_New
--	----------------------------------

Entity_MoveTo

Cet événement est déclenché lorsque l'utilisateur déplace la mise en évidence d'un objet à un autre.

Vous pouvez ajouter du code à cette procédure d'événement pour effectuer des actions lorsque l'utilisateur déplace la mise en évidence.

Rubrique	Description
Syntaxe	<entity>_MoveTo
Type	Sous-type
Renvoie	
Disponibilité	AnalysisItem_MoveTo AnalysisItemFolder_MoveTo Baseline_MoveTo Bug_MoveTo BusinessModel_MoveTo BusinessModelActivity_MoveTo BusinessModelFolder_MoveTo BusinessModelPath_MoveTo Component_MoveTo ComponentFolder_MoveTo (anciennement MoveToComponentFolder) ComponentStep_MoveTo Cycle_MoveTo DashboardFolder_MoveTo DashboardPage_MoveTo DesignStep_MoveTo Library_MoveTo LibraryFolder_MoveTo Release_MoveTo ReleaseFolder_MoveTo Req_MoveTo Resource_MoveTo

	ResourceFolder_MoveTo
	Run_MoveTo
	Step_MoveTo
	Test_MoveTo
	TestConfiguration_MoveTo
	TestFolder_MoveTo
	TestSet_MoveTo
	TestSetFolder_MoveTo
	TestSetTests_MoveTo

Exemple - Mettre à jour une configuration lors du déplacement

Mettez à jour la configuration lors du déplacement vers une autre entité.

```

Sub Bug_MoveTo
    Select Case Bug_Fields.Field("BG_STATUS").value
        Case "New"
            Setup_Status_New
        Case "Open"
            Setup_Status_Open
        Case "Fixed"
            Setup_Status_Fixed
        Case "Closed"
            Setup_Status_Closed
    End select
End sub

```

Exemple - Liste de dépendances

Le code suivant montre comment modifier une liste associée à un champ, en fonction de la valeur d'un autre champ.

Supposons que des champs définis par l'utilisateur ont été ajoutés à l'entité d'exigence nommée *SUB_AREA* (RQ_USER_01) et *TESTING_AREA* (RQ_USER_02), et qu'une liste définie par l'utilisateur a été ajoutée par zone de test sous le nom *SUB_LIST_<testing area>* (zone de test).

Ce code doit être appelé dans l'événement *<entity>_MoveTo* et *<entity>_FieldChange*.

```
Req_Fields.field("RQ_USER_02").List = Lists("SUB_LIST_" &
Req_Fields.field("RQ_USER_01").value)
```

Entity_FieldCanChange

Cet événement est déclenché avant qu'ALM ne modifie une valeur de champ, pour déterminer si le champ peut être modifié.

Vous pouvez ajouter du code à cette procédure d'événement pour empêcher la modification d'un champ dans des cas particuliers.

Rubrique	Description
Syntaxe	<entity>_FieldCanChange (FieldName, NewValue) où FieldName est le nom du champ et NewValue est la valeur du champ.
Type	Fonction
Retourne	Vrai ou Faux
Disponibilité	AnalysisItem_FieldCanChange AnalysisItemFolder_FieldCanChange Baseline_FieldCanChange Bug_FieldCanChange BusinessModel_FieldCanChange BusinessModelActivity_FieldCanChange BusinessModelFolder_FieldCanChange BusinessModelPath_FieldCanChange Component_FieldCanChange ComponentFolder_FieldCanChange ComponentStep_FieldCanChange Cycle_FieldCanChange DashboardFolder_FieldCanChange DashboardPage_FieldCanChange DesignStep_FieldCanChange Library_FieldCanChange LibraryFolder_FieldCanChange Release_FieldCanChange ReleaseFolder_FieldCanChange Req_FieldCanChange

	Resource_FieldCanChange
	ResourceFolder_FieldCanChange
	Run_FieldCanChange
	Step_FieldCanChange
	Test_FieldCanChange
	TestConfiguration_FieldCanChange
	TestFolder_FieldCanChange
	TestSet_FieldCanChange
	TestSetFolder_FieldCanChange
	TestSetTests_FieldCanChange

Exemple - Autoriser ou refuser une modification

Cette fonction permet d'attribuer ou non, à certains groupes d'utilisateurs, l'autorisation de modifier le champ *Statut* dans l'anomalie selon sa valeur actuelle ou nouvelle.

```

Function Bug_FieldCanChange(Fieldname, NewValue)
On Error Resume Next (Sur erreur, reprendre au suivant)
if FieldName = "BG_STATUS" then
    if User.IsInGroup("QATester") then
        if Bug_Fields.Field("BG_STATUS").value ="Fixed" then
            Select Case NewValue
                Case "Fixed", "Closed"
                    Bug_FieldCanChange = true
                Case else
                    Bug_FieldCanChange = false
            End select
            Exit function
        End select
    End if
End if
End if
On Error GoTo 0
End Function

```

Entity_FieldChange

Cet événement est déclenché lorsque la valeur du champ spécifié change. Chaque modification de valeur déclenche l'événement de modification de champ lorsque le champ perd sa mise en évidence.

Vous pouvez ajouter du code à cette procédure d'événement pour effectuer une action lorsque la valeur d'un champ donné est modifiée. Par exemple, vous pouvez masquer ou afficher un champ en fonction de la valeur que l'utilisateur entre dans un autre champ.

Rubrique	Description
Syntaxe	<entity>_FieldChange (FieldName) où FieldName est le nom du champ.
Type	Sous-type
Renvoie	
Disponibilité	AnalysisItem_FieldChange AnalysisItemFolder_FieldChange Baseline_FieldChange Bug_FieldChange BusinessModel_FieldChange BusinessModelActivity_FieldChange BusinessModelFolder_FieldChange BusinessModelPath_FieldChange Component_FieldChange ComponentFolder_FieldChange ComponentStep_FieldChange Cycle_FieldChange DashboardFolder_FieldChange DashboardPage_FieldChange DesignStep_FieldChange Library_FieldChange LibraryFolder_FieldChange Release_FieldChange ReleaseFolder_FieldChange Req_FieldChange Resource_FieldChange ResourceFolder_FieldChange

	Run_FieldChange Step_FieldChange Test_FieldChange TestConfiguration_FieldChange TestFolder_FieldChange TestSet_FieldChange TestSetFolder_FieldChange TestSetTests_FieldChange
--	--

Exemple - Valeurs de dépendance

Lors de la modification du statut d'un test à la valeur *Automatiser*, une description de modèle est ajoutée.

```

Sub Test_FieldChange(FieldName)
On Error Resume Next (Sur erreur, reprendre au suivant)
  if Test_Fields.Field("TS_STATUS").Value="To Automate" then
    if Test_Fields.Field("TS_DESCRIPTION").value="" then
      myComments="<html><body><b>TO AUTOMATE-" & Now & "/" Checked
by " & user.UserName & "</b><br></body></html>"
      Test_Fields.Field("TS_DESCRIPTION").value= myComments
    Else
      myComments="<br><b>TO AUTOMATE-" & Now & "/" Checked by "&
user.UserName & "</b><br>"
      Test_Fields.Field("TS_DESCRIPTION").value =
Test_Fields.Field("TS_DESCRIPTION").value & "<br> "&
myComments
    End if
  End if
On Error GoTo 0
End Sub

```

Exemple - Mettre à jour une configuration lors de la modification

Mettez à jour la configuration lorsqu'un champ, par exemple, le statut d'une anomalie, change :

```

Sub Bug_FieldChange(FieldName)
On Error Resume Next (Sur erreur, reprendre au suivant)

```

```

If FieldName="BG_STATUS" then
  Select Case Bug_Fields.Field("BG_STATUS").value
    Case "New"
      Setup_Status_New
    Case "Open"
      Setup_Status_Open
    Case "Fixed"
      Setup_Status_Fixed
    Case "Closed"
      Setup_Status_Closed
  End select
End if
On Error GoTo 0
End Sub

```

Voir aussi l'exemple [Liste de dépendances](#).

Entity_ CanPost

Cet événement est déclenché avant qu'ALM ne publie un objet sur le serveur, pour vérifier si l'objet peut être publié.

Vous pouvez ajouter du code à cette procédure d'événement pour empêcher la publication d'un objet dans des cas particuliers.

Rubrique	Description
Syntaxe	<entity>_CanPost
Type	Fonction
Renvoi	Vrai ou Faux
Disponibilité	AnalysisItem_CanPost AnalysisItemFolder_CanPost Baseline_CanPost Bug_CanPost BusinessModel_CanPost BusinessModelFolder_CanPost BusinessModelPath_CanPost

Component_CanPost
ComponentFolder_CanPost
Cycle_CanPost
DashboardFolder_CanPost
DashboardPage_CanPost
Library_CanPost
LibraryFolder_CanPost
Release_CanPost
ReleaseFolder_CanPost
Req_CanPost
Resource_CanPost
ResourceFolder_CanPost
Run_CanPost
Step_CanPost
Test_CanPost
TestConfiguration_CanPost
TestFolder_CanPost
TestSet_CanPost
TestSetFolder_CanPost

Exemple - Désactiver la mise à jour

Si une exigence est finalisée sans commentaire, l'utilisateur n'est pas autorisé à la soumettre.

```
Function Req_CanPost
```

```
On Error Resume Next (Sur erreur, reprendre au suivant)
```

```
if Req_Fields.Field("RQ_REQ_PRIORITY").IsModified then
```

```
    if Req_Fields.Field("RQ_DEV_COMMENTS").IsModified=false then
```

```
        Req_CanPost=false
```

```
        MsgBox "The priority was updated, you have to add a comment"
```

```
        Exit function
```

```
    End if
```

```
End if
```

```
On Error GoTo 0
```

```
End Function
```

Entity_CanDelete

Cet événement est déclenché avant qu'ALM ne supprime un objet du serveur, pour vérifier si l'objet peut être supprimé.

Rubrique	Description
Syntaxe	<entity>_CanDelete
Type	Fonction
Renvoie	Vrai ou Faux
Disponibilité	AnalysisItem_CanDelete AnalysisItemFolder_CanDelete Baseline_CanDelete Bug_CanDelete BusinessModel_CanDelete BusinessModelFolder_CanDelete BusinessModelPath_CanDelete Component_CanDelete ComponentFolder_CanDelete Cycle_CanDelete DashboardFolder_CanDelete DashboardPage_CanDelete Library_CanDelete LibraryFolder_CanDelete Release_CanDelete ReleaseFolder_CanDelete Req_CanDelete Resource_CanDelete ResourceFolder_CanDelete Test_CanDelete TestConfiguration_CanDelete TestFolder_CanDelete TestSet_CanDelete TestSetFolder_CanDelete

Entity_AfterPost

Cet événement est déclenché après la publication d'un objet sur le serveur. Une fois publiés, les champs de projet ne doivent plus être modifiés, car les nouvelles valeurs ne sont pas stockées dans la base de données.

Rubrique	Description
Syntaxe	<entity>_AfterPost
Type	Sous-type
Renvoie	
Disponibilité	AnalysisItem_AfterPost AnalysisItemFolder_AfterPost Baseline_AfterPost Bug_AfterPost BusinessModel_AfterPost BusinessModelFolder_AfterPost BusinessModelPath_AfterPost Component_AfterPost ComponentFolder_AfterPost Cycle_AfterPost DashboardFolder_AfterPost DashboardPage_AfterPost Library_AfterPost LibraryFolder_AfterPost Release_AfterPost ReleaseFolder_AfterPost Req_AfterPost Resource_AfterPost ResourceFolder_AfterPost Run_AfterPost Step_AfterPost Test_AfterPost TestConfiguration_AfterPost TestFolder_AfterPost TestSet_AfterPost TestSetFolder_AfterPost

Exemple - Envoyer un e-mail

Une notification est envoyée par e-mail à l'auteur de l'exigence si le champ Mise à jour cible est modifié. Pour envoyer un e-mail, vous devez ajouter une fonction personnalisée, appelée sendreqmail.

```
Sub Req_AfterPost
If Req_Fields.field("RQ_TARGET_RCYC").IsModified Then
    Sendreqmail Req_Fields.field("RQ_REQ_ID").Value,
    Req_Fields.field("RQ_REQ_AUTHOR").Value, "", "Target Cycle has
    changed", "Please Review"
End if
End sub

Sub sendreqmail (ReqId,Mto,cc,msubject,mcomment)
Dim tdc, bgf, bg
    Set tdc = TDConnection
    Set rf = tdc.ReqFactory
    Set req = rf.Item(ReqId)
    req.Mail mto , cc, 2, mSubject, mComment

    Set req = Nothing
    Set rf = Nothing
    Set tdc = Nothing
End sub
```

Exemple de flux de travail - Définir une configuration

Ce code de flux de travail permet de mettre à jour les propriétés de champ suivantes : visibilité, obligatoire, lecture seule et ordre.

Ajoutez ces fonctions au nœud Module Anomalies.

Vérifier les utilisateurs

Vérifiez si l'utilisateur fait partie d'un groupe particulier avant de prendre une décision sur l'action suivante.

```
If User.IsInGroup("Developer") then
    Mygroup="DEV"
```

```
End if
```

Définir l'apparence du champ

Cette sous-routine permet de définir l'apparence du champ : visibilité, statut obligatoire, numéro de page et ordre à l'écran.

```
Sub SetFieldApp( FieldName, Vis, Req, PNo, VOrder )  
    With Bug_Fields(FieldName)  
        .IsVisible = Vis  
        .IsRequired = Req  
        .PageNo = PNo  
        .ViewOrder = VOrder  
    End With  
End Sub
```

Réinitialiser aux valeurs initiales

Ajoutez la sous-routine suivante pour masquer tous les champs d'anomalie.

```
Sub ResetMetadata  
    For i=0 to Bug_Fields.Count  
        Bug_Fields.FieldById(i).IsVisible = false  
    Next  
End sub
```

Définir le statut

La sous-routine suivante permet de définir le statut Nouveau selon les autorisations de l'utilisateur.

Vous devez rédiger cette sous-routine pour chaque statut.

```
Sub Setup_Status_New  
If User.IsInGroup("Developer") then  
    Mygroup="DEV"  
ElseIf User.IsInGroup("QATester") then  
    Mygroup="QA"  
ElseIf User.IsInGroup("Documentation") then  
    Mygroup="DOC"
```

```

End if

Call ResetMetadata `set to initial status
Select case Mygroup
  Case "DEV"
    SetFieldApp "BG_ACTUAL_FIX_TIME", True, False, 0, 0
    SetFieldApp "BG_CLOSING_DATE", True, False, 0, 1
    SetFieldApp "BG_CLOSING_VERSION", True, False, 0, 2
    SetFieldApp "BG_DETECTED_BY", True, True, 0, 3
    SetFieldApp "BG_DETECTED_IN_RCYC", True, False, 0, 4
    SetFieldApp "BG_DETECTED_IN_REL", True, False, 0, 5
    SetFieldApp "BG_DETECTION_DATE", True, True, 0, 6
    SetFieldApp "BG_DETECTION_VERSION", True, False, 0, 7
    SetFieldApp "BG_ESTIMATED_FIX_TIME", True, False, 0, 8
    SetFieldApp "BG_PLANNED_CLOSING_VER", True, False, 0, 8
    SetFieldApp "BG_PRIORITY", True, False, 0, 10
  Case " QA"
    SetFieldApp "BG_ACTUAL_FIX_TIME", True, False, 0, 0
    SetFieldApp "BG_CLOSING_DATE", True, False, 0, 1
    SetFieldApp "BG_CLOSING_VERSION", True, False, 0, 2
    SetFieldApp "BG_DETECTED_BY", True, True, 0, 3
    SetFieldApp "BG_DETECTED_IN_RCYC", True, False, 0, 4
    SetFieldApp "BG_DETECTED_IN_REL", True, False, 0, 5
    SetFieldApp "BG_DETECTION_DATE", True, True, 0, 6
    SetFieldApp "BG_DETECTION_VERSION", True, False, 0, 7
  Case "DOC"
    SetFieldApp "BG_ACTUAL_FIX_TIME", True, False, 0, 0
    SetFieldApp "BG_CLOSING_DATE", True, False, 0, 1
    SetFieldApp "BG_CLOSING_VERSION", True, False, 0, 2
    SetFieldApp "BG_DETECTED_BY", True, True, 0, 3
    SetFieldApp "BG_DETECTED_IN_RCYC", True, False, 0, 4
End Select
End sub

```

4 Conclusions

La demande de logiciels pertinents et performants favorise l'innovation et la réussite de l'entreprise. La criticalité métier croissante des logiciels, combinée à l'émergence de tendances complexes et évolutives, telles que la virtualisation et le cloud, continuent à faire avancer l'amélioration des processus.

HP ALM répond aux besoins du cycle de vie des applications modernes en offrant un meilleur alignement des équipes, y compris l'intégration dans les équipes de stratégie et planification, en offrant des pratiques conseillées pour stimuler l'innovation et éviter les délais tactiques, ainsi qu'une passerelle vers la dernière étape critique de l'organisation des opérations. HP ALM est extensible et dynamique, prêt à s'adapter à la nature dynamique d'ALM. Sa souplesse permet de couvrir diverses industries, du secteur pharmaceutique à la construction automobile, divers types de développement, du type classique en cascade au type moderne Agile, diverses structures organisationnelles, de la structure plate à la structure hiérarchique ou matricielle, etc.

À bien des égards, ce résultat découle des nombreuses fonctionnalités de personnalisation intégrées dans le produit et qui fournissent les outils nécessaires pour différencier les processus métier propres à chaque organisation adoptant ALM. La création de scripts de flux de travail accorde à l'administrateur du projet le pouvoir d'adapter les procédures et les écrans standard aux besoins spécifiques du projet. Le présent document permet de mieux comprendre les modèles d'utilisation, expose les avantages et les inconvénients des différentes méthodes de codage, donne des précisions sur les événements les plus communs et contient de nombreux exemples pratiques pour vous aider à rédiger le code.

Nous pensons que les pratiques conseillées répertoriées dans ce document facilitent l'adoption du flux de travail HP ALM dans votre organisation.