

DevTest Solutions

Prise en main

Version 8.0



La présente documentation, qui inclut des systèmes d'aide et du matériel distribués électroniquement (ci-après nommés "Documentation"), vous est uniquement fournie à titre informatif et peut être à tout moment modifiée ou retirée par CA.

La présente Documentation ne peut être copiée, transférée, reproduite, divulguée, modifiée ou dupliquée, en tout ou partie, sans autorisation préalable et écrite de CA. La présente Documentation est confidentielle et demeure la propriété exclusive de CA. Elle ne peut pas être utilisée ou divulguée, sauf si (i) un autre accord régissant l'utilisation du logiciel CA mentionné dans la Documentation passé entre vous et CA stipule le contraire ; ou (ii) si un autre accord de confidentialité entre vous et CA stipule le contraire.

Nonobstant ce qui précède, si vous êtes titulaire de la licence du ou des produits logiciels décrits dans la Documentation, vous pourrez imprimer ou mettre à disposition un nombre raisonnable de copies de la Documentation relative à ces logiciels pour une utilisation interne par vous-même et par vos employés, à condition que les mentions et légendes de copyright de CA figurent sur chaque copie.

Le droit de réaliser ou de mettre à disposition des copies de la Documentation est limité à la période pendant laquelle la licence applicable du logiciel demeure pleinement effective. Dans l'hypothèse où le contrat de licence prendrait fin, pour quelque raison que ce soit, vous devrez renvoyer à CA les copies effectuées ou certifier par écrit que toutes les copies partielles ou complètes de la Documentation ont été retournées à CA ou qu'elles ont bien été détruites.

DANS LES LIMITES PERMISES PAR LA LOI APPLICABLE, CA FOURNIT LA PRÉSENTE DOCUMENTATION "TELLE QUELLE", SANS AUCUNE GARANTIE, EXPRESSE OU TACITE, NOTAMMENT CONCERNANT LA QUALITÉ MARCHANDE, L'ADÉQUATION À UN USAGE PARTICULIER, OU DE NON-INFRACTION. EN AUCUN CAS, CA NE POURRA ÊTRE TENU POUR RESPONSABLE EN CAS DE PERTE OU DE DOMMAGE, DIRECT OU INDIRECT, SUBI PAR L'UTILISATEUR FINAL OU PAR UN TIERS, ET RÉSULTANT DE L'UTILISATION DE CETTE DOCUMENTATION, NOTAMMENT TOUTE PERTE DE PROFITS OU D'INVESTISSEMENTS, INTERRUPTION D'ACTIVITÉ, PERTE DE DONNÉES OU DE CLIENTS, ET CE MÊME DANS L'HYPOTHÈSE OÙ CA AURAIT ÉTÉ EXPRESSÉMENT INFORMÉ DE LA POSSIBILITÉ DE TELS DOMMAGES OU PERTES.

L'utilisation de tout produit logiciel mentionné dans la Documentation est régie par le contrat de licence applicable, ce dernier n'étant en aucun cas modifié par les termes de la présente.

CA est le fabricant de la présente Documentation.

Le présent Système étant édité par une société américaine, vous êtes tenu de vous conformer aux lois en vigueur du Gouvernement des Etats-Unis et de la République française sur le contrôle des exportations des biens à double usage et aux autres réglementations applicables et ne pouvez pas exporter ou réexporter la documentation en violation de ces lois ou de toute autre réglementation éventuellement applicable au sein de l'Union Européenne.

Copyright © 2014 CA. Tous droits réservés. Tous les noms et marques déposées, dénominations commerciales, ainsi que tous les logos référencés dans le présent document demeurent la propriété de leurs détenteurs respectifs.

Support technique

Pour une assistance technique en ligne et une liste complète des sites, horaires d'ouverture et numéros de téléphone, contactez le support technique à l'adresse <http://www.ca.com/worldwide>.

Table des matières

Chapitre 1: Portail DevTest	9
Ouverture du portail DevTest	9
Accès au portail DevTest	10
Page d'accueil du portail DevTest	17
Chapitre 2: Didacticiels de CA Application Test	19
Didacticiel 1 : projets, scénarios de test et propriétés.....	20
Etape 1 : démarrage de DevTest Workstation	20
Etape 2 : création d'un projet	21
Etape 3 : création d'un scénario de test.....	21
Etape 4 : ajout d'une propriété à la configuration de projet	22
Etape 5 : ajout d'une étape de test	23
Etape 6 : ajout d'un message de journal.....	24
Etape 7 : ajout d'un second message de journal.....	25
Etape 8 : exécution de l'étape Ma publication de message de journal	26
Etape 9 : observation des valeurs de propriété	26
Etape 10 : exécution de l'étape Output Log Message (Message de journal de sortie).....	27
Résumé.....	27
Didacticiel 2 : ensembles de données	28
Etape 1 - création d'un ensemble de données.....	29
Etape 2 : création d'un scénario de test.....	29
Etape 3 : ajout d'une propriété à la configuration de projet	30
Etape 4 : ajout d'une étape de test pour l'étape Output Log Message (Message de journal de sortie)	31
Etape 5 : création d'une autre étape Output Log Message (Message de journal de sortie)	32
Etape 6 : exécution du test	33
Etape 7 : ajout de l'ensemble de données	34
Etape 8 : modification du comportement de l'ensemble de données.....	35
Didacticiel 2 : résumé	36
Didacticiel 3 : filtres et assertions	36
Etape 1 : création d'un scénario de test à partir d'un scénario de test existant.....	36
Etape 2 : modification d'une action de l'étape de test	37
Etape 3 : ajout d'une assertion	38
Etape 4 : test de l'assertion	40
Etape 5 : ajout d'un filtre	42
Etape 6 : test du filtre.....	43
Didacticiel 3 : résumé	45

Didacticiel 4 : manipulation d'objets Java (POJO)	45
Etape 1 : création d'un scénario de test.....	46
Etape 2 : création d'une étape de test Dynamic Java Execution (Exécution Java dynamique).....	47
Etape 3 : Appel sur l'objet Java	50
Etape 4 : ajout d'un filtre intégré	54
Etape 5 : vérification de la création de la propriété.....	56
Didacticiel 4 : résumé	57
Didacticiel 5 : exécution d'une application Web de serveur de démonstration	57
Etape 1 : lancement de l'application Web	58
Etape 2 - Connexion à l'application Web	59
Etape 3 : création d'un compte	60
Etape 4 - clôture d'un compte.....	63
Etape 5 : déconnexion de l'application Web.....	63
Didacticiel 5 : résumé	64
Didacticiel 6 : test d'un site Web.....	64
Didacticiel 6, partie A : enregistrement du scénario de test.....	65
Didacticiel 6, partie B : exécution du scénario de test	72
Didacticiel 6, partie C : modification des étapes de test de demande	73
Didacticiel 7 : test d'un objet Enterprise JavaBean (EJB)	79
Etape 1 : création d'un scénario de test.....	79
Etape 2 : création d'une configuration.....	80
Etape 3 : ajout d'une étape de test EJB.....	81
Etape 4 : connexion au serveur	81
Etape 5 : localisation de l'interface EJB.....	82
Etape 6 : configuration de l'étape EJB.....	84
Etape 8 : vérification de l'exécution de la méthode	88
Etape 9 : ajout d'une autre étape de test EJB	89
Didacticiel 7 : résumé	90
Didacticiel 8 : test d'un service Web	91
Etape 1 : création d'un scénario de test.....	91
Etape 2 : ajout d'une étape de test Web Service Execution (XML) (Exécution de service Web XML)	92
Etape 3 : création d'un client de service Web.....	94
Etape 4 : exécution de la demande de service Web	95
Etape 5 : affichage de la demande et de la réponse	96
Didacticiel 8 : résumé	97
Didacticiel 9 : examen et test d'une base de données.....	98
Etape 1 : création d'un scénario de test.....	98
Etape 2 : ajout de propriétés de base de données à la configuration	99
Etape 3 : ajout d'une étape de test SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC)	100
Etape 4 : connexion à la base de données	101
Etape 5 : exécution d'une requête SQL.....	102

Etape 6 : ajout d'une assertion	103
Etape 7 : exécution du scénario de test	104
Etape 8 : modification de l'assertion.....	105
Etape 9 : ajout d'un filtre	106
Etape 10 : test du filtre et de l'assertion	107
Didacticiel 9 : résumé	108
Didacticiel 10 : simulation d'un test rapide.....	109
Etape 1 : ouverture du scénario de test.....	109
Etape 2 : vérification du scénario de test.....	110
Etape 2, partie A : exécution d'un test rapide.....	111
Etape 2, partie B : affichage des rapports générés	114
Didacticiel 10 : résumé	115

Chapitre 3: Didacticiels sur les tests d'application mobile 117

Didacticiel 1 : enregistrement d'un scénario de test iOS	117
Etape 1 : démarrage de DevTest Workstation	118
Etape 2 : création d'un projet	118
Etape 3 : création d'un fichier de configuration pour un simulateur iOS	119
Etape 4 : création d'un actif de simulateur iOS.....	120
Etape 5 : enregistrement d'un scénario de test	122
Etape 6 : exécution du scénario de test dans l'ITR.....	123
Didacticiel 2 : enregistrement d'un scénario de test d'application Android	124
Etape 1 : démarrage de DevTest Workstation	124
Etape 2 : création d'un projet	125
Etape 3 : création d'un fichier de configuration pour un simulateur Android	125
Etape 4 : création d'un actif d'émulateur Android.....	126
Etape 5 : enregistrement d'un scénario de test	127
Etape 6 : exécution du scénario de test dans l'ITR.....	128

Chapitre 4: Didacticiel sur CA Service Virtualization 129

Conditions préalables.....	129
Procédure de création et de test d'une image de service virtuel.....	130
Etape 1 : démarrage de DevTest Workstation	130
Etape 2 : démarrage du VSE.....	131
Etape 3 : démarrage du serveur de démonstration	131
Etape 4 : exécution d'un scénario de test	132
Etape 5 : création d'un fichier de configuration.....	133
Etape 6 : activation du fichier de configuration	134
Etape 7 : configuration de l'enregistreur du VSE	135
Etape 8 : enregistrement du scénario de test	137
Etape 9 : déploiement du modèle de service virtuel	139

Etape 10 : test au niveau du modèle de service virtuel	141
Visualisation du didacticiel.....	142
Chapitre 5: Didacticiel sur CA Continuous Application Insight	143
Etape 1 - augmentation des niveaux de capture par défaut.....	144
Etape 2 - génération de transactions à partir de l'application de démonstration	145
Etape 3 - affichage des transactions dans la fenêtre d'analyse des transactions	146
Glossaire	149

Chapitre 1: Portail DevTest

Ce chapitre traite des sujets suivants :

- [Ouverture du portail DevTest](#) (page 9)
- [Accès au portail DevTest](#) (page 10)

Ouverture du portail DevTest

Ouvrez le portail DevTest à partir d'un navigateur Web.

Remarque : Pour plus d'informations sur les composants de serveur qui doivent être exécuté, reportez-vous à la section Démarrage des processus ou services DevTest de la rubrique *Installation*.

Procédez comme suit:

1. Effectuez l'une des opérations suivantes :
 - Dans un navigateur Web, entrez **http://localhost:1507/devtest**. Si le registre se trouve sur un ordinateur distant, remplacez **localhost** par le nom ou l'adresse IP de l'ordinateur.
 - Sélectionnez View (Afficher), DevTest Portal (Portail) à partir de la station de travail DevTest Workstation.
2. Entrez votre nom d'utilisateur et votre mot de passe.
3. Cliquez sur Log in (Connexion).

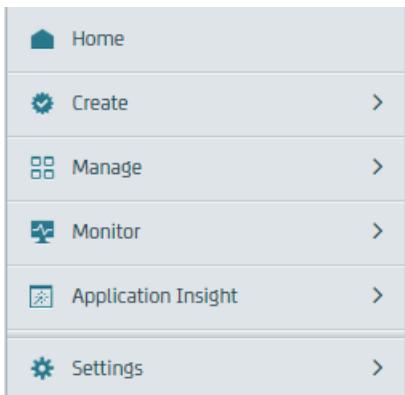
Accès au portail DevTest

Accédez au portail DevTest. Remplacez le nom d'hôte du DevTest Server sur lequel les processus ou services suivants sont exécutés : registre, portail, intermédiaire et VSE.

`http://nom-hôte:1507/devtest`

Connectez-vous au portail DevTest avec votre ID d'utilisateur et votre mot de passe. Si vous n'en avez pas, connectez-vous comme utilisateur standard. Contactez l'administrateur pour plus d'informations.

Utilisez la page d'accueil du portail DevTest pour accéder aux pages et réaliser les activités principales, ainsi qu'à la documentation correspondante. Vous pouvez développer la section de votre choix du volet de navigation pour afficher des liens vers les tâches pertinentes. Cliquez sur un lien pour afficher l'interface utilisateur pour une tâche sélectionnée, en général, un onglet du portail. Certains liens vous redirigent vers la console DevTest.



Dans le cas de liens que vous utiliserez souvent, vous pouvez cliquer sur l'option Quick Links (Liens rapides) correspondante. Pour obtenir de la documentation relative, cliquez sur le lien correspondant Getting Started (Prise en main). Les volets Getting Started (Prise en main) et Quick Links (Liens rapides) s'affichent dans l'onglet Home (Accueil).

Remarque : Lorsque vous cliquez la première fois sur un lien Getting Started (Prise en main), une boîte de dialogue de connexion s'ouvre. Votre connexion permettant d'afficher la documentation est conservée pendant votre session sur le portail DevTest.

The screenshot shows the DevTest Portal interface. At the top, there's a navigation bar with a menu icon, the 'ca technologies' logo, and the text 'DevTest Portal'. Below this is a sidebar with links: 'Home', 'Create >', 'Manage >', 'Monitor >', 'Application Insight >', and 'Settings >'. To the right of the sidebar are two panels: 'Getting Started' and 'Quick Links'. The 'Getting Started' panel contains a list of 10 items, each with an icon and text: 'How to record and save a virtual service', 'How to edit a virtual service', 'How to create a test', 'How to monitor a test', 'How to manage test artifacts', 'How to manage a ticket', 'How to work with the shelf', and 'How to document a transaction for a manual test'. The 'Quick Links' panel also contains a list of 8 items, each with an icon and text: 'Create a Virtual Service', 'Manage Virtual Services', 'Monitor Tests', 'Create Tests', 'Manage Tests', 'Analyze Business Transactions', and 'Manage Tickets'. The 'Home' tab is selected in the top navigation bar.

Vous trouverez ci-après des détails sur le mappage des options Getting Started (Prise en main) et Quick Links (Liens rapides) vers les options du menu de navigation.

Créer

Permet de réaliser des actions de création communes.

Notez l'icône suivant la section Virtualize using RR pairs (Virtualisation à l'aide de paires demande-réponse). Cette icône indique que cliquer sur le lien vous redirige vers une autre interface utilisateur.

This screenshot compares three parts of the DevTest Portal interface:

- Left Column (Create):** Shows the 'Create' menu item with a dropdown arrow. Underneath are four options: 'API Test' (highlighted with a blue box), 'Virtualize Website (HTTP/S)' (highlighted with a red box), 'Virtualize JDBC', and 'Virtualize using RR pairs' (with a small external link icon).
- Middle Column (Getting Started):** Shows the 'Getting Started' section with a list of 10 items. The second item, 'How to create a test', is highlighted with a blue box.
- Right Column (Quick Links):** Shows the 'Quick Links' section with a list of 8 items. The first item, 'Create a Virtual Service', is highlighted with a red box.

Test API

Ouvre l'onglet API Test (Test API).

Getting Started (Prise en main) : Création d'un test dans la rubrique *Utilisation de CA Application Test*

Quick Links (Liens rapides) : Crédation de tests

Virtualisation de site Web (HTTP/S)

Ouvre l'onglet Virtualize Website (HTTP/S) (Virtualiser un site Web, HTTP/S).

Getting Started (Prise en main) : Procédure d'enregistrement d'un service virtuel dans la rubrique *Utilisation du CA Service Virtualization*

Quick Links (Liens rapides) : Crédation d'un service virtuel

Virtualisation du protocole JDBC

Ouvre l'onglet Virtualize JDBC) (Virtualiser le protocole JDBC)

Rubrique : JDBC dans la rubrique Utilisation du CA Service Virtualization

Virtualisation de paires demande-réponse

Redirection : Console DevTest

Ouvre : l'onglet Create Virtual Service on VSEasy (Créer un service virtuel sur VSEasy)

Rubrique : Crédation et déploiement d'un service virtuel avec VSEasy dans la rubrique *Utilisation du CA Service Virtualization*

Gérer

Gérez des artefacts de test.

The screenshot shows the DevTest interface. On the left, there's a sidebar titled 'Manage' with a dropdown menu set to 'Bank v6'. Below it are several collapsed sections: 'API Tests' (circled in blue), 'Tests', 'Test Suites', and '+ Virtual Services' (circled in red). In the center, there's a 'Getting Started' panel with a list of links: 'How to record and save a virtual service', 'How to edit a virtual service' (highlighted with a red box), 'How to create a test', 'How to monitor a test', 'How to manage test artifacts' (highlighted with a blue box), 'How to manage a ticket', 'How to work with the shelf', and 'How to document a transaction for a manual test'. To the right is a 'Quick Links' panel containing links: 'Create a Virtual Service' (highlighted with a red box), 'Manage Virtual Services' (highlighted with a red box), 'Monitor Tests', 'Create Tests', 'Manage Tests' (highlighted with a blue box), 'Analyze Business Transactions', and 'Manage Tickets'.

Test API

Gérez des tests créés avec le portail DevTest.

Getting Started (Prise en main) : Gestion des artefacts de test, Gestion des tests API dans la rubrique *Utilisation de CA Application Test*

Quick Links (Liens rapides) : Gestion des tests

Test: Current Status (Today) (Statut actuel, aujourd'hui)

Tests

Affichez des informations sur les scénarios de test et exécutez des scénarios de test créés avec la station de travail DevTest Workstation.

Getting Started (Prise en main) : Gestion des artefacts de test, Gestion des tests dans la rubrique *Utilisation de CA Application Test*

Quick Links (Liens rapides) : Gestion des tests

Test: Current Status (Today) (Statut actuel, aujourd'hui)

Suites de tests

Affichez des informations sur les suites de tests et exécutez des suites de tests créées avec la station de travail DevTest Workstation.

Getting Started (Prise en main) : Gestion des artefacts de test, Gestion des suites de tests dans la rubrique *Utilisation de CA Application Test*

Quick Links (Liens rapides) : Gestion des tests

Test: Current Status (Today) (Statut actuel, aujourd'hui)

Services virtuels

Ouvre les onglets : http-rest, WebServicesModel, http

Getting Started (Prise en main) : Procédure de modification d'un service virtuel dans la rubrique *Utilisation du CA Service Virtualization*

Quick Links (Liens rapides) : Gestion des services virtuels

Moniteur

Surveillez des tests ou des services virtuels. Lorsqu'aucun test ou aucune suite ne répond aux critères de sélection, vous pouvez cliquer sur Create a Test (Créer un test) ou Execute a Test (Exécuter un Test).

Monitor	Getting Started	Quick Links
Monitoring Tests	How to record and save a virtual service How to edit a virtual service How to create a test How to monitor a test! How to manage test artifacts How to manage a ticket How to work with the shelf How to document a transaction for a manual test	Create a Virtual Service Manage Virtual Services Monitor Tests Create Tests Manage Tests Analyze Business Transactions Manage Tickets

Surveillance de tests

La fenêtre Monitor Test (Surveiller des tests) permet d'afficher des tests API, des scénarios de test et des suites de tests exécutés dans CA Application Test.

Cliquez sur le bouton de filtre pour basculer vers les options de recherche.

Ouvre l'onglet Monitoring Tests (Surveillance de tests)

Getting Started (Prise en main) : Surveillance d'un test dans la rubrique *Utilisation de CA Application Test*

Quick Links (Liens rapides) : Surveillance des tests

Services virtuels

Redirection : Console DevTest

Ouvre la console de serveur, réseau DevTest, où vous pouvez cliquer sur VSE.

Rubrique : Création et déploiement d'un service virtuel avec VSEasy dans la rubrique *Utilisation du CA Service Virtualization*

Application Insight

Démarrez des tâches communes avec CA Continuous Application Insight.

The screenshot shows the Application Insight interface. On the left is a sidebar with the following buttons:

- Analyze Transactions
- Document Transactions
- Manage Tickets
- Explore Defects

The main area is divided into two sections:

- Getting Started**:
 - How to record and save a virtual service
 - How to edit a virtual service
 - How to create a test
 - How to monitor a test
 - How to manage test artifacts
 - How to manage a ticket
 - How to work with the shelf
 - How to document a transaction for a manual test
- Quick Links**:
 - Create a Virtual Service
 - Manage Virtual Services
 - Monitor Tests
 - Create Tests
 - Manage Tests
 - Analyze Business Transactions
 - Manage Tickets

Analyze Transactions (Analyse de transactions)

Ouvre l'onglet Analyze Transactions (Analyser des transactions)

Getting Started (Prise en main) : Utilisation de l'étagère dans la rubrique *Utilisation de CA Continuous Application Insight*

Quick Links (Liens rapides) : Analyse des transactions organisationnelles

Document Transactions (Documentation de transactions)

Ouvre l'onglet Document Transactions (Documenter des transactions)

Getting Started (Prise en main) : Documentation de transactions à des fins de test manuel dans la rubrique *Utilisation de CA Continuous Application Insight*

Manage Tickets (Gérer des tickets)

Ouvre l'onglet Manage Tickets (Gérer des tickets)

Getting Started (Prise en main) : Gestion d'un ticket dans la rubrique *Utilisation de CA Continuous Application Insight*

Quick Links (Liens rapides) : Gestion des tickets

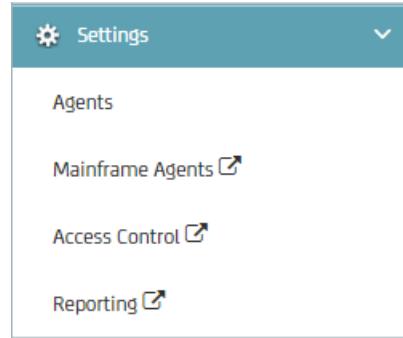
Explore Defects (Explorer les défauts)

Ouvre l'onglet Explore Defects (Explorer les défauts)

Rubrique : Utilisation des défauts dans la rubrique *Utilisation de CA Continuous Application Insight*

Paramètres

L'onglet Settings (Paramètres) n'a pas d'entrées de liens rapides ni de prise en main.



Agents

Ouvre l'onglet Agents.

Rubriques : consultez *Agents*.

Mainframe Agents

Redirections : console DevTest, console de serveur, réseau DevTest

Rubrique : Passerelle Mainframe dans la rubrique *Agents*.

Access Control (Contrôle d'accès)

Redirections : console DevTest, console de serveur Développez l'onglet Administration pour afficher les options de sécurité.

Rubrique : Contrôle d'accès (liste de contrôle d'accès) dans la rubrique *Administration*.

Reporting (Génération de rapports)

Redirections : Console de DevTest, Reporting (Génération de rapports)

Rubriques : Rapports dans la rubrique *Utilisation de CA Application Test*.

Page d'accueil du portail DevTest

La page d'accueil du portail DevTest contient une collection de portlets qui permettent de surveiller l'activité de votre application. Chaque portlet contient des informations sur un composant DevTest que vous pouvez actualiser manuellement.

La page d'accueil contient les portlets suivants :

Prise en main

Affiche des liens vers des rubriques de la documentation qui décrivent des fonctions clés du portail.

Quick Links (Liens rapides)

Affiche des liens vers des opérations de DevTest Solutions.

Aide

Affiche un lien vers la documentation de DevTest, pour poser une question à la communauté de DevTest, ou signaler un problème.

Virtual Services: Current Status (Services virtuels : statut actuel)

Affiche le nombre total de transactions actuelles, le nombre de services virtuels déployés et le nombre de services virtuels enregistrés.

Test: Current Status (Test : statut actuel)

Affiche les résultats de test disponibles. Cliquez sur Show More... (Afficher plus) pour accéder à la fenêtre Monitoring Test (Surveillance de tests).

Agents: Current Status (Agents : statut actuel)

Affiche le statut d'un agent. Sélectionnez un nom d'agent ou cliquez sur Show More... (Afficher plus) pour accéder aux opérations de gestion d'agents dans la fenêtre Agents. Pour afficher les informations de l'agent actuel, actualisez manuellement le portlet.

New Ticket Alerts (Alertes de nouveaux tickets)

Affiche une liste de nouveaux tickets et le nombre tickets nouveaux, identifiés, ou fermés. Le nom, la date et l'heure du ticket s'affiche dans la liste. Sélectionnez New (Nouveau), Identified (Identifié), ou Closed (Fermé) pour accéder à la fenêtre Manage Tickets (Gérer des tickets). La fenêtre Manage Tickets (Gérer des tickets) permet d'afficher des tickets, de mettre à jour des tickets existants, d'afficher une liste de tickets et de rechercher des tickets dans la base de données CA Continuous Application Insight.

Path Alerts (Alertes de chemin)

Affiche une liste de chemins avec des indicateurs d'exception. Sélectionnez un chemin permet d'accéder à la fenêtre Explore Defects (Explorer les défauts).

Points of Interest (Points d'intérêt)

Affiche une liste de tous les points d'intérêt épinglés dans la fenêtre Explore Defects (Explorer les défauts). Sélectionnez le nom dans la liste de points d'intérêt pour accéder à la fenêtre Explore Defects (Explorer les défauts).

Chapitre 2: Didacticiels de CA Application Test

Cette section contient une série de didacticiels qui présentent différents aspects de CA Application Test. Les didacticiels obéissent à un ordre séquentiel. Effectuez-les dans l'ordre présenté.

Les premiers didacticiels vous guident dans l'utilisation de DevTest Workstation pour créer des scénarios de test simples. Ils vous permettent de vous familiariser avec les concepts de base, tels que les projets, les propriétés, les ensembles de données, les filtres et les assertions.

Les didacticiels ultérieurs vous permettent d'acquérir des connaissances plus approfondies sur la procédure de configuration des étapes de test pour interagir et tester plusieurs technologies communes. Ces technologies incluent des objets Java, des pages Web, des objets EJB (Enterprise JavaBeans), des services Web et des bases de données. Vous apprenez également à simuler un test rapide.

Pour mettre en pratique les didacticiels, vous devez disposer de DevTest Workstation et d'un accès à un registre.

Certains didacticiels utilisent le serveur de démonstration comme système testé. Pour plus d'informations sur l'installation du serveur de démonstration, consultez la rubrique *Installation*.

Ce chapitre traite des sujets suivants :

- [Didacticiel 1 : projets, scénarios de test et propriétés](#) (page 20)
- [Didacticiel 2 : ensembles de données](#) (page 28)
- [Didacticiel 3 : filtres et assertions](#) (page 36)
- [Didacticiel 4 : manipulation d'objets Java \(POJO\)](#) (page 45)
- [Didacticiel 5 : exécution d'une application Web de serveur de démonstration](#) (page 57)
- [Didacticiel 6 : test d'un site Web](#) (page 64)
- [Didacticiel 7 : test d'un objet Enterprise JavaBean \(EJB\)](#) (page 79)
- [Didacticiel 8 : test d'un service Web](#) (page 91)
- [Didacticiel 9 : examen et test d'une base de données](#) (page 98)
- [Didacticiel 10 : simulation d'un test rapide](#) (page 109)

Didacticiel 1 : projets, scénarios de test et propriétés

Tâches de didacticiel

Ce didacticiel vous permet de :

- Créer un projet
- Créer un scénario de test
- Ajouter des propriétés
- Ajouter des étapes de test simples
- Utiliser l'utilitaire d'exécution d'un test interactif

Conditions préalables

- Vous devez installer la station de travail DevTest Workstation et entrer les informations de licence de DevTest.
- Consultez également le [glossaire](#) (page 149).

Etape 1 : démarrage de DevTest Workstation

Procédez comme suit:

1. Démarrez le registre.
 - Si le DevTest Server est installé sur votre ordinateur :
 - a. Démarrez le registre en cliquant sur le menu Démarrer, Tous les programmes, DevTest Solutions, EnterpriseDashboard (Tableau de bord d'entreprise). Patientez jusqu'à ce le message Enterprise Dashboard started (Tableau de bord d'entreprise démarré) s'affiche.
 - b. Démarrez le registre en cliquant sur le menu Démarrer, Tous les programmes, DevTest Solutions, Registry (Registre).
 - Si DevTest Workstation est installé sur votre ordinateur, utilisez un registre exécuté sur un autre ordinateur.
2. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, Workstation (Station de travail).
3. Lorsque la boîte de dialogue Set DevTest Registry (Définir le registre DevTest) s'affiche, sélectionnez un registre et cliquez sur OK.
4. La boîte de dialogue Login (Connexion) s'ouvre. Entrez un nom d'utilisateur et un mot de passe valides et cliquez sur Login (Connexion).

Etape 2 : création d'un projet

Le projet que vous créez contient tous les exemples de fichier de scénario de test qui sont requis pour les didacticiels.

Procédez comme suit:

1. A partir du menu principal de DevTest Workstation, sélectionnez File (Fichier), New (Créer), Project (Projet).
La boîte de dialogue Create New Project (Créer un projet) s'affiche.
2. Dans le champ Project Name (Nom du projet), supprimez la valeur par défaut et saisissez **Mes didacticiels**.
3. Cliquez sur Create (Créer).

Le projet Mes didacticiels est créé.

Etape 3 : création d'un scénario de test

Un scénario de test regroupe les spécifications qui permettent de mener à bien un test sur un composant professionnel dans le système testé.

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur le dossier Tests et sélectionnez Create New Test Case (Créer un scénario de test).
2. Définissez le nom de fichier sur **Didacticiel 1**.
3. Cliquez sur Save (Enregistrer).

DevTest Workstation ouvre un nouvel onglet Didacticiel 1. La flèche verte dans l'éditeur de modèles représente le démarrage du scénario de test.

Etape 4 : ajout d'une propriété à la configuration de projet

Dans cette étape, vous définissez une propriété globale dans la configuration de projet. Vous accédez à cette propriété ultérieurement dans le didacticiel.

La configuration par défaut est appelée project.config et est créée automatiquement pour un nouveau projet. Le fichier project.config se trouve dans le dossier Configs (Configurations) du panneau Project (Projet). L'extension de fichier n'est pas affichée. Vous pouvez ajouter des propriétés au fichier project.config et, le cas échéant, créer un fichier de configuration.

Procédez comme suit:

1. Dans le panneau Project (Projet), double-cliquez sur **Project** sous le dossier Mes didacticiels > Configs (Configurations).
L'éditeur de propriétés s'ouvre.
2. Cliquez sur  Add (Ajouter) au bas de l'éditeur de propriétés pour ajouter une ligne.
3. Dans le champ Key (Clé), saisissez **config_prop**.
4. Dans le champ Value (Valeur), saisissez **42**.



Save

Save (Enregistrer)

5. A partir de la barre d'outils principale, cliquez sur  Save (Enregistrer)

Etape 5 : ajout d'une étape de test

Un scénario de test inclut une ou plusieurs étapes de test. Dans cette procédure, vous ajoutez une étape de test Output Log Message (Message de journal de sortie) pour enregistrer du texte dans le fichier journal.

Procédez comme suit:

1. Cliquez sur l'onglet Didacticiel 1.



2. Cliquez sur Add Step (Ajouter une étape), sélectionnez Utilities (Utilitaires) et sélectionnez Output Log Message (Message de journal de sortie).

Une étape nommée Output Log Message est ajoutée à l'éditeur de modèles.

3. Cliquez avec le bouton droit de la souris sur l'étape Output Log Message et sélectionnez Rename (Renommer).
4. Remplacez le nom par Ma publication de message de journal.
5. Vérifiez que Ma publication de message de journal est toujours sélectionnée. Dans le volet droit, cliquez sur la flèche à côté de Output Log Message (Message de journal de sortie).

La barre d'état Ouput Log Message s'ouvre.

Etape 6 : ajout d'un message de journal

Dans l'éditeur de journal ouvert, ajoutez un message de journal comprenant plusieurs propriétés.

Les propriétés incluses dans le message de journal proviennent de plusieurs sources :

- La propriété LISA_HOME est définie automatiquement.
- La propriété java.version est une propriété système.
- Vous avez ajouté la propriété config_prop à la configuration du projet à l'étape 4.
- Vous créez une propriété avec le nom **MyOutputLogMessage_step_prop** dans ce message de journal.

La syntaxe d'une propriété est **{{property_name}}**.

Procédez comme suit:

1. Dans l'éditeur de journal, supprimez le texte d'espace réservé.
2. Copiez et collez le texte suivant dans l'éditeur de journal :

Le répertoire de base LISA est : {{LISA_HOME}}. LISA définit cette propriété.

Valeur de config_prop : {{config_prop}}. Cette propriété est définie dans la configuration.

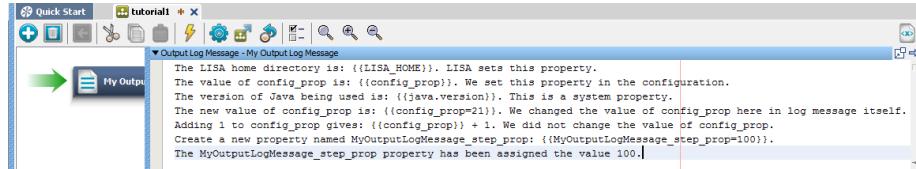
La version de Java utilisée est : {{java.version}}. Il s'agit d'une propriété système.

Nouvelle valeur de config_prop : {{config_prop=21}}. Vous avez modifié la valeur de config_prop dans le message de journal auparavant.

L'ajout de 1 à config_prop donne le résultat suivant : {{config_prop}} + 1. La valeur de config_prop n'a pas été modifiée. Créez une nouvelle propriété nommée MyOutputLogMessage_step_prop : {{MyOutputLogMessage_step_prop=100}}.

La valeur 100 a été affectée à la propriété MyOutputLogMessage_step_prop.

L'éditeur de journal est similaire au graphique suivant.



Etape 7 : ajout d'un second message de journal

La deuxième étape de test dans le scénario de test enregistre un message différent dans le fichier journal.

Procédez comme suit:

1. Fermez l'étape du premier message de journal en cliquant sur la flèche dans le coin supérieur gauche de la fenêtre.



2. Cliquez sur  Add Step (Ajouter une étape), sélectionnez Utilities (Utilitaires) et sélectionnez Output Log Message (Message de journal de sortie).

Une étape nommée Output Log Message est ajoutée à l'éditeur de modèles et la barre d'état Output Log Message s'ouvre.

3. Dans l'éditeur de journaux, supprimez le texte de l'espace réservé.
4. Copiez et collez le texte suivant dans l'éditeur de journaux :
Valeur actuelle de config_prop : {{config_prop}}.
Valeur actuelle de MyOutputLogMessage_step_prop : {{MyOutputLogMessage_step_prop}}.

Remarque : Le message de journal ne modifie pas les valeurs de **config_prop** ou de **MyOutputLogMessage_step_prop**.

5. Fermez l'étape du second message de journal en cliquant sur la flèche dans le coin supérieur gauche de la fenêtre.
6. A partir de la barre d'outils de l'application principale, cliquez sur Save (Enregistrer)



 ou sélectionnez File (Fichier), Save (Enregistrer), Didacticiel 1.

Etape 8 : exécution de l'étape Ma publication de message de journal

L'utilitaire ITR (Interactive Test Run, exécution d'un test interactif) vous guide dans un scénario de test et vous permet de le vérifier.

Procédez comme suit:



1. A partir de la barre d'outils, cliquez sur Start Interactive Test Run (Démarrer l'exécution d'un test interactif).

L'ITR s'ouvre. L'ITR contient un volet Execution History (Historique d'exécution) sur la gauche et un ensemble d'onglets sur la droite.

2. Dans le volet Execution History, cliquez sur Execute Next Step (Exécuter l'étape suivante).

L'étape Ma publication de message de journal est exécutée. L'onglet Response (Réponse) affiche la réponse à l'étape Ma publication de message de journal. Les valeurs réelles remplacent les propriétés.

Etape 9 : observation des valeurs de propriété

L'ITR vous permet également d'observer la création et la modification des propriétés.

Procédez comme suit:

1. Dans l'ITR, cliquez sur l'onglet Properties (Propriétés).

L'onglet Properties affiche la valeur de chaque propriété avant et après l'exécution de l'étape Ma publication de message de journal. Les valeurs que l'étape a créées sont surlignées en vert. Les valeurs modifiées par l'étape sont surlignées en jaune. Notez que la valeur de config_prop a été modifiée de 42 à 21.

2. Comparez ces valeurs avec la réponse de l'étape 8.

Etape 10 : exécution de l'étape Output Log Message (Message de journal de sortie)

Dans cette procédure, vous utilisez l'ITR pour exécuter la deuxième étape du scénario de test.

Procédez comme suit:

1. Dans l'ITR, cliquez sur  Execute Next Step (Exécuter l'étape suivante) pour exécuter l'étape Output Log Message (Message de journal de sortie).
2. Pour afficher la réponse, cliquez sur l'onglet Response (Réponse). Même si vous définissez config_prop sur 42 dans le fichier project.config, vous avez remplacé la valeur par 21 dans l'étape Ma publication de message de journal, et la valeur n'a pas été modifiée dans l'étape Output Log Message. La valeur de la propriété **MyOutputLogMessage_step_prop** a également été reportée de l'étape Ma publication de message de journal à l'étape Output Log Message.
3. Pour afficher les valeurs de propriété actuelles et précédentes, cliquez sur l'onglet Properties (Propriétés).
4. Lorsque vous avez terminé, fermez les onglets du didacticiel 1 et du projet.

Résumé

Dans ce didacticiel, vous avez abordé les propriétés. Vous avez vu que les propriétés obéissent à une syntaxe spéciale, {{property_name}}. Vous pouvez définir des propriétés à l'aide d'une variation de cette syntaxe : {{property_name=value}}. Après avoir défini une propriété, utilisez ou modifiez-la dans les étapes ultérieures d'un scénario de test.

Ce didacticiel vous a permis de :

- Découvrir la procédure de création et d'enregistrement d'un scénario de test
- Découvrir la procédure d'ajout d'une étape de test simple (Output Log Message (Message de journal de sortie))
- Utiliser une configuration pour stocker des propriétés
- Découvrir l'utilitaire ITR

Didacticiel 2 : ensembles de données

Ce didacticiel présente la procédure de création et d'utilisation d'un ensemble de données simple. Vous apprendrez également à fournir les données d'un ensemble de données à un scénario de test.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Créer un ensemble de données simple
- Utiliser l'ensemble de données de différentes façons
- Itérer une série d'étapes de test à l'aide d'un ensemble de données

Conditions préalables

- Vous avez effectué le didacticiel 1 sur les propriétés.
- DevTest Workstation est ouvert.

Etape 1 - création d'un ensemble de données

Dans ce didacticiel, vous utilisez un fichier texte de valeurs séparées par des virgules comme ensemble de données. Cette option est l'un des choix possibles pour créer un ensemble de données. Après avoir créé le fichier texte, vous l'importez dans le projet Mes didacticiels.

Procédez comme suit:

1. Dans un éditeur de texte comme Bloc-notes, créez un fichier texte.
2. Copiez et collez les propriétés et les valeurs suivantes dans l'éditeur de texte.
N'utilisez pas d'espaces dans le fichier texte.
month,day,year
3,2,1956
4,7,2007
1,3,2010
5,8,{{yearglobal}}
8,10,2004
12,11,{{yearglobal}}
10,12,2007
3,5,2011
- La première ligne spécifie les noms des propriétés auxquelles ces données sont affectées (month, day, year). Les lignes restantes spécifient les données qui sont lues et utilisées dans le scénario de test. Deux des lignes incluent une propriété nommée **yearglobal**.
3. Enregistrez le fichier en tant que dates.txt.
4. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur le dossier Data (Données) dans le projet Mes didacticiels et sélectionnez Import Files (Importer des fichiers).
5. Accédez au dossier dans lequel vous avez enregistré le fichier dates.txt et sélectionnez le nom de fichier.
6. Cliquez sur Open (Ouvrir). Le fichier dates.txt s'affiche dans le dossier Data.

Etape 2 : création d'un scénario de test

Vous ajoutez un scénario de test au projet Mes didacticiels.

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur le dossier Tests et sélectionnez Create New Test Case (Créer un scénario de test).
2. Définissez le nom du fichier sur **Didacticiel 2**.
3. Cliquez sur Save (Enregistrer).

Etape 3 : ajout d'une propriété à la configuration de projet

Le fichier dates.txt inclut une propriété nommée yearglobal. Dans cette procédure, vous ajoutez la propriété yearglobal à la configuration de projet.

Procédez comme suit:

1. Dans le panneau Project (Projet), double-cliquez sur project.config.
2. Cliquez sur  Add (Ajouter) pour ajouter une ligne.
3. Dans le champ Key (Clé), saisissez yearglobal.
4. Dans le champ Value (Valeur), saisissez 1999.



5. Cliquez sur Save (Enregistrer) .

Etape 4 : ajout d'une étape de test pour l'étape Output Log Message (Message de journal de sortie)

Pour enregistrer un texte dans le journal, utilisez l'étape de test Output Log Message (Message de journal de sortie).

Procédez comme suit:

1. Cliquez sur l'onglet Didacticiel 2.



2. Cliquez sur Add (Ajouter).

Le menu Add Step s'affiche.

3. Sélectionnez Utilities (Utilitaires), puis Output Log Message (Message de journal de sortie).

Une étape nommée **Output Log Message** (Message de journal de sortie) est ajoutée à l'éditeur de modèles.

4. Cliquez avec le bouton droit de la souris sur Output Log Message et sélectionnez Rename (Renommer). Remplacez le nom d'hôte par DSstep1.

5. Dans le volet droit, cliquez sur la flèche près de Output Log Message (Message de journal de sortie).

La barre d'état Ouput Log Message s'ouvre.

6. Supprimez le texte d'espace réservé.

7. Saisissez le message de journal suivant :

Date : {{month}}/{{day}}/{{year}}

Remarque : Les accolades sont importantes. Le scénario de test est exécuté uniquement si vous les incluez.

8. Pour fermer la barre d'état Output Log Message (Message de journal de sortie), cliquez n'importe où dans l'éditeur de modèles.



9. Cliquez sur Save (Enregistrer).

Etape 5 : création d'une autre étape Output Log Message (Message de journal de sortie)

Créez une autre étape de test similaire à l'étape de test DSstep1.

Procédez comme suit:



1. Cliquez sur  Add (Ajouter).

Le menu Add Step (Ajouter une étape) s'affiche.

2. Sélectionnez Utilities (Utilitaires), puis Output Log Message (Message de journal de sortie).

Une étape nommée Output Log Message est ajoutée à l'éditeur de modèles.

3. Cliquez avec le bouton droit de la souris sur Output Log Message et sélectionnez Rename (Renommer). Remplacez le nom par DSstep2.

4. Dans le volet droit, cliquez sur la flèche à côté de Output Log Message (Message de journal de sortie).

La barre d'état Ouput Log Message s'ouvre.

5. Supprimez le texte d'espace réservé.

6. Entrez le message de journal suivant :

Date : {{month}}/{{day}}/{{year}}

7. Pour fermer la barre d'état Output Log Message (Message de journal de sortie), cliquez n'importe où dans l'éditeur de modèles.



8. Cliquez sur  Save (Enregistrer).

Etape 6 : exécution du test

Pour exécuter le test et voir ce qui se produit, utilisez l'ITR (Interactive Test Run, exécution d'un test interactif).

Procédez comme suit:



Start Interactive Test Run (Démarrer l'exécution d'un test interactif).

L'ITR s'ouvre.



1. A partir de la barre d'outils, cliquez sur Start Interactive Test Run (Démarrer l'exécution d'un test interactif).
2. Dans le volet Execution History (Historique d'exécution), cliquez sur Automatically execute test (Exécuter automatiquement le test).
3. Une fois le test terminé, cliquez sur OK.
4. Dans le volet Execution History, cliquez sur DSstep1 et DSstep2.

Remarquez que les propriétés month, day et year n'ont pas été remplacées par des valeurs. Ce résultat est attendu, car vous n'avez pas ajouté l'ensemble de données au scénario de test.

Etape 7 : ajout de l'ensemble de données

Ajoutez l'ensemble de données dates.txt à l'étape de test DSstep1.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez DSstep1.
2. Dans le volet droit, double-cliquez sur l'onglet d'étape Data Sets (Ensembles de données).
3. Cliquez sur  Add (Ajouter) sous l'élément Data Sets.
4. Dans la liste Common DataSets (Ensembles de données communs), sélectionnez l'option Read rows from a Delimited Data File (Lire les lignes à partir d'un fichier de données délimité).

L'ensembles de données est ajouté à l'étape de test.

L'éditeur d'ensembles de données s'ouvre dans le volet droit.

5. Dans l'éditeur d'ensembles de données, définissez le nom sur DatesDS.
6. Cliquez sur  File Location (Emplacement de fichier), puis recherchez le fichier dates.txt dans le répertoire LISA_HOME\Projects\My Tutorials\Data et sélectionnez-le.

7. Cliquez sur Test and Keep (Tester et conserver).

Si le test réussit, la fenêtre Data Set Editor (Editeur d'ensembles de données) renvoie un message Test successful (Test réussi).

8. Cliquez sur OK.

9. A partir de la barre d'outils, cliquez sur  Start Interactive Test Run (Démarrer l'exécution d'un test interactif), puis sélectionnez Start new ITR (Démarrer une nouvelle ITR).

10. Dans le volet Execution History (Historique d'exécution), cliquez sur  Automatically execute test (Exécuter automatiquement le test).

11. Une fois le test terminé, cliquez sur OK.

12. Dans le volet Execution History (Historique d'exécution), cliquez sur DSstep1 et DSstep2.

La première ligne de données dans l'ensemble de données est affichée dans l'onglet Response (Réponse). Les deux réponses d'étape affichent la même date, car seul l'ensemble de données de DSstep1 est lu.

Etape 8 : modification du comportement de l'ensemble de données

Modifiez maintenant l'ensemble de données de sorte à l'exécuter en boucle dans l'étape de test jusqu'à la lecture complète des lignes de l'ensemble de données.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez l'étape de test DSstep1.
2. Dans le panneau des éléments de l'étape DSstep1, cliquez sur la flèche à côté de DatesDS sous l'élément Data Sets (Ensembles de données).
L'éditeur d'ensembles de données s'ouvre.
3. Dans le champ At end of data (A la fin des données), sélectionnez l'option Execute (Exécuter).
4. Cliquez sur la flèche déroulante dans le champ Execute (Exécuter) et sélectionnez End the Test (Terminer le test) dans la liste de choix qui s'affiche.
Ce paramètre permet de mettre fin au test lorsque toutes les lignes de données ont été lues.
5. Cliquez sur Test and Keep (Tester et conserver).
6. Cliquez sur OK pour fermer le message de confirmation du test.
7. Dans l'éditeur de modèles, sélectionnez l'étape de test DSstep2.
8. Dans l'onglet Step Information (Informations sur l'étape), définissez la liste déroulante Next (Suivante) sur DSstep1.

Cette configuration force les deux étapes de test à effectuer une boucle. Les flèches dans l'éditeur de modèles indiquent l'ordre d'exécution : DSstep1, DSstep2, puis DatesDS.

9. A partir de la barre d'outils, cliquez sur  Start Interactive Test Run (Démarrer l'exécution d'un test interactif), puis sélectionnez Start new ITR (Démarrer une nouvelle ITR).

10. Dans l'ITR, cliquez sur Automatically execute test (Exécuter automatiquement le



- test).

Le scénario de test s'exécute en boucle jusqu'à ce que toutes les lignes de données de l'ensemble de données aient été lues.

11. Une fois le test terminé, cliquez sur OK.



12. Cliquez sur  Save (Enregistrer).

Didacticiel 2 : résumé

Ce didacticiel vous a permis de :

- Créer un ensemble de données séparées par des virgules
- Utiliser l'ensemble de données pour exécuter un scénario de test simple
- Découvrir la méthode qu'utilise une étape de test pour accéder aux données dans l'ensemble de données

Didacticiel 3 : filtres et assertions

Dans ce didacticiel, vous modifiez le scénario de test qui a été créé dans le didacticiel 2 de sorte à inclure un filtre et une assertion.

Pour une introduction sur les filtres et les assertions, consultez les rubriques Filtres et Assertions dans la rubrique *Utilisation de CA Application Test*.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Enregistrer un scénario de test existant sous un nouveau nom
- Ajouter une assertion à une étape de test
- Ajouter un filtre à une étape de test

Conditions préalables

- Vous avez effectué le didacticiel 2 sur les ensembles de données.
- DevTest Workstation est ouvert.

Etape 1 : création d'un scénario de test à partir d'un scénario de test existant

Dans cette étape, vous ouvrez le scénario tutorial2.tst et l'enregistrez en tant que tutorial3.tst.

Procédez comme suit:

1. Ouvrez le scénario de test tutorial2.tst dans le projet Mes didacticiels.
2. Dans la barre de menus, sélectionnez File (Fichier), Save As (Enregistrer sous).
3. Dans le champ File Name (Nom de fichier), entrez **Didacticiel 3**.
4. Cliquez sur Save (Enregistrer).

Le scénario de test Didacticiel 3 est créé et enregistré sous le projet Mes didacticiels.

Etape 2 : modification d'une action de l'étape de test

Modifiez l'action Next Steps (Etapes suivante) pour les deux étapes de test de façon à ce que DSstep1 soit l'étape suivante. Seule la première étape de l'ensemble de données est lue.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez DSstep1.
2. Dans l'onglet Step Information (Informations sur l'étape), définissez l'option Next Step (Etape suivante) sur **DSstep1**.
Avec cette action, la sortie revient à la même étape DSStep1. Pour le moment, les icônes d'alerte s'affichent à côté de l'étape DSStep2.
3. Dans l'éditeur de modèles, double-cliquez sur DSstep2 et changez l'étape Output Log Message (Message de journal de sortie) comme suit :
La date contient 1999. Date : {{month}}/{{day}}/{{year}}

Remarque : Les accolades sont importantes. Le scénario de test est exécuté uniquement si vous les incluez.



4. Cliquez sur **Save** Save (Enregistrer).

Etape 3 : ajout d'une assertion

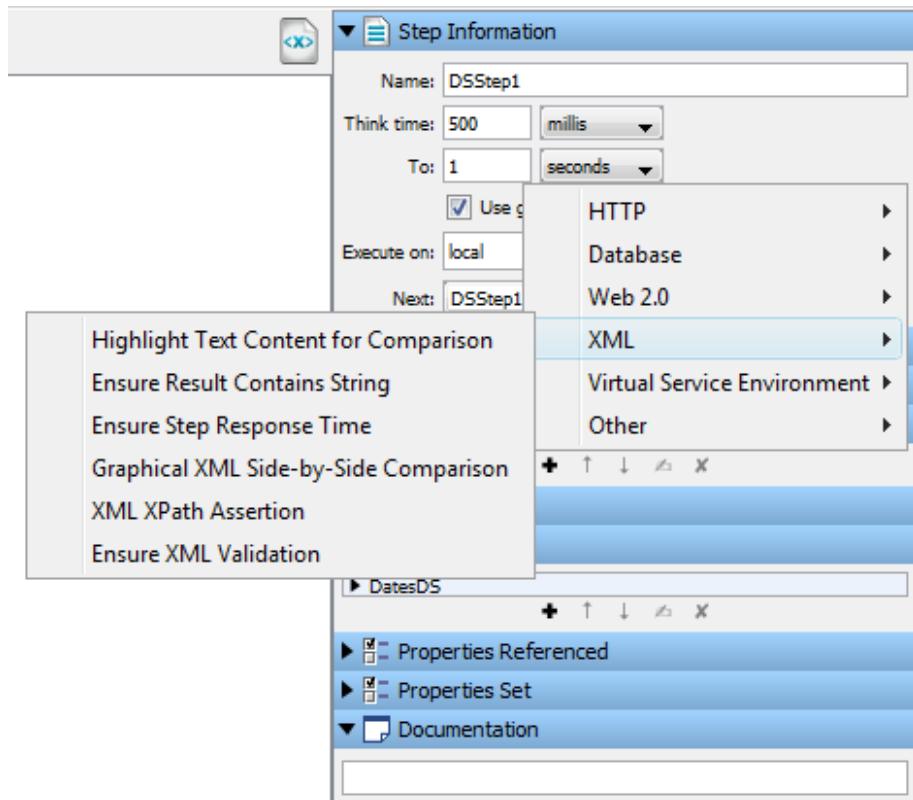
Vous pouvez ajouter différents types d'assertions à un scénario de test. Dans cette procédure, ajoutez une assertion XML nommée **Ensure Result Contains String** (Vérifier que le résultat contient une chaîne).

La logique d'assertion est la suivante :

- Si la réponse contient la chaîne 1999, l'étape DSstep2 est exécutée en suivant.
- Si la réponse ne contient pas la chaîne 1999, l'étape DSstep1 est exécutée en suivant.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez DSstep1.
2. Ouvrez l'onglet Assertions.
3. Cliquez sur  Add (Ajouter).
4. A partir du sous-menu XML, sélectionnez Ensure Result Contains String (Vérifier que le résultat contient une chaîne).

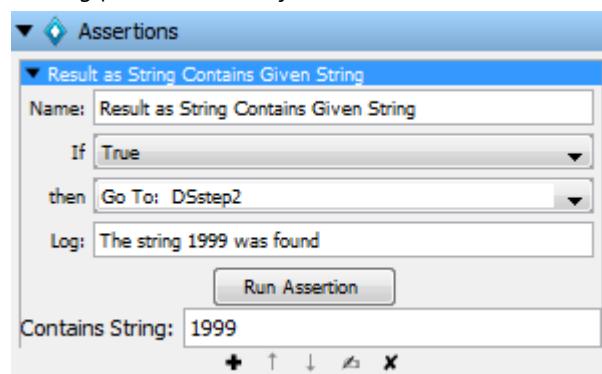


5. La nouvelle assertion appliquée à DSstep1 est ajoutée à l'onglet Assertions.

L'éditeur d'assertions s'ouvre.

6. Dans l'éditeur d'assertions, procédez comme suit :
 - a. Dans la liste If (Si), sélectionnez True.
 - b. Dans la liste then (alors), sélectionnez Go To (Aller à) : DSstep2.
 - c. Dans le champ Log (Journal), entrez The string 1999 was found (La chaîne 1999 a été trouvée).
 - d. Dans le champ Contains String (Contient la chaîne) entrez 1999.

Figure 1: Capture d'écran de la boîte de dialogue d'assertion Result as String Contains Given String (Le résultat sous forme de chaîne contient la chaîne appropriée.)



7. Cliquez sur  Save (Enregistrer).

Etape 4 : test de l'assertion

Pour déterminer si l'assertion fonctionne comme prévu, utilisez l'utilitaire Interactive Test Run (Exécution d'un test interactif).

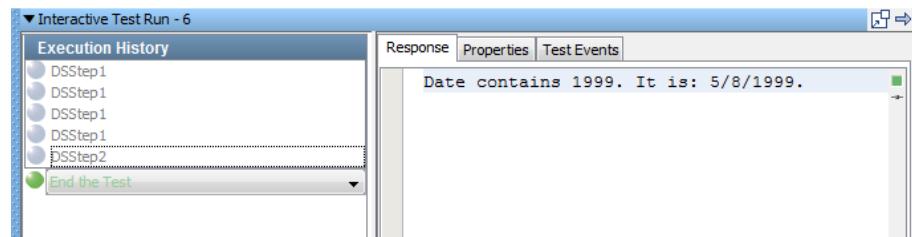
Procédez comme suit:

1. Démarrez une nouvelle session d'ITR.



2. Dans le volet Execution History (Historique d'exécution), cliquez sur Automatically execute test (Exécuter automatiquement le test).
3. Une fois le test terminé, cliquez sur OK.
4. Vérifiez l'onglet Response (Réponse).

Remarque : L'étape DSstep2 est ensuite exécutée, lorsque DSstep1 détecte une date contenant l'année 1999.



5. Cliquez sur l'onglet Properties (Propriétés) et vérifiez le comportement des propriétés.

Response	Properties	Test Events
Initial property values may be changed prior to executing the step. They are read-only after execution. Create a new property by editing an existing key.		
Key	Value	Previous Value
LISA_DOC_PATH	C:\Lisa\Projects\My Tut...	C:\Lisa\Projects\My Tut...
LISA_LAST_STEP	end	DSStep2
lisa.DSStep2.rsp	Date contains 1999. It is...	Date contains 1999. It i...
lisa.end.rsp	The test has ended	
year	1999	1999
day	8	8
robot	0	0
lisa.DSStep2.rsp.time	0	0
LISA_TC_PATH	C:\Lisa\Projects\My Tut...	C:\Lisa\Projects\My Tut...
LISA_HOST	Diana-PC	Diana-PC
LISA_PROJ_NAME	My Tutorials	My Tutorials
lisa.end.rsp.time	0	
LISA_USER	arhoades@Diana-PC	arhoades@Diana-PC
instance	0	0
testCaseId	66366438313934302D6...	66366438313934302D6...
LISA_TC_URL	file:/C:/Lisa/Projects/My...	file:/C:/Lisa/Projects/My...
lisa.DatesDS.returnedPr...	[DatesDS_RowNum, mo...	[DatesDS_RowNum, mo...
LASTRESPONSE	The test has ended	Date contains 1999. It i...
config_prop	42	42
testCase	tutorial3	tutorial3
lisa.DSStep1.rsp.time	0	0
yearglobal	1999	1999
lisa.DSStep1.rsp	Date is: 5/8/1999	Date is: 5/8/1999
LISA_PROJ_ROOT	C:/Lisa/Projects/My Tut...	C:/Lisa/Projects/My Tut...
month	5	5
LISA_DOC_URL	file:/C:/Lisa/Projects/My...	file:/C:/Lisa/Projects/My...
DatesDS_RowNum	4	4

6. Cliquez sur l'onglet Test Events (Événements de test) et vérifiez les événements générés.

▼ Interactive Test Run - 6		Execution History				Response Properties Test Events			
		Execution History		Response Properties		Test Events			
		DSStep1		EventID	Timestamp	Short	Long		
		DSStep1		Cycle ended ...	Mon Jun 27 14:08:1...	66366438313934302D62...	N/A		
		DSStep1		Cycle ending	Mon Jun 27 14:08:1...	66366438313934302D62...	Signaled...		
		DSStep1		Cycle history	Mon Jun 27 14:08:1...	66366438313934302D62...			
		DSStep2							
		End the Test							

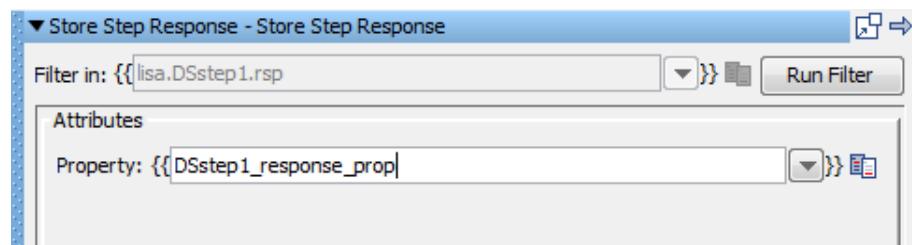
Etape 5 : ajout d'un filtre

Vous pouvez ajouter différents types de filtre à un scénario de test. Dans cette procédure, vous ajoutez un filtre d'utilitaire nommé Store Step Response (Stocker la réponse d'étape). Ce type de filtre vous permet d'enregistrer la réponse d'étape en tant que propriété.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez DSstep1.
2. Ouvrez l'onglet Filters (Filtres).
3. Cliquez sur  Add (Ajouter).
4. Dans le sous-menu Utility Filters (Filtres d'utilitaire), sélectionnez Store Step Response (Stocker la réponse d'étape).
L'éditeur de filtres s'ouvre.
5. Dans l'éditeur de filtres, définissez le nom de la propriété sur **DSstep1_response_prop**.

La réponse d'étape est stockée dans cette propriété.



6. Dans l'éditeur de modèles, double-cliquez sur DSstep2 et ajoutez le texte suivant à la fin du texte Output Log Message (Message de journal de sortie) :
Valeur de DSstep1_response_prop : {{DSstep1_response_prop}}.



7. Cliquez sur  Save (Enregistrer).

Etape 6 : test du filtre

Vous pouvez utiliser l'utilitaire ITR (Exécution d'un test interactif) pour déterminer si le filtre fonctionne comme prévu.

Procédez comme suit:

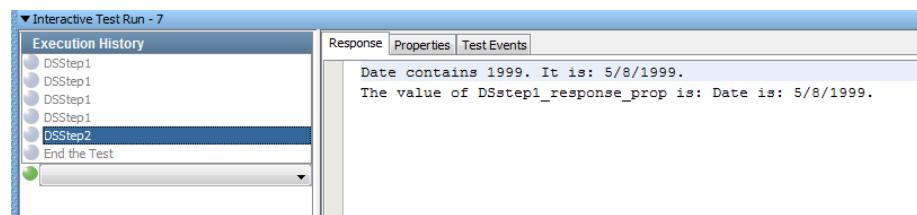
1. Démarrez une nouvelle session d'ITR.



2. Dans le volet Execution History (Historique d'exécution), cliquez sur Automatically execute test (Exécuter automatiquement le test).
3. Une fois le test terminé, cliquez sur OK.
4. Vérifiez l'onglet Response (Réponse).

L'étape de test DSStep2 affiche le texte supplémentaire que vous avez ajouté au message de journal de sortie.

Figure 2: Capture d'écran de l'onglet Response (Réponse) de l'ITR pour le didacticiel 3



5. Cliquez sur l'onglet Properties (Propriétés) et notez l'emplacement dans lequel la propriété DSstep1_response_prop est créée et modifiée.

Key	Value	Previous Value
LISA_DOC_PATH	C:\Lisa\Projects\My Tutorials\Tests	C:\Lisa\Projects\My Tutorials\Tests
LISA_LAST_STEP	DSStep2	DSStep1
lisa.DSStep2.rsp	Date contains 1999, It is: 5/8/1999,...	
year	1999	1999
day	8	8
robot	0	0
lisa.DSStep2.rsp.time	0	
LISA_TC_PATH	C:\Lisa\Projects\My Tutorials\Tests	C:\Lisa\Projects\My Tutorials\Tests
LISA_HOST	Diana-PC	Diana-PC
LISA_PROJ_NAME	My Tutorials	My Tutorials
LISA_USER	arhoades@Diana-PC	arhoades@Diana-PC
DSstep1_response_prop	Date is: 5/8/1999	Date is: 5/8/1999
instance	0	0
testCaseId	30636437383534612D646530382D3...	30636437383534612D646530382D3...
LISA_TC_URL	file:/C:/Lisa/Projects/My%20Tutorial...	file:/C:/Lisa/Projects/My%20Tutorial...
lisa.DatesDS.returnedProps	[DatesDS_RowNum, month, lisa.Dat...]	[DatesDS_RowNum, month, lisa.Dat...]
LASTRESPONSE	Date contains 1999, It is: 5/8/1999,...	Date is: 5/8/1999
config_prop	42	42
testCase	tutorial3	tutorial3
lisa.DSStep1.rsp.time	0	0
yeardglobal	1999	1999
lisa.DSStep1.rsp	Date is: 5/8/1999	Date is: 5/8/1999
LISA_PROJ_ROOT	C:/Lisa/Projects/My Tutorials	C:/Lisa/Projects/My Tutorials
month	5	5
LISA_DOC_URL	file:/C:/Lisa/Projects/My%20Tutorial...	file:/C:/Lisa/Projects/My%20Tutorial...
DatesDS_RowNum	4	4

6. Cliquez sur l'onglet Test Events (Événements de test) et vérifiez les événements générés.

EventID	Timestamp	Short	Long
Info message	Mon Jun 27 14:14:34 CDT 2011	DSStep2	Date contains 19...
Log message	Mon Jun 27 14:14:34 CDT 2011	Will execute the default next step	

Didacticiel 3 : résumé

Ce didacticiel vous permet de :

- Découvrir les filtres et les assertions
- Ouvrir et modifier un scénario de test existant
- Découvrir la procédure d'ajout d'une assertion simple
- Découvrir la procédure d'ajout d'un filtre simple
- Utiliser l'utilitaire ITR (Exécution d'un test interactif) pour déterminer si l'assertion et le filtre fonctionnent comme prévu.

Informations complémentaires

DevTest fournit des filtres et des assertions pour répondre à la plupart des situations que vous rencontrez lors du développement d'un scénario de test. Si aucun filtre approprié n'existe, DevTest fournit un mécanisme permettant de développer des filtres et des assertions personnalisés via le kit SDK (Kit de développement logiciel). Pour plus d'informations, reportez-vous à la section *Utilisation du kit SDK*.

Didacticiel 4 : manipulation d'objets Java (POJO)

Dans ce didacticiel, vous créez et manipulez un objet Java simple et utilisez la classe **java.util.Date** pour créer un objet de date.

Créez d'abord l'objet et déterminez la procédure d'appel des méthodes sur celui-ci. Vous pouvez ensuite inclure l'objet dans un simple éditeur de modèles DevTest.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Utiliser l'étape de test Dynamic Java Execution (Exécution Java dynamique)
- Utiliser l'éditeur Complex Object Editor (Editeur d'objets complexes) pour des objets simples
- Utiliser des filtres intégrés et enregistrer les résultats dans une propriété

Conditions préalables

- Vous avez effectué le didacticiel 3 sur les filtres et les assertions.
- DevTest Workstation est ouvert.

Etape 1 : création d'un scénario de test

Pour créer un scénario de test :

- Dans le projet Mes didacticiels, créez un scénario de test nommé Didacticiel 4.

Etape 2 : création d'une étape de test Dynamic Java Execution (Exécution Java dynamique)

L'étape de test Dynamic Java Execution (Exécution Java dynamique) vous permet de créer un objet Java à partir d'une classe dans la variable classpath de DevTest. Dans la procédure suivante, vous utilisez la classe **java.util.Date**.

Procédez comme suit:

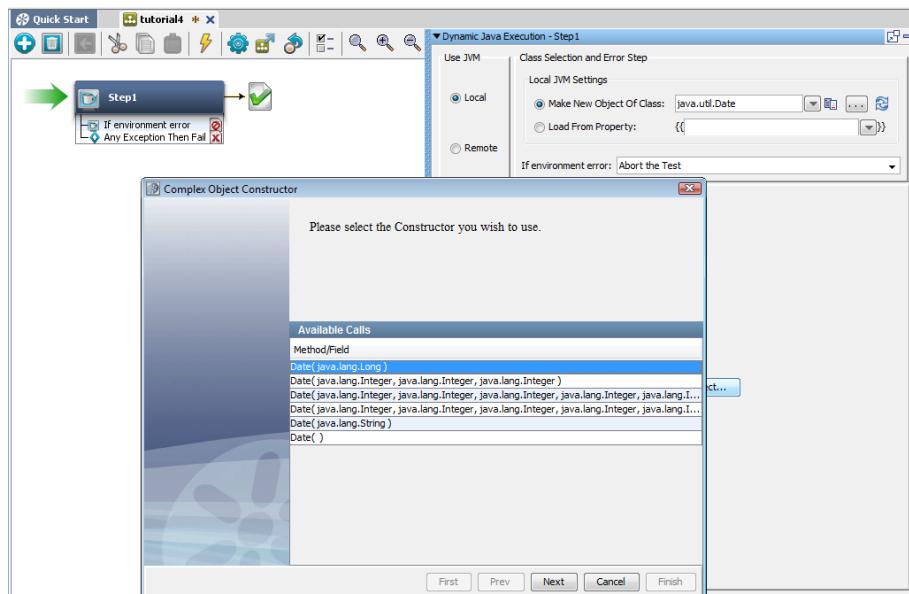


- Cliquez sur **Add Step (Ajouter une étape)**.

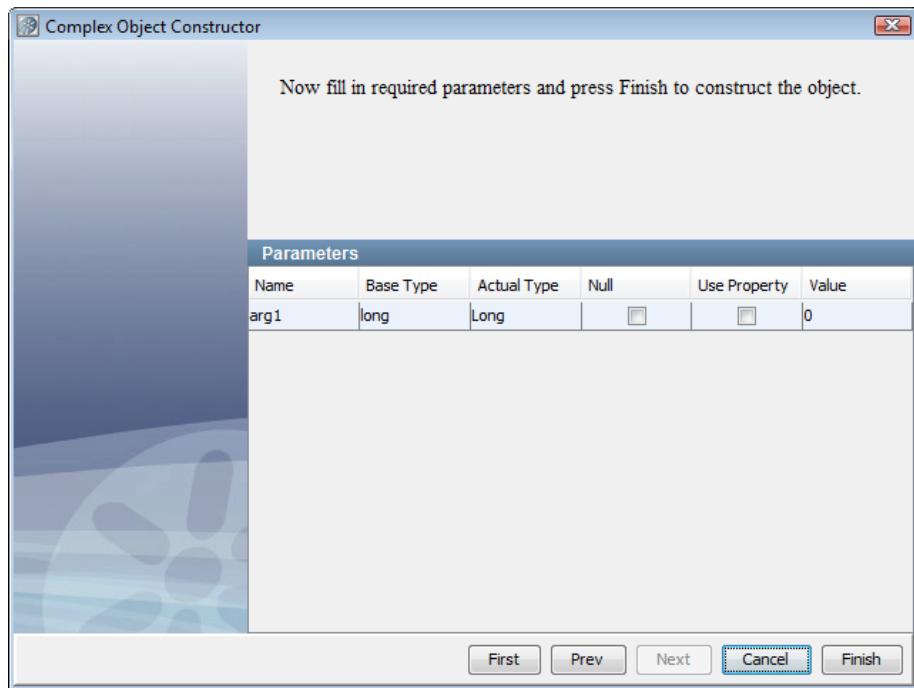
Le menu Add Step s'affiche.

- Sélectionnez Java/J2EE, puis Dynamic Java Execution (Exécution Java dynamique).

L'éditeur Dynamic Java Execution s'ouvre.

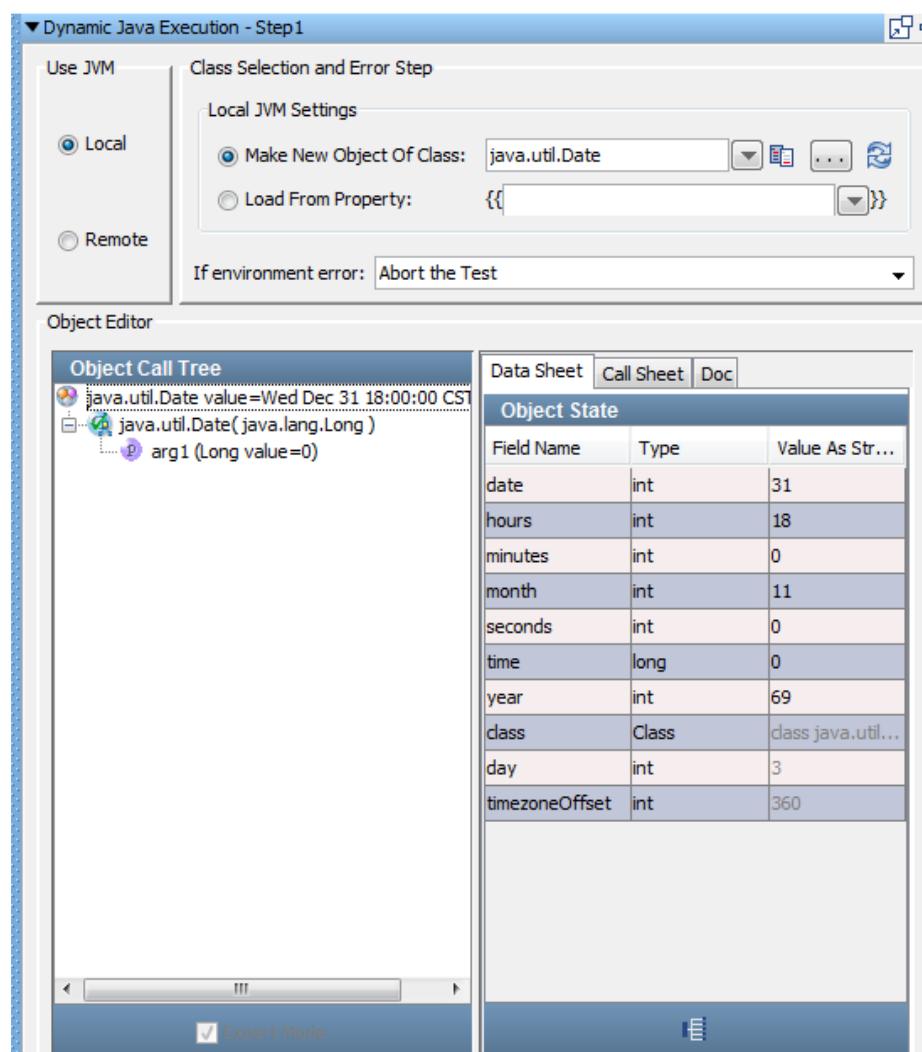


- Dans la zone Local JVM Settings (Paramètres de la machine virtuelle Java locale), vérifiez que l'option Make New Object of Class (Créer un objet de classe) est sélectionnée.
- Dans le champ à droite de Make New Object of Class, saisissez **java.util.Date**.
- Cliquez sur Construct/Load Object (Générer/charger un objet).
L'assistant Complex Object Constructor (Constructeur d'objets complexes) s'affiche. La première étape affiche les constructeurs disponibles.
- Sélectionnez le constructeur **Date(java.lang.Long)**.



7. Cliquez sur Next (Suivant).
8. Cliquez sur Finish (Terminer).

L'éditeur Complex Object Editor (Editeur d'objets complexes) s'ouvre.



Vous disposez désormais d'un objet Java à manipuler dans l'éditeur Complex Object Editor.

Etape 3 : Appel sur l'objet Java

L'éditeur Complex Object Editor (Editeur d'objets complexes) est divisé en deux panneaux. Le panneau de gauche contient l'arborescence Object Call Tree (Arborescence des appels d'objet), qui suit les appels de méthode, ainsi que leurs paramètres d'entrée et les valeurs renvoyées. Les icônes suivantes sont utilisées pour identifier les branches dans l'arborescence Object Call Tree :



Type (classe) de l'objet actuellement chargé, suivi de la réponse à l'appel de la méthode `toString()` de l'objet



Constructeur appelé. Cette icône est affichée si plusieurs constructeurs existent.



Appel de méthode qui n'a pas été exécuté.



Appel de méthode qui a été exécuté.



Paramètres d'entrée (type et valeur actuelle) pour la méthode englobante.

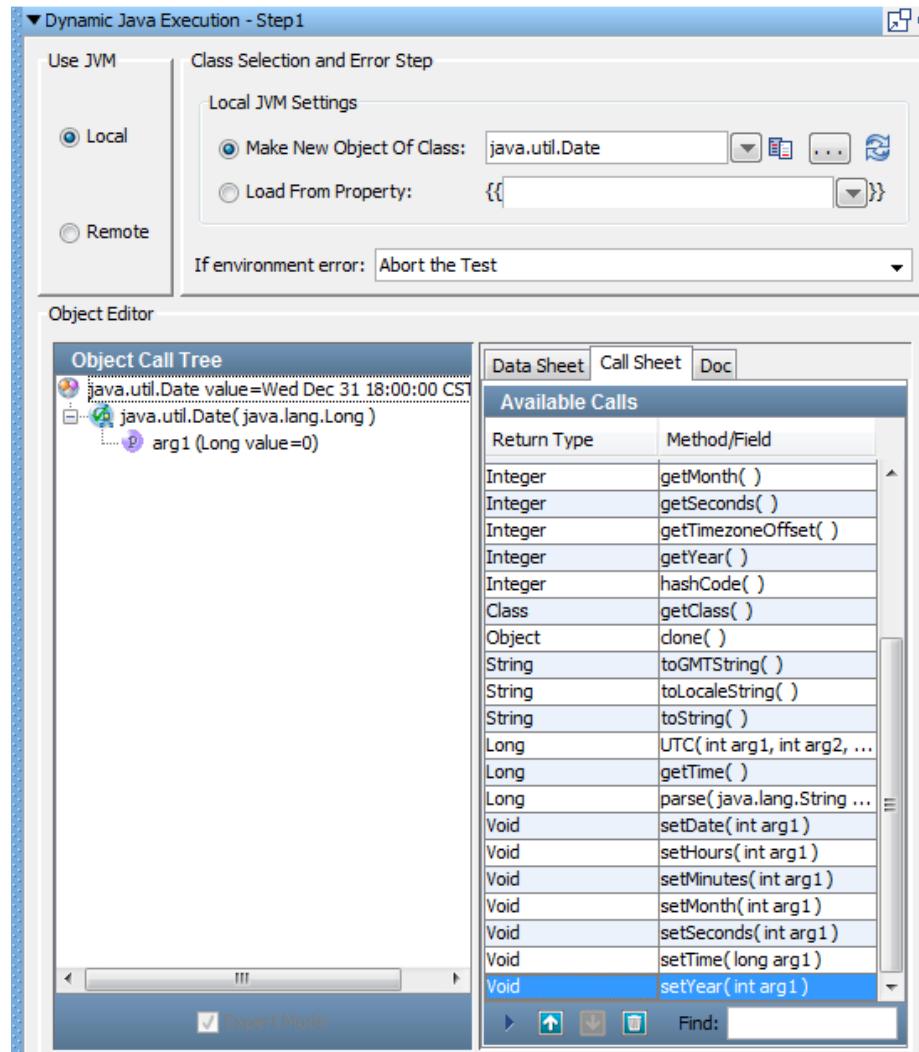


Valeur renvoyée (valeur actuelle si l'appel a été exécuté) pour la méthode englobante.

Le contenu du panneau droit varie selon la sélection effectuée dans le panneau gauche.

Pour appeler l'objet Java :

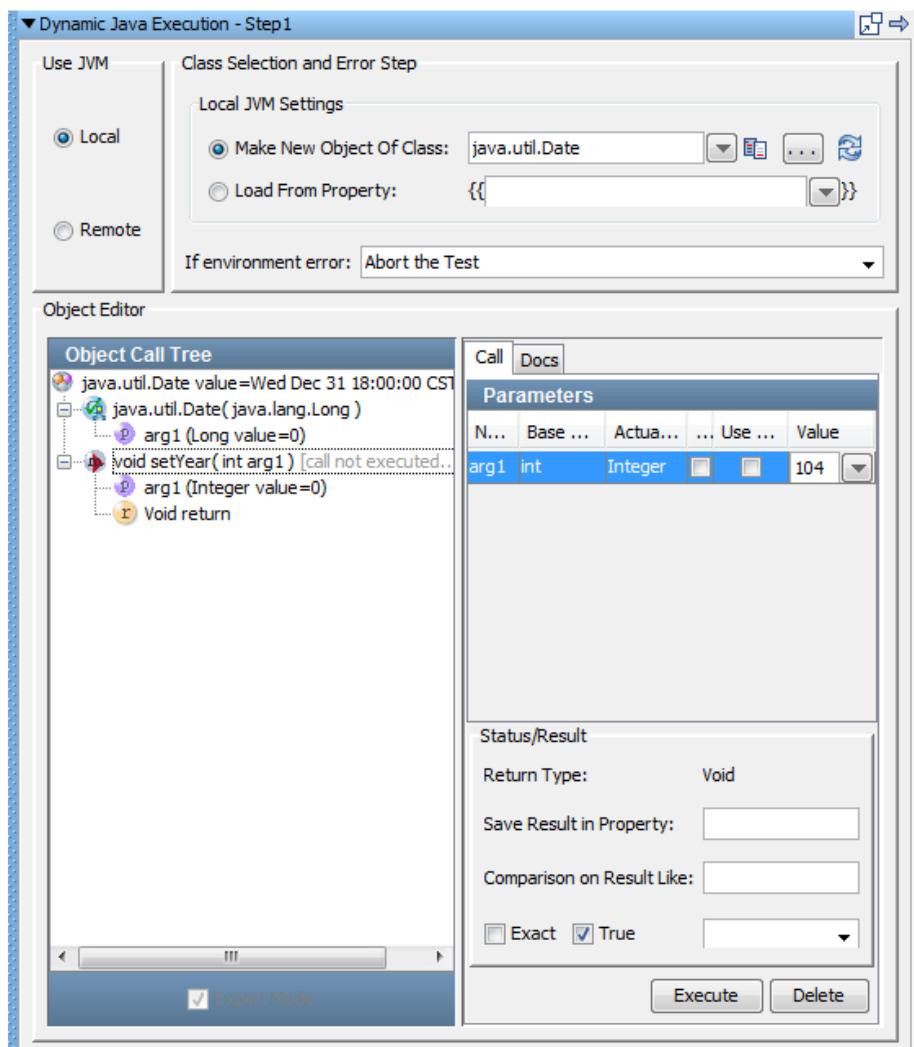
1. Dans le panneau droit de l'éditeur Complex Object Editor (Editeur d'objets complexes), cliquez sur l'onglet Call Sheet (Feuille d'appel).
L'onglet Call Sheet (Feuille d'appel) affiche les méthodes disponibles que vous pouvez appeler.
2. Double-cliquez sur la méthode `setYear()`. Vous pouvez également sélectionner la méthode `setYear()` et cliquer  sur Add selected method to Object Call Tree (Ajouter la méthode sélectionnée à l'arborescence des appels d'objet).



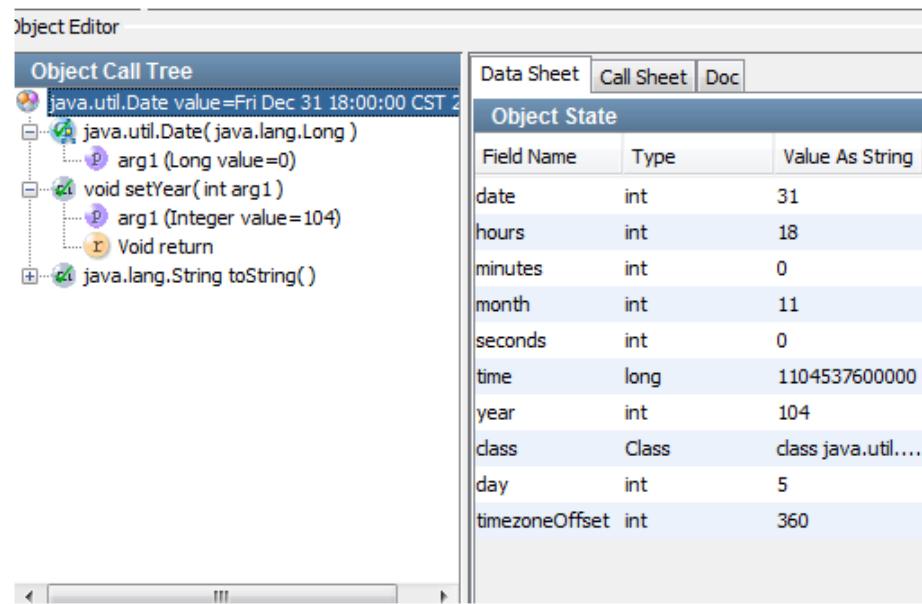
La méthode `setYear()` est ajoutée à l'arborescence Object Call Tree (Arborescence des appels d'objet). Le panneau droit affiche désormais les onglets Call (Appel) et Docs (Documents).

L'onglet Call (Appel) répertorie les informations d'argument.

3. Dans le champ Value (Valeur) pour arg1, entrez **104**.
4. Cliquez sur Execute (Exécuter).



5. Dans l'arborescence Object Call Tree (Arborescence des appels d'objet), sélectionnez l'objet **java.util.Date**.
6. Dans l'onglet Data Sheet (Feuille de données), vérifiez que le champ year est défini sur 104.

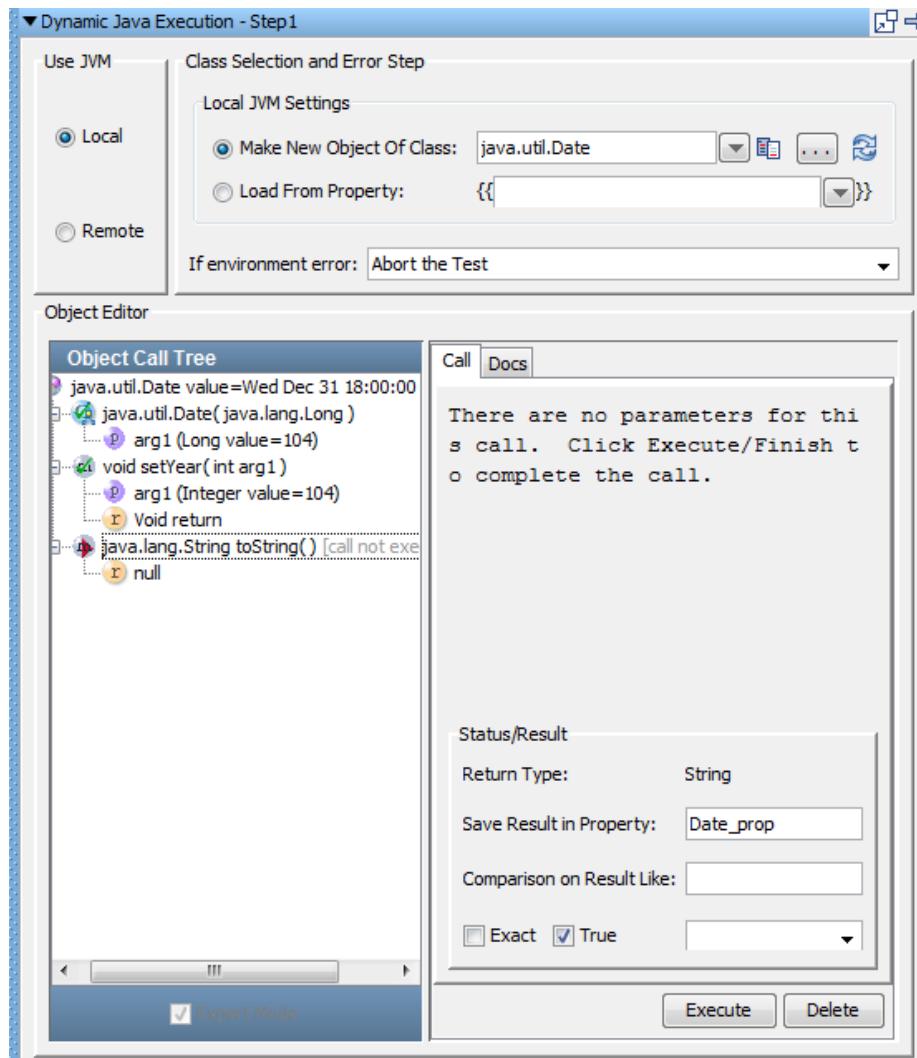


Etape 4 : ajout d'un filtre intégré

Vous pouvez ajouter un filtre intégré à partir de l'éditeur Complex Object Editor (Editeur d'objets complexes). Les filtres intégrés (et les assertions) ne conduisent pas à l'ajout d'un filtre à l'étape de test dans le panneau des éléments. La gestion des filtres intégrés est toujours effectuée dans l'éditeur Complex Object Editor.

Procédez comme suit:

1. Sélectionnez l'objet **java.util.Date**, puis cliquez sur l'onglet Call Sheet (Feuille d'appel).
2. Pour récupérer la date à inclure dans une propriété, appelez la méthode `toString()`.
Le panneau droit affiche désormais les onglets Call (Appel) et Docs (Documents).
3. Dans le champ Save Result in Property (Enregistrer le résultat dans la propriété) de la zone Status/Result (Statut/résultat) de l'onglet Call (Appel), ajoutez un filtre intégré en entrant **Date_prop** comme nom de propriété.



4. Cliquez sur Execute (Exécuter).



5. Cliquez sur Save Save (Enregistrer).

Etape 5 : vérification de la création de la propriété

Vous pouvez afficher la fenêtre Property Window (Fenêtre de propriété) pour vérifier que la propriété **Date_prop** a été créée.

Procédez comme suit:



1. Cliquez sur Show model properties (Afficher les propriétés du modèle) dans la barre d'outils du scénario de test. Vous pouvez également sélectionner Help (Aide), View Properties (Afficher les propriétés) à partir du menu principal.
2. Cliquez sur Refresh (Actualiser) .
3. Localisez la propriété **Date_prop**.

The screenshot shows the 'Property Window' dialog box. It has tabs for 'Properties' and 'Patterns', with 'Properties' selected. Below the tabs is a toolbar with a refresh button. The main area is a table titled 'Property Window' with columns 'Key' and 'Value'. The table contains the following data:

Key	Value
yearglobal	1999
config_prop	42
LISA_LAST_STEP	
LISA_DOC_PATH	C:\Lisa\Projects\My Tutorials\Tests
lisa.designtime.testcaseinfo	com.itko.lisa.editor.TestCaseInfo@ebacb7
Date_prop	Wed Dec 31 18:00:00 CST 1969
lisa.Step1.rsp	Wed Dec 31 18:00:00 CST 1969
LISA_TC_PATH	C:\Lisa\Projects\My Tutorials\Tests
robot	0
LISA_HOST	Diana-PC
LISA_PROJ_NAME	My Tutorials
instance	0
LISA_USER	arhoades@Diana-PC
testCaseId	336261653535642D643838352D3461
LISA_TC_URL	file:/C:/Lisa/Projects/My%20Tutorials/Tests
testCase	Test Case
LISA_PROJ_ROOT	C:/Lisa/Projects/My Tutorials
LISA_DOC_URL	file:/C:/Lisa/Projects/My%20Tutorials/Tests

4. Cliquez sur Close (Fermer).

Didacticiel 4 : résumé

Ce didacticiel vous a permis de :

- Créer une étape de test pour manipuler un objet Java de type **java.util.Date**
- Utiliser l'éditeur Complex Object Editor (Editeur d'objets complexes) pour manipuler l'objet Java
- Découvrir la procédure d'ajout de filtres intégrés à des objets et à des résultats d'enregistrement dans une propriété.

Didacticiel 5 : exécution d'une application Web de serveur de démonstration

Dans ce didacticiel, vous accédez à une application Web simple fournie avec DevTest.

L'application LISA Bank est un système frontal simple connecté à une table de base de données contenant des informations bancaires. La logique métier de l'application repose sur des objets Enterprise JavaBeans et des services Web. A partir de l'application Web, vous pouvez afficher le profil de l'utilisateur, créer un compte, ajouter des adresses, etc.

L'objectif de ce didacticiel est de vous familiariser avec l'application. Cette application est utilisée dans les didacticiels ultérieurs en tant que système testé.

Conditions préalables

- Le serveur de démonstration est en cours d'exécution.

Etape 1 : lancement de l'application Web

Procédez comme suit:

1. Ouvrez un nouveau navigateur.
2. Entrez l'URL suivante. Remplacez **localhost** par l'adresse IP de votre ordinateur.
`http://localhost:8080/lisabank/`

La page de connexion s'affiche.



The screenshot shows the LISAfinancial Online login interface. At the top right are links for [Help](#), [Log In](#), [Home](#), [Location Finder](#), [Contact Us](#), and [Printable](#). A welcome message "Welcome!" is displayed. The main content area features a banner with the text "Sound financial footing for the things that matter" and a rainbow over mountains. On the left, there's a login form with fields for Name (containing "admin") and Password (containing "*****"), and a "Login" button. Below the login form, it says "Tested by iTKO LISA™". On the right, there are two sections: "New at LISAfinancial" (listing "New Flexible-Interest Loan Plans", "Just added to MyMoney™: All-in-One Planning Dashboards", and "Q1 2007 Financial Results for LISAfinancial") and "Market Rates" (a table showing mortgage rates). The table has columns for Mortgage, Rate, APR, and Points.

Mortgage	Rate	APR	Points
1-Yr ARM	5.75%	7.02%	0.00%
3-Yr ARM	5.75%	6.79%	0.00%
5-Yr ARM	5.75%	6.60%	0.00%
30-Yr ARM	6.62%	6.65%	0.00%

Etape 2 - Connexion à l'application Web

Vous utilisez le nom d'utilisateur prédéfini **lisa_simpson**.

Procédez comme suit:

1. Dans le champ Name (Nom), saisissez **lisa_simpson**.
2. Dans le champ Password (Mot de passe), saisissez **golisa**.
3. Cliquez sur Login (Connexion).

La page d'accueil s'affiche. Le côté gauche contient des boutons pour différentes actions que vous pouvez effectuer : View Profile (Afficher le profil), New Account (Nouveau compte), Close Account (Clôturer le compte), Add Address (Ajouter une adresse), Delete Address (Supprimer une adresse) et Log Out (Déconnexion).

Account Number	Account Type	Name	Current Balance	Available Balance
No Accounts				
Total			\$0	\$0

MyMoney Tools

- [Mortgage Planner](#)
- [Retirement Planner](#)
- [Online Transactions](#)
- [Fund Allocation](#)
- [Identity Security](#)

Fund Statistics

Bar chart showing Fund Statistics by Region (East, West, North) across 1st, 2nd, 3rd, and 4th Quarters.

Region	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
East	30	45	55	20
West	40	35	40	30
North	50	55	50	45

Etape 3 : création d'un compte

Remarquez que l'utilisateur actuel n'a pas de compte.

Procédez comme suit:

1. Cliquez sur New Account (Nouveau compte).
2. Dans la liste Account Type (Type de compte), sélectionnez SAVINGS (Epargne).
3. Dans le champ Account Name (Nom du compte), entrez **Mes économies**.
4. Dans le champ Initial Balance (Solde initial), entrez **100.00**.
5. Cliquez sur Add Account (Ajouter un compte).

The screenshot shows the LISAfinancial Online application interface. At the top, there's a banner with a globe and navigation links: Help | Log Out, Home | Location Finder | Contact Us, and Welcome lisa simpson (lisa_simpson). The main content area has a left sidebar with buttons for View Profile, New Account (which is highlighted in blue), Close Account, Add Address, Delete Address, and Log Out. The right side shows a form titled 'New Account Details' with fields for Username (lisa_simpson), Account Type (SAVINGS), * Account Name (My Savings), and * Initial Balance (100). A large 'Add Account' button is at the bottom right of the form.

Le nouveau compte d'épargne est ajouté à la section Accounts (Comptes).

The screenshot shows the LISAfinancial Online application interface. At the top right, there are links for 'Help | Log Out'. Below that, the page title is 'LISAfinancial Online' and the sub-page title is 'MyMoney Home > Accounts'. The main content area displays a table of accounts:

Account Number	Account Type	Name	Current Balance	Available Balance
2500744684	SAVINGS	My Savings	\$100.00	\$100.00
Total			\$100.00	\$100.00

Below the account table, there are two sections: 'MyMoney Tools' and 'Fund Statistics'. The 'Fund Statistics' section contains a bar chart showing fund values across four quarters (1st, 2nd, 3rd, 4th) for three regions: East, West, and North.

The left sidebar contains several navigation links: View Profile, New Account, Close Account, Add Address, Delete Address, and Log Out. It also includes a 'Tested by iTKO LISA™' logo.

6. Pour créer deux comptes de plus, Compte_courant et Prêt_auto, répétez les étapes précédentes. Définissez le solde initial du compte courant sur 600.00 \$. Définissez le solde initial du prêt automobile sur 10000.00 \$.
N'utilisez pas de virgules dans le champ Initial Balance (Solde initial).

LISAfinancial Online

Help | Log Out

Home | Location Finder | Contact Us

Welcome lisa simpson (lisa_simpson)

Tuesday, Jun 28, 2011

Welcome lisa !

[View Profile](#)

[New Account](#)

[Close Account](#)

[Add Address](#)

[Delete Address](#)

[Log Out](#)

MyMoney Home >

Accounts

Account Number	Account Type	Name	Current Balance	Available Balance
2500744684	SAVINGS	My Savings	\$100.00	\$100.00
2572255363	CHECKING	My Checking	\$600.00	\$600.00
2588563046	AUTO_LOAN	My Car Loan	\$10000.00	\$10000.00
Total			\$10700.00	\$10700.00

MyMoney Tools

- [Mortgage Planner](#)
- [Retirement Planner](#)
- [Online Transactions](#)
- [Fund Allocation](#)
- [Identity Security](#)

Fund Statistics

Quarter	East	West	North
1st Qtr	25	35	45
2nd Qtr	20	40	45
3rd Qtr	10	30	90
4th Qtr	15	25	40

Etape 4 - clôture d'un compte

L'application vous permet de clôturer des comptes.

Procédez comme suit:

1. Cliquez sur Close Account (Clôturer le compte).
2. Sélectionnez **Mes économies** dans la liste et cliquez sur Select Account (Sélectionner un compte).

Select Account	
Select Account	<input type="text" value="2500744684 (My Savings)"/>
<input type="button" value="Select Account"/>	
Account Details	
Username	<input type="text" value="lisa_simpson"/>
Account ID	<input type="text" value="2500744684"/>
Account Type	<input type="text" value="SAVINGS"/>
Account Name	<input type="text" value="My Savings"/>
Balance	<input type="text" value="100.0"/>
<input type="button" value="Confirm Delete"/>	

3. Cliquez sur Confirm Delete (Confirmer la suppression).

Le compte **Mes économies** est supprimé de la section Accounts (Comptes).

Etape 5 : déconnexion de l'application Web

Pour vous déconnectez de l'application Web :

1. Cliquez sur Log Out (Déconnexion).

Didacticiel 5 : résumé

Ce didacticiel vous a permis de :

- Vous connecter à l'application LISA Bank
- Créer des comptes
- Clôturer un compte

Didacticiel 6 : test d'un site Web

Dans ce didacticiel, vous utilisez l'enregistreur Web pour enregistrer le chemin via un site Web. Créez ensuite les étapes de test HTTP/HTML Request (Demande HTTP/HTML) pour chaque paire requête-réponse HTTP.

L'étape de test HTTP/HTML Request (Demande HTTP/HTML) permet d'effectuer des demandes auprès d'un serveur Web et de recevoir les résultats d'un scénario de test. Pour vérifier que les pages fonctionnent comme prévu, testez un site Web simple.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Créer un scénario de test qui contient des étapes HTTP/HTML Request (Demande HTTP/HTML) avec l'enregistreur Web
- Modifier et exécuter le scénario de test produit par l'enregistreur
- Ajouter un ensemble de données au scénario de test

Conditions préalables

- Vous avez effectué le didacticiel 5 sur l'exécution d'une application Web de serveur de démonstration
- DevTest Workstation est ouvert.
- Vous avez accès au serveur de démonstration.

Parties du didacticiel

- [Didacticiel 6, partie A : enregistrement du scénario de test](#) (page 65)
- [Didacticiel 6, partie B : exécution du scénario de test](#) (page 72)
- [Didacticiel 6, partie C : modification des étapes de test de demande](#) (page 73)

Didacticiel 6, partie A : enregistrement du scénario de test

A l'aide de l'enregistreur Web, créez un scénario de test contenant des étapes HTTP/HTML Request (Demande HTTP/HTML).

Etape 1 : création d'un scénario de test

Procédez comme suit:

1. Dans le projet Mes didacticiels, créez un scénario de test nommé Didacticiel 6a dans le sous-dossier Tests.

L'éditeur de modèles s'ouvre.

Etape 2 : démarrage de l'enregistreur Web

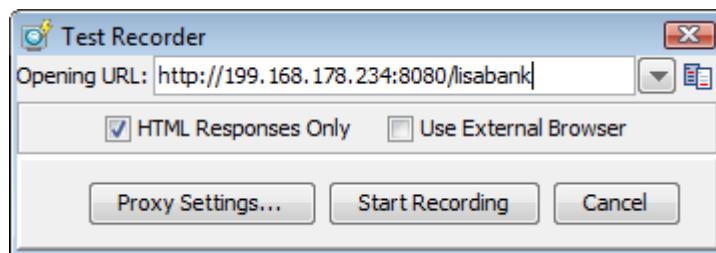
Dans ce didacticiel, vous enregistrez un site Web via le proxy HTTP.

Procédez comme suit:

1. Dans le menu principal, sélectionnez Actions, Record Test Case for User Interface (Enregistrer le scénario de test pour l'interface utilisateur), Web Recorder (HTTP proxy) (Enregistreur Web (proxy HTTP)).

La boîte de dialogue Test Recorder (Enregistreur de test) s'ouvre.

2. Dans le champ Opening URL (URL à ouvrir), entrez l'URL suivante : Remplacez l'adresse IP du chemin par l'adresse IP de votre ordinateur.



3. Cliquez sur Start Recording (Lancer l'enregistrement).

La fenêtre Test Recorder (Enregistreur de test) s'ouvre. La fenêtre Test Recorder contient les onglets Browser (Navigateur) et Recorded Elements (Eléments enregistrés). La page de connexion de l'application LISA Bank s'affiche dans l'onglet Browser.

Etape 3 : enregistrement de l'application LISA bank

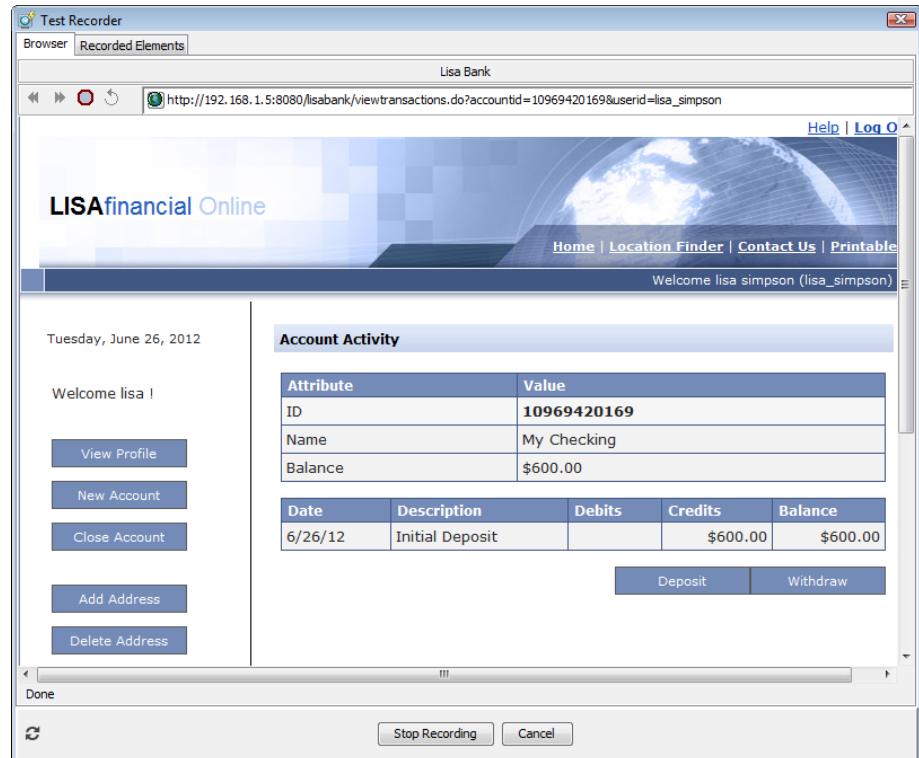
Lorsque vous effectuez des actions dans l'application LISA Bank, les informations de demande et de réponse pour chaque page visitée sont enregistrées.

Procédez comme suit:

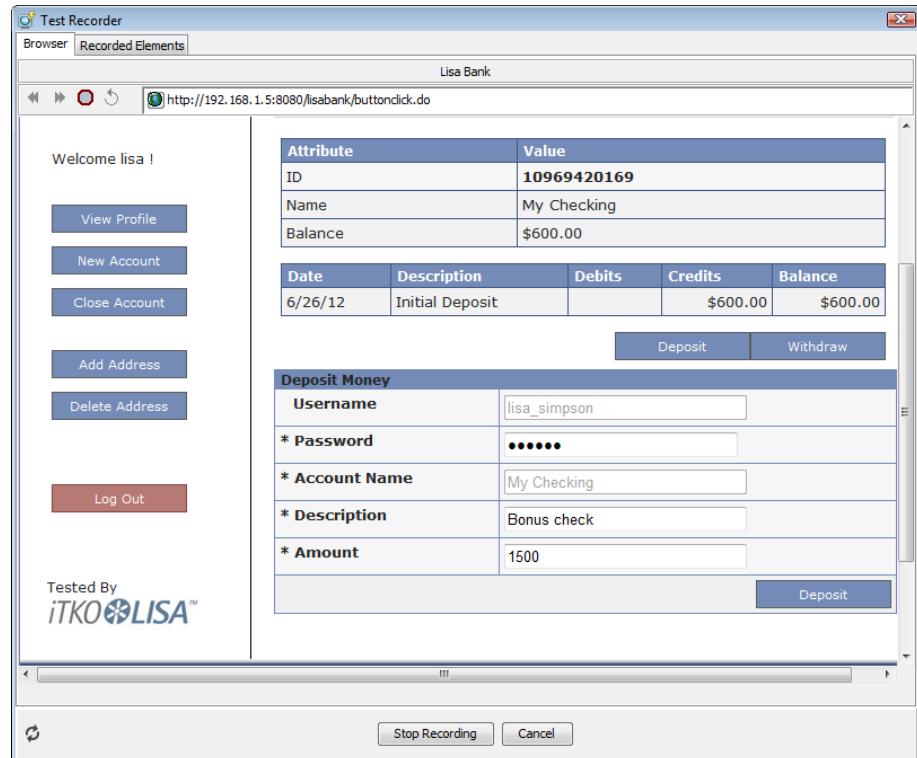
1. Dans le champ Name (Nom), saisissez **lisa_simpson**. Dans le champ Password (Mot de passe), saisissez **golisa**.



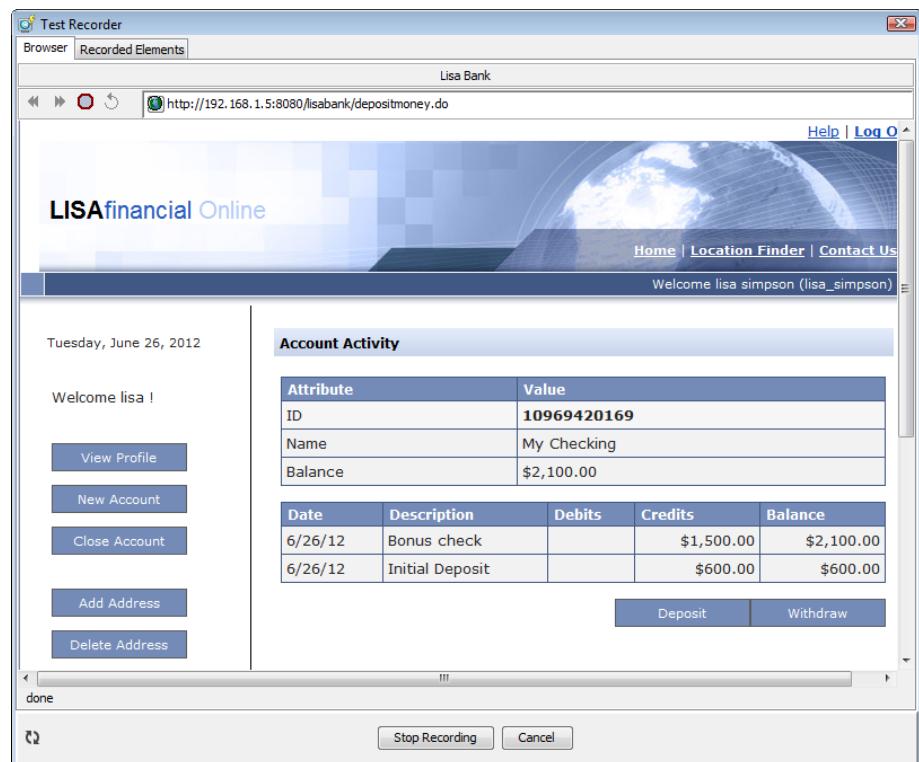
2. Cliquez sur Login (Connexion).
La page d'accueil s'affiche.
3. Dans la section Accounts (Comptes), cliquez sur le numéro du compte courant.
La fenêtre Account Activity (Activité du compte) s'affiche.
4. Cliquez sur Deposit (Déposer).



5. Dans la zone Deposit Money (Déposer de l'argent), entrez le mot de passe **golisa**, une description de la transaction et le montant du dépôt.
6. Cliquez sur Deposit (Déposer).



La fenêtre Account Activity indique le solde mis à jour et un enregistrement du dépôt.



7. A partir du volet de navigation gauche, cliquez sur Log Out (Déconnexion).

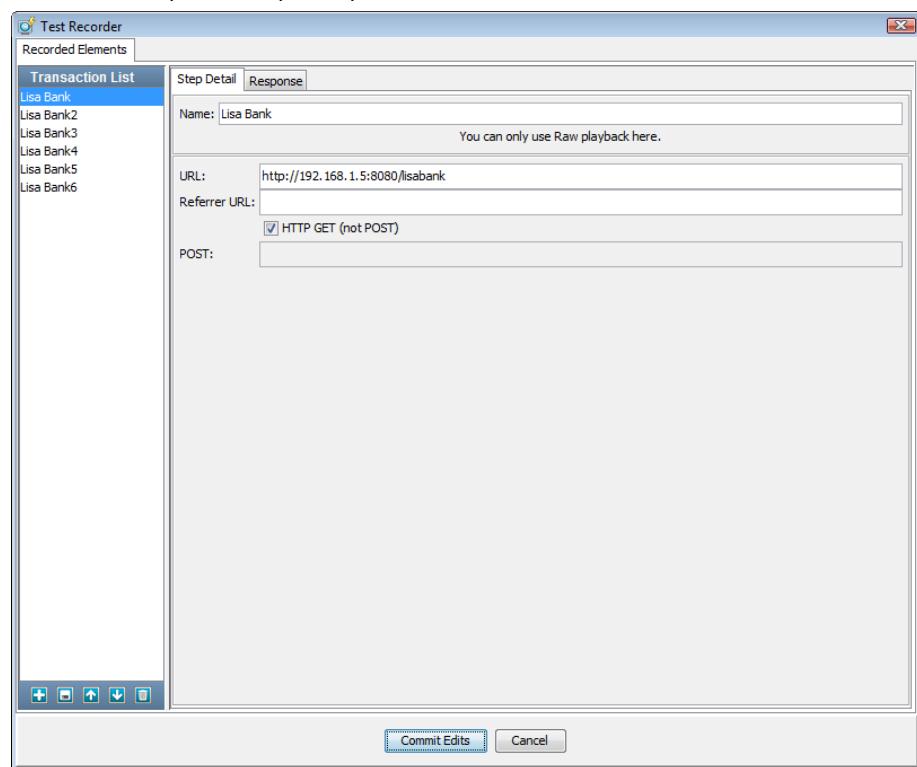
Etape 4 : arrêt de l'enregistreur Web

Après avoir arrêté l'enregistrement, vous pouvez afficher des détails sur les transactions.

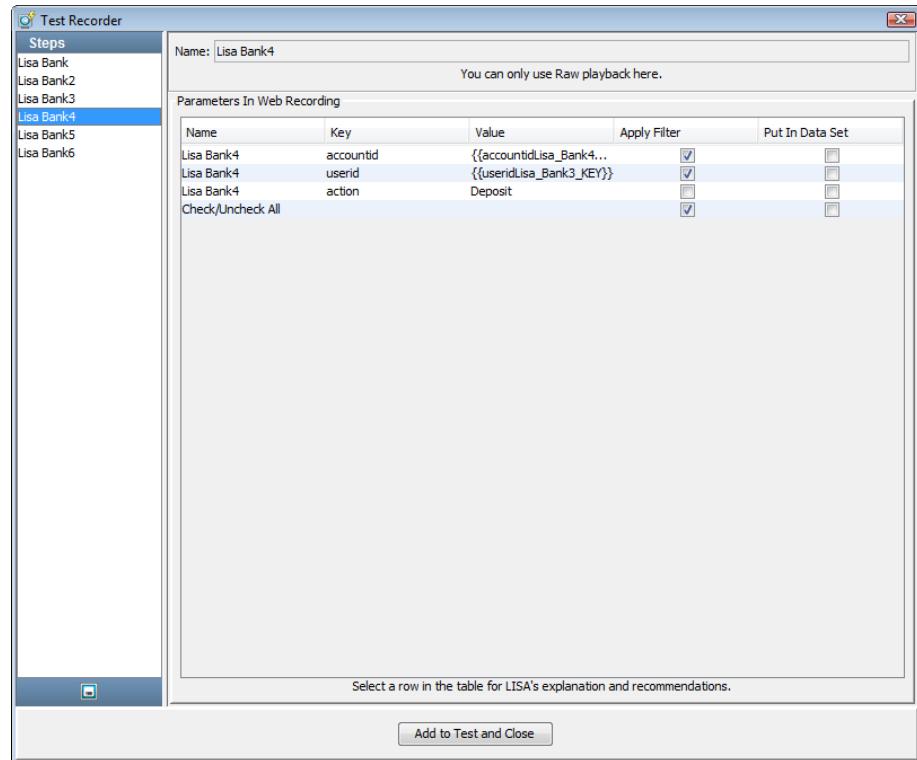
Procédez comme suit:

1. Au bas de la fenêtre Test Recorder (Enregistreur de test), cliquez sur Stop Recording (Arrêter l'enregistrement).

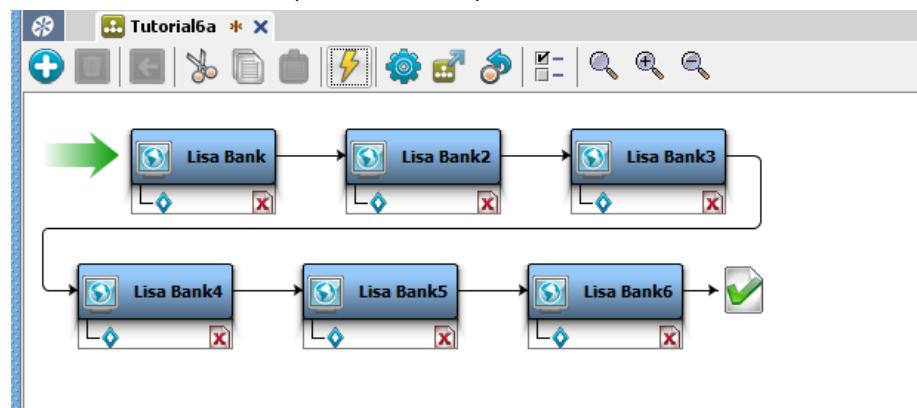
Les filtres et les propriétés sont automatiquement créés pour les références de page Web. Les champs de formulaire pour le dépôt sont également affichés. Le volet gauche affiche une liste de transactions (étapes). Le volet droit affiche les détails de l'étape et la réponse pour la transaction sélectionnée.



2. Cliquez sur Commit Edits (Valider les modifications).
La page de paramètres s'affiche.
3. Cliquez sur Add to Test and Close (Ajouter au test et fermer).



L'éditeur de modèles est rempli avec un nouveau scénario de test, avec toutes les informations de transaction de l'enregistrement enregistré. Chaque étape de test dans le scénario de test représente une requête HTTP.



4. Enregistrez le scénario de test.

Didacticiel 6, partie B : exécution du scénario de test

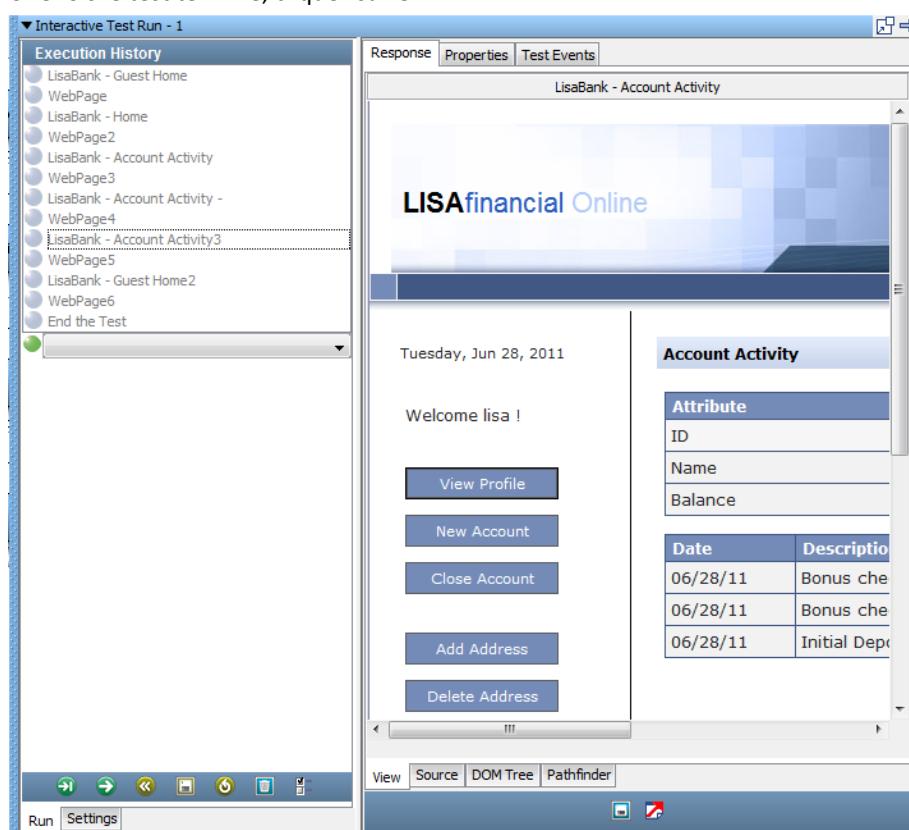
Dans cette section qui suit la partie A, exécutez le scénario de test enregistré dans l'utilitaire ITR (Exécution d'un test interactif) et afficher les résultats.

Procédez comme suit:

1. Démarrez une nouvelle session d'ITR.



2. Dans le volet Execution History (Historique d'exécution), cliquez sur Automatically execute test (Exécuter automatiquement le test).
3. Une fois le test terminé, cliquez sur OK.



L'onglet View (Afficher) contient les pages renvoyées lorsque l'ITR lit le dépôt dans le compte courant.

L'onglet Source indique le code HTML de la page capturée au cours de l'étape.

DevTest est utilisé comme navigateur et envoie les mêmes demandes HTTP au serveur Web.

4. Fermez l'utilitaire ITR.

Didacticiel 6, partie C : modification des étapes de test de demande

L'enregistreur Web produit une étape de test HTTP/HTML Request (Demande HTTP/HTML) pour chaque demande.

Vous pouvez modifier ces étapes de test, comme les autres étapes de test présentes dans DevTest. L'enregistreur utilise les paramètres entrés lors de l'enregistrement comme valeurs pour les paramètres POST et GET dans la demande.

Pour généraliser le test LISA Bank, remplacez les valeurs de montant du dépôt et de description codées de manière irréversible (par exemple, espèces et 1 000.00) par des propriétés d'un ensemble de données. Vous avez utilisé auparavant des ensembles de données dans le didacticiel 2 sur les ensembles de données.

Dans l'étape du test LISA Bank5 à partir des résultats de l'enregistrement, les paramètres Host Name (Nom d'hôte) et Port sont paramétrés et ajoutés à la configuration. Les valeurs de la description et du montant sont codées de manière irréversible.

Dans cette partie du didacticiel, vous utilisez un ensemble de données de calcul numérique pour paramétriser le scénario de test de sorte qu'il dépose différentes sommes d'argent. Lorsque vous exécutez le scénario de test, des valeurs de dépôt différentes de celles enregistrées sont utilisées.

Etape 1 : copie d'un scénario de test

Procédez comme suit:

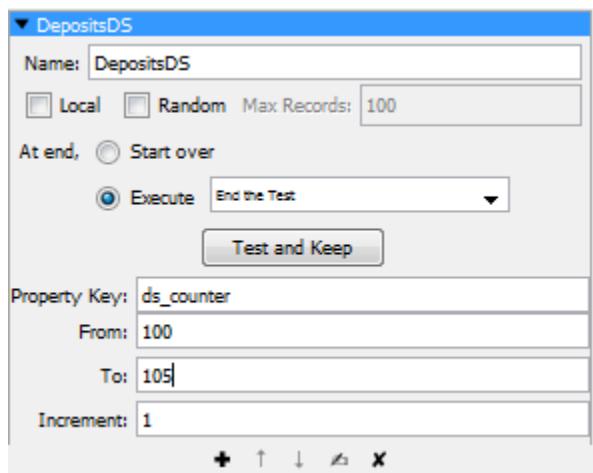
1. Vérifiez que le scénario de test Didacticiel 6a est ouvert dans l'éditeur de modèles.
2. Dans la barre de menu, sélectionnez File (Fichier), Save As (Enregistrer sous).
3. Dans le champ File Name (Nom de fichier), entrez Didacticiel 6c.
4. Cliquez sur Save (Enregistrer).

Etape 2 : ajout de l'ensemble de données

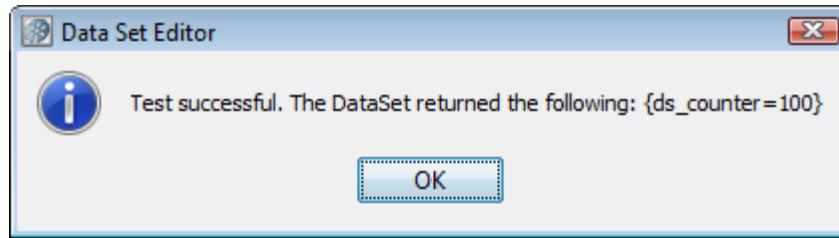
Dans la procédure suivante, vous ajoutez un ensemble de données de calcul numérique. Ce type d'ensemble de données vous permet d'affecter un nombre à une propriété. Vous pouvez modifier le nombre par une valeur fixe chaque fois que l'ensemble de données est utilisé.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez la première étape de test.
 2. Dans le volet droit, double-cliquez sur l'onglet d'étape Data Sets (Ensembles de données).
 3. Cliquez sur  Add (Ajouter) sous l'élément Data Sets.
 4. Dans la liste Common Datasets (Ensembles de données communs), sélectionnez Create a numeric counting Data Set (Créer un ensemble de données de calcul numérique).
- L'éditeur d'ensembles de données s'ouvre dans le volet droit.
5. Entrez les valeurs suivantes :
 - a. Dans le champ Name (Nom), saisissez DepositsDS.
 - b. Dans le champ At end (A la fin), sélectionnez l'option Execute (Exécuter) et End the Test (Arrêter le test) dans la liste.
 - c. Dans le champ Property Key (Clé de la propriété), entrez ds_counter.
 - d. Dans le champ From (De), entrez 100.
 - e. Dans le champ To (A), entrez 105.
 - f. Dans le champ Increment (Incrément), entrez 1.



6. Pour tester l'ensemble de données, cliquez sur Test and Keep (Tester et conserver). La première ligne de données dans l'ensemble de données est affichée dans un message de confirmation.



- Cliquez sur OK.

Etape 3 : modification des paramètres POST pour le dépôt enregistré

Utilisez la propriété ds_counter, que vous avez créée dans l'ensemble de données, pour spécifier des montants d'argent variables pour le dépôt.

Procédez comme suit:

- Dans l'éditeur de modèles, double-cliquez sur l'étape LISA Bank5.
- Dans la zone POST Parameters (Paramètres POST), remplacez la valeur de la clé de description par **deposit {{ds_counter}}** (déposer {{ds_counter}}).
- Changez la valeur de la clé de montant par **{{ds_counter}}**.

The screenshot shows the 'HTTP/HTML Request - Lisa Bank5' configuration window. It includes fields for Protocol (http), Host Name ({{SERVER}}), Port (opt) ({{PORT}}), and Path (/lisabank/depositmoney.do). Below these are two tables: 'URL Parameters' and 'POST Parameters'. The 'POST Parameters' table has three entries: accountname ({{accountnameLisa_Bank5_KEY}}), description (deposit {{ds_counter}}), and amount ({{ds_counter}}). To the right of the tables is a 'All Known State' table showing various environment variables. At the bottom, there are tabs for 'URL Transaction Info', 'HTTP Headers', and 'Response'.

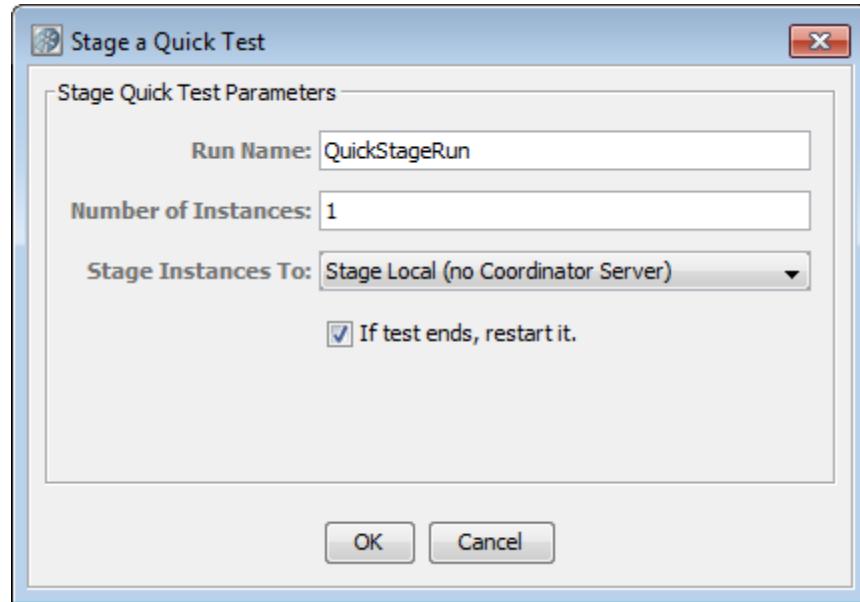
Key	Value
SERVER	192.168.1.5
PORT	8080
lisa.desig...	com.itko.li...
robot	0
LISA_TC_...	C:\Lisa\Pr...
LISA_HOST	Diana-PC
LISA_PRO...	My Tutorials
LISA_USER	arhoades
instance	0
lisa.Lisa B...	<!-- jspst...
ds_counter	100
testCaseId	37376665...
LISA_TC_...	file:/C:/Li...
testCase	Test Case
LISA_PRO...	C:/Lisa/Pr...

- Enregistrez le scénario de test.

Etape 4 : simulation du scénario de test

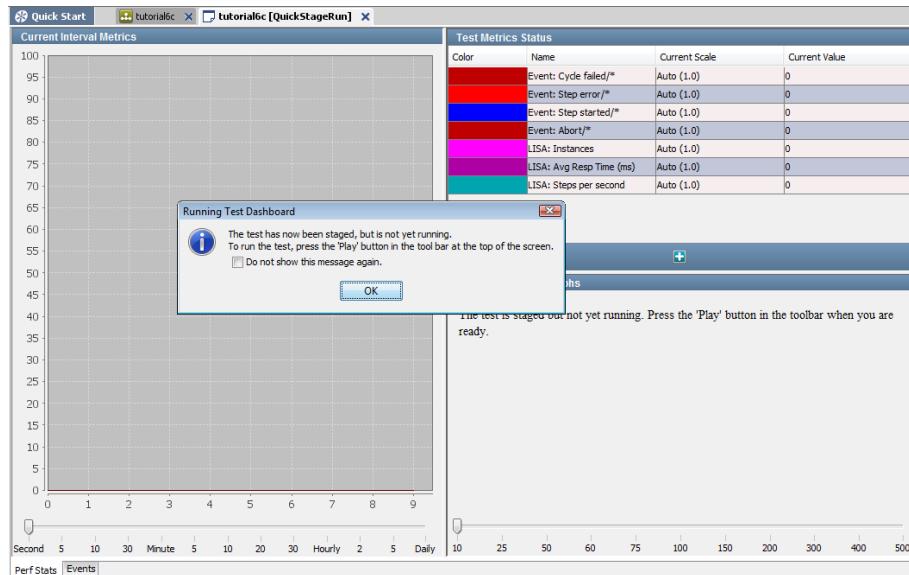
Pour simuler (ou exécuter) un test rapide :

1. Dans la barre d'outils, cliquez sur  Stage a quick test (Simuler un test rapide).
2. Dans la fenêtre Stage a Quick Test, vérifiez que l'option If test ends, restart it (Si le test se termine, redémarrez-le).



3. Cliquez sur OK.

Le moniteur Test Monitor (Moniteur de tests) s'ouvre, mais le test n'a pas encore démarré.

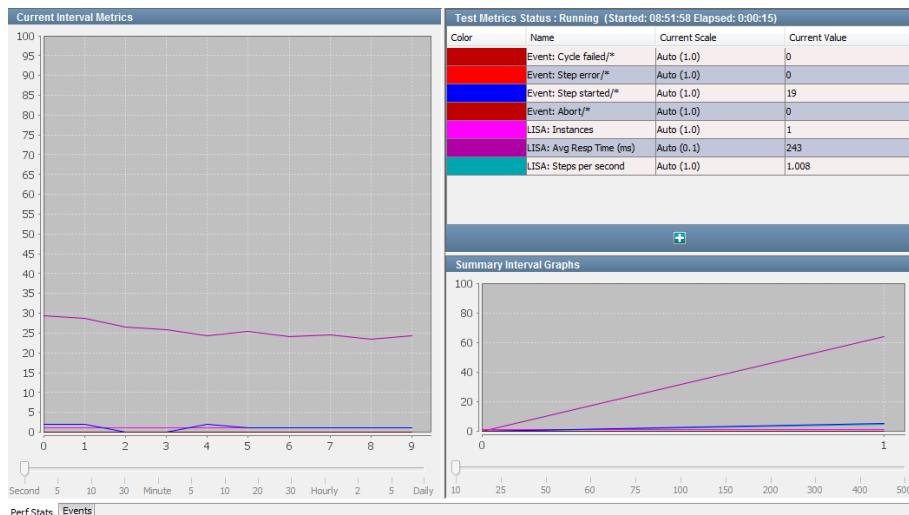


4. Cliquez sur OK.



5. A partir de la barre d'outils, cliquez sur Play (Lire).

Les graphiques linéaires indiquent la progression du test.



Etape 5 : affichage des nouveaux dépôts dans LISA Bank

Procédez comme suit:

1. Connectez-vous à l'application LISA Bank à nouveau à l'aide de l'utilisateur lisa_simpson et du mot de passe golisa.
2. Pour afficher les dépôts, cliquez sur le lien de numéro de compte pour le compte courant.

Notez que le début des dépôts commencent par 100 et augmentent de 1 jusqu'à atteindre 105.

The screenshot shows the LISA financial Online interface. At the top right, there are links for 'Help | Log Out'. Below that, a welcome message 'Welcome lisa simpson (lisa_simpson)'. On the left, a sidebar displays the date 'Tuesday, June 26, 2012' and a list of account management options: 'View Profile', 'New Account', 'Close Account', 'Add Address', 'Delete Address', and 'Log Out'. Below this is a 'Tested By' section with the iTKO LISA logo. The main content area is titled 'Account Activity'. It shows a summary table with columns 'Attribute' and 'Value' containing information about the account: ID (13060025435), Name (My Checking), and Balance (\$2,715.00). Below this is a detailed transaction history table with columns 'Date', 'Description', 'Debits', 'Credits', and 'Balance'. The transactions listed are: deposit 105, deposit 104, deposit 103, deposit 102, deposit 101, deposit 100, Bonus Check, and Initial Deposit. At the bottom right of the main area are 'Deposit' and 'Withdraw' buttons.

Didacticiel 6 : résumé

Dans ce didacticiel, vous avez utilisé un ensemble de données de calcul numérique pour fournir des données au test enregistré.

Ce didacticiel vous a permis de :

- Copier un scénario de test et d'ajouter un ensemble de données de calcul numérique
- Modifier les paramètres POST pour le dépôt enregistré
- Simuler un test rapide

Didacticiel 7 : test d'un objet Enterprise JavaBean (EJB)

L'application LISA Bank fournit un ensemble complet d'objets EJB pour interagir avec un compte, obtenir des informations sur l'utilisateur et le compte à partir de l'interface Java.

Dans ce didacticiel, vous utilisez l'étape de test d'exécution d'objet Enterprise JavaBean pour appeler des méthodes EJB dans un scénario de test et tester la réponse avec une assertion. Testez un simple objet EJB pour vérifier que les méthodes addUser et deleteUser fonctionnent comme prévu.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Utiliser l'étape Execute an Enterprise Java Bean (Exécuter un objet EJB (Enterprise JavaBean))
- Utiliser l'éditeur Complex Object Editor (Editeur d'objets complexes) avec des objets EJB

Conditions préalables

- Vous avez effectué le didacticiel 6 permettant de tester un site Web.
- DevTest Workstation est ouvert.
- Vous avez accès au serveur de démonstration.

Etape 1 : création d'un scénario de test

Procédez comme suit:

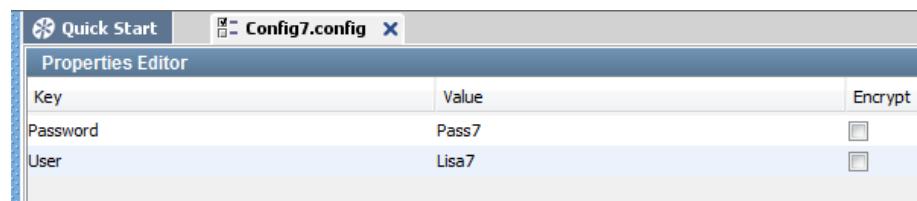
1. Dans le volet Project (Projet), cliquez avec le bouton droit de la souris sur le dossier Tests et sélectionnez Create New Test Case (Créer un scénario de test).
2. Définissez le nom de fichier sur Didacticiel 7.
3. Cliquez sur Save (Enregistrer).

Etape 2 : création d'une configuration

Vous avez utilisé préalablement des configurations dans le didacticiel 2 sur les ensembles de données.

Procédez comme suit:

1. Ouvrez le fichier project.config.
2. Si la configuration ne contient pas les propriétés User (Utilisateur) et Password (Mot de passe), ajoutez-les. Vous ne devez pas définir les valeurs.
3. Créez une configuration nommée config7.
4. Ajoutez la propriété User à la configuration config7 et définissez la valeur sur **Lisa7**.
5. Ajoutez la propriété Password à la configuration config7 et définissez la valeur sur **Pass7**.



Key	Value	Encrypt
Password	Pass7	<input type="checkbox"/>
User	Lisa7	<input type="checkbox"/>

6. Cliquez sur Save (Enregistrer).
7. Dans le volet Project (Projet), cliquez avec le bouton droit de la souris sur la configuration config7 et sélectionnez Make Active (Activer).

La configuration s'affiche désormais en violet.

Etape 3 : ajout d'une étape de test EJB

L'étape de test Enterprise JavaBean Execution (Exécution d'un objet EJB) vous permet d'effectuer des appels sur un objet EJB en cours d'exécution.

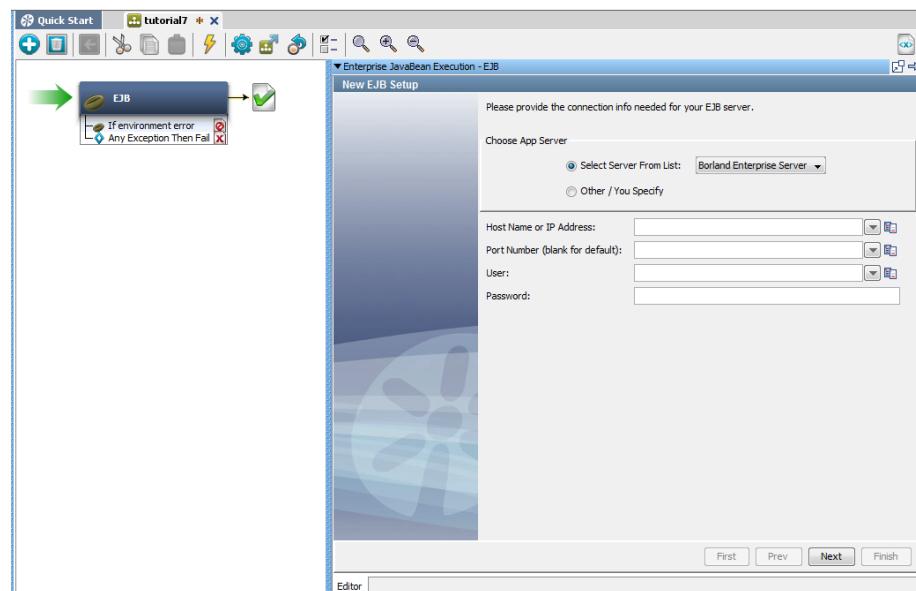
Procédez comme suit:

1. Cliquez sur l'onglet Didacticiel 7.



2. Cliquez sur Add Step (Ajouter une étape).
3. Sélectionnez Java/J2EE, puis Enterprise JavaBean Execution.

L'assistant New EJB Setup (Configuration d'un nouveau serveur EJB) s'affiche.



Etape 4 : connexion au serveur

L'assistant New EJB Setup (Configuration d'un nouveau serveur EJB) vous invite à spécifier les informations de connexion pour le serveur EJB.

Procédez comme suit:

1. A partir de la liste déroulante Select Server From List (Sélectionner un serveur dans la liste), sélectionnez JBoss 3.2/4.0.
2. Dans le champ Host Name or IP Address (Nom d'hôte ou adresse IP), entrez **localhost**.
3. Cliquez sur Next (Suivant).

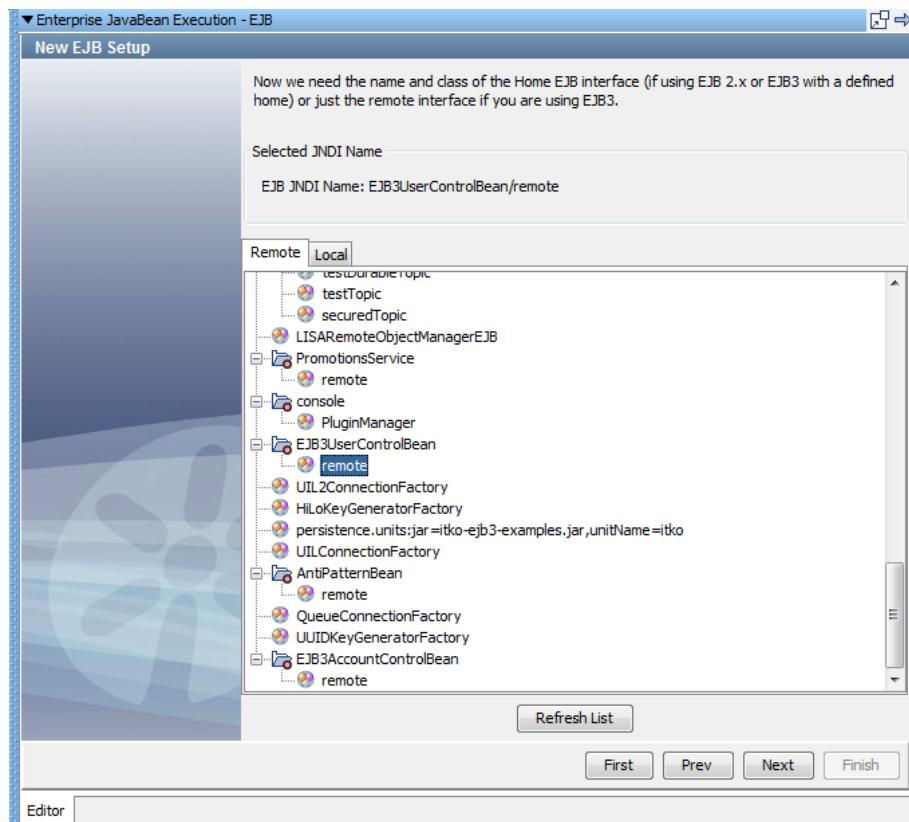
La liste de noms JNDI est récupérée à partir du serveur EJB.

Etape 5 : localisation de l'interface EJB

L'assistant New EJB Setup (Configuration d'un nouveau serveur EJB) vous invite à spécifier le nom de l'interface EJB.

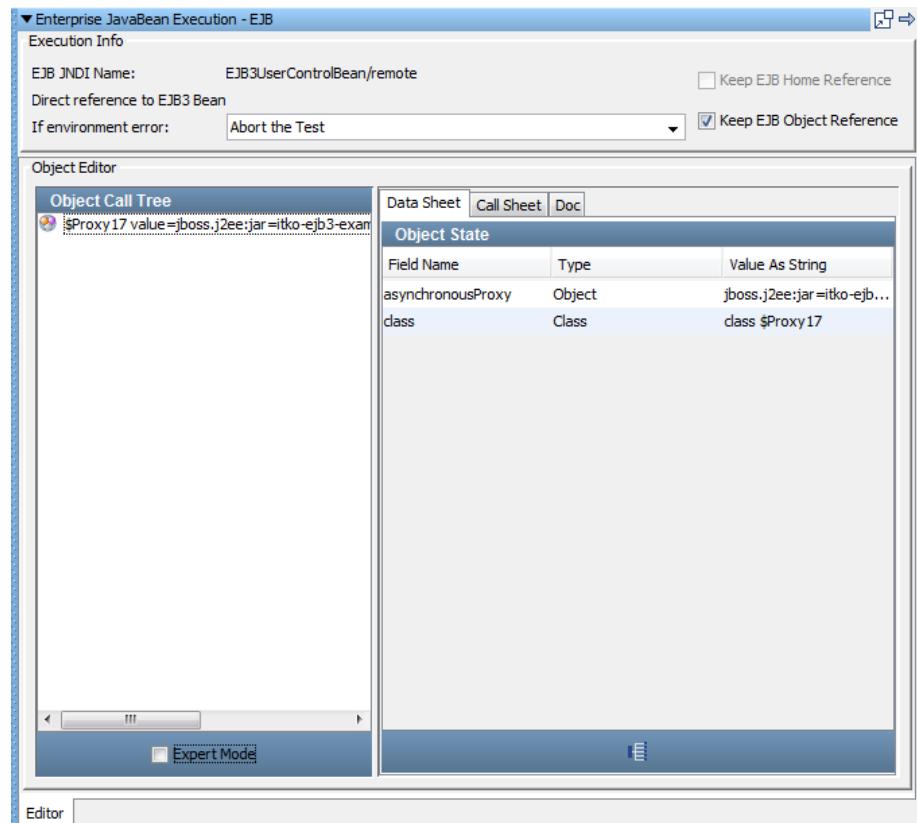
Procédez comme suit:

1. Dans l'onglet Remote (Distant), sélectionnez EJB3UserControlBean/remote.



2. Cliquez sur Next (Suivant).

L'éditeur Complex Object Editor (Editeur d'objets complexes) s'ouvre.



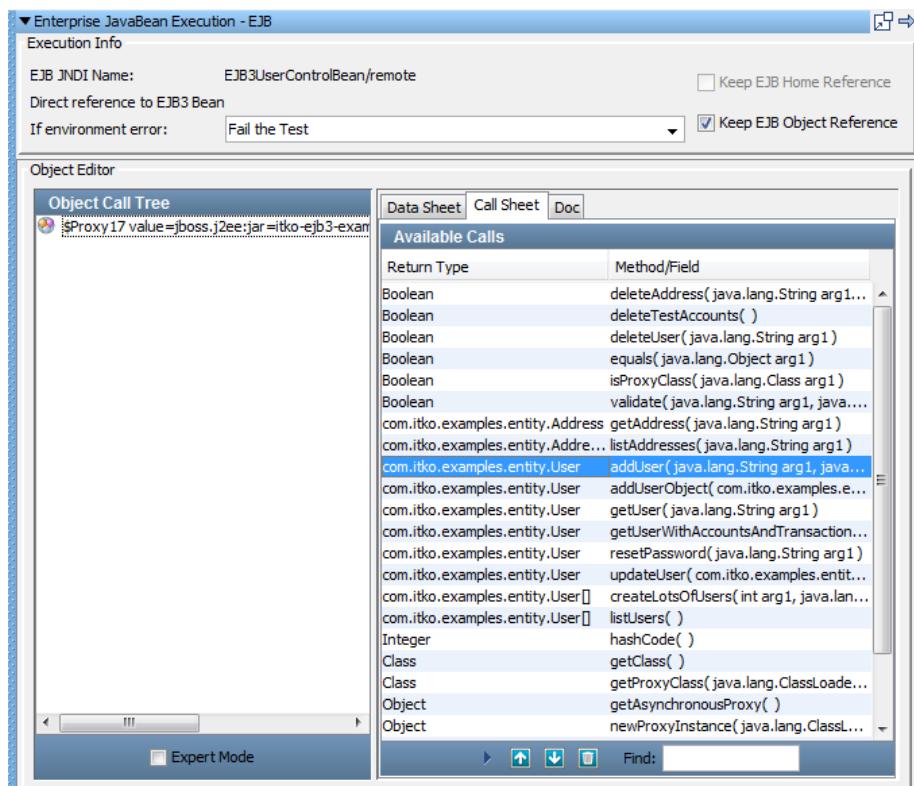
Etape 6 : configuration de l'étape EJB

Procédez comme suit:

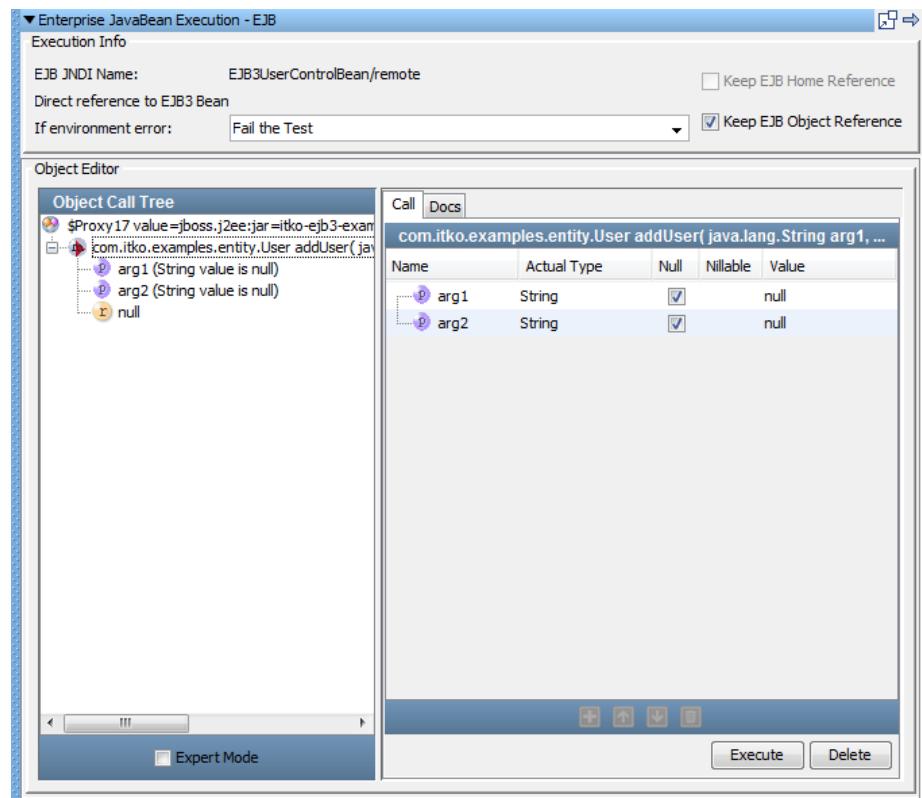
1. Si vous utilisez le même objet EJB à intervalles réguliers, sélectionnez la case à cocher Keep EJB Object Reference (Conserver la référence à l'objet EJB), si ce n'est pas déjà le cas.
2. Définissez le champ If environment error (En cas d'erreur d'environnement) sur l'étape à exécuter si une exception se produit lors de l'exécution de cette étape EJB. Dans la liste, sélectionnez Fail the Test (Faire échouer le test).
3. Dans la zone Object Editor (Editeur d'objets), sélectionnez l'onglet Call Sheet (Feuille d'appel), puis la méthode addUser.



4. Cliquez sur Add selected method to Object Call Tree (Ajouter la méthode sélectionnée à l'arborescence des appels d'objet).



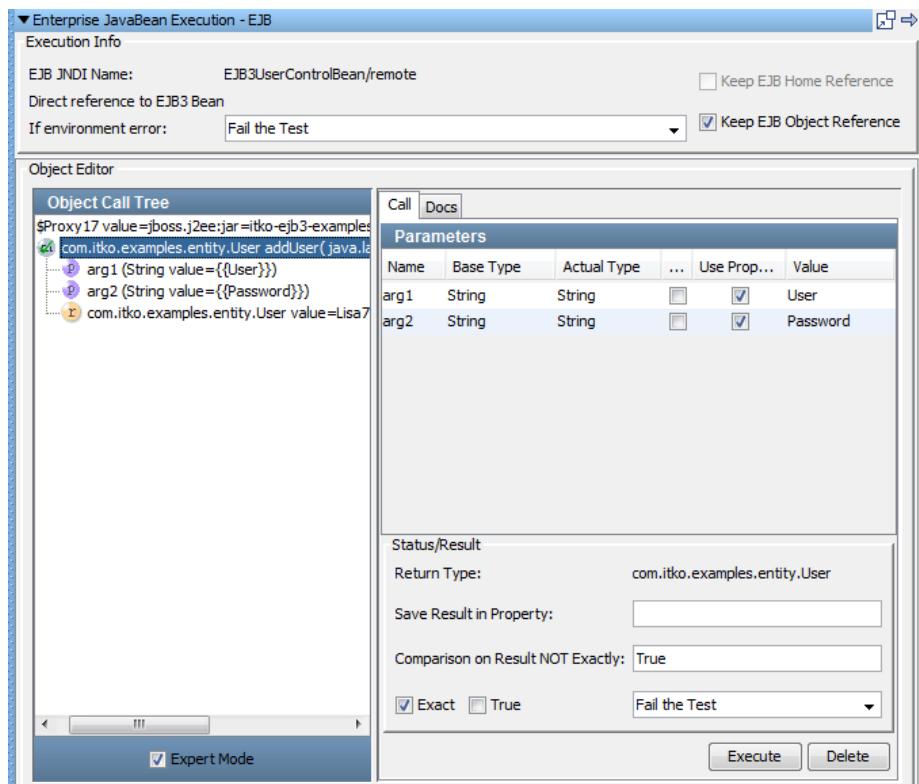
L'arborescence Object Call Tree (Arborescence des appels d'objet) affiche la méthode addUser.



Important : Vous pouvez uniquement ajouter l'utilisateur et le mot de passe une fois. Si vous voulez exécuter ce didacticiel plusieurs fois, modifiez les valeurs associées à l'utilisateur et au mot de passe.

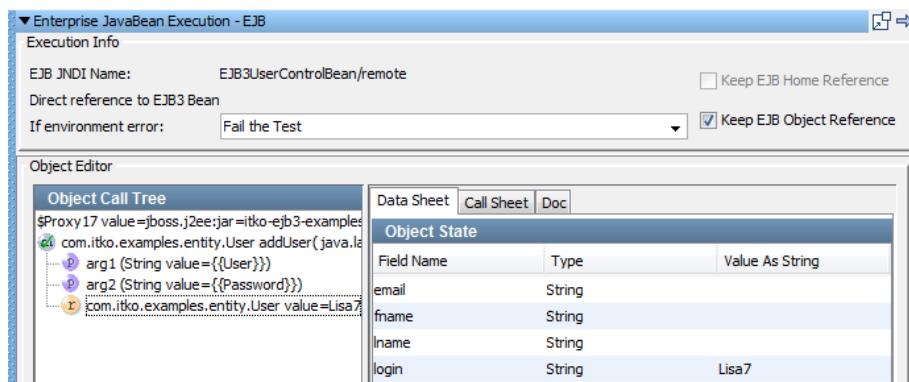
Pour entrer les paramètres de méthode et ajouter une assertion intégrée, procédez comme suit :

1. Dans le volet Object Call Tree (Arborescence des appels d'objet), sélectionnez Expert Mode (Mode Expert), s'il n'est pas déjà activé.
2. Cochez les cases Use Property (Utiliser une propriété) pour chaque argument. Use Property est un en-tête de colonne de la zone Parameters (Paramètres).
3. Dans la colonne Value (Valeur) pour arg1, sélectionnez **User** (Utilisateur) dans la liste de propriétés Defaults (Valeurs par défaut).
4. Dans la colonne Value (Valeur) pour arg2, sélectionnez **Password** (Mot de passe) dans la liste de propriétés Defaults (Valeurs par défaut).
5. Dans la zone Status/Result (Statut/résultat), ajoutez l'assertion intégrée en sélectionnant Exact et en décochant la case True.
6. Dans le champ Comparison on Result NOT Exactly (Comparaison en cas de non-correspondance exacte du résultat), entrez **True**.
7. A partir de la liste déroulante Exact, sélectionnez Fail the Test (Faire échouer le test).
8. Cliquez sur Execute (Exécuter).



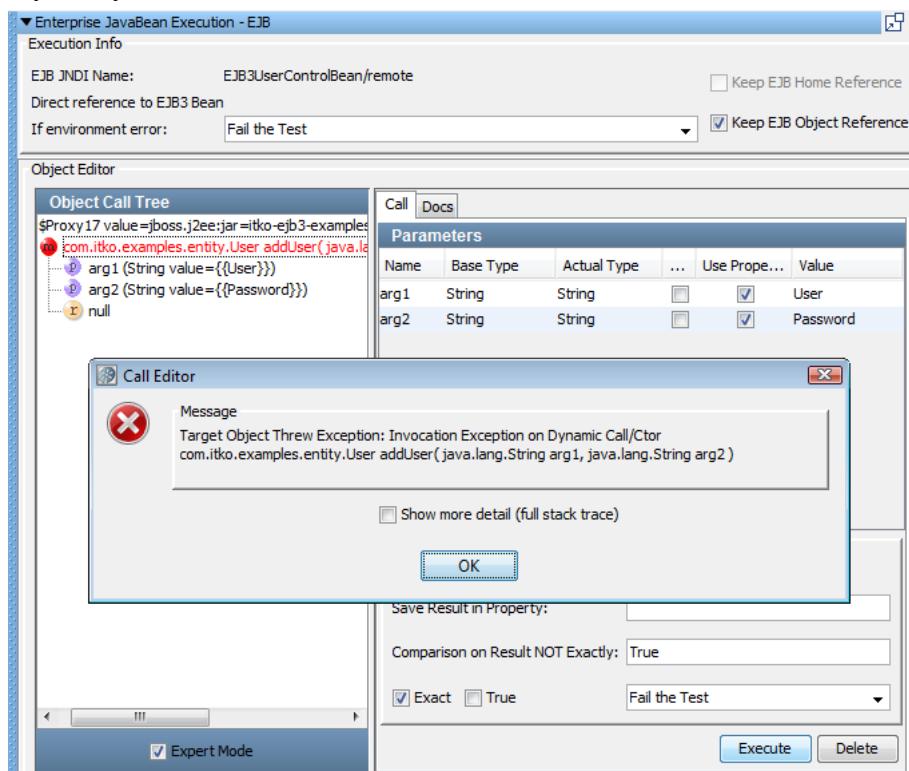
Les paramètres de la méthode sont affichés dans le volet Object Call Tree (Arborescence des appels d'objet), à côté de l'icône Input Parameter (Paramètre

d'entrée) . La valeur renvoyée pour cette méthode est la valeur du champ User (Utilisateur), Lisa7, dans l'arborescence Object Call Tree.



9. Testez la méthode addUser à nouveau en cliquant sur Execute (Exécuter).

10. La valeur renvoyée devient nulle et une erreur se produit, car l'utilisateur a déjà été ajouté.



Save Result in Property:

Comparison on Result NOT Exactly: True

Exact True

If the Test: Fail the Test

OK

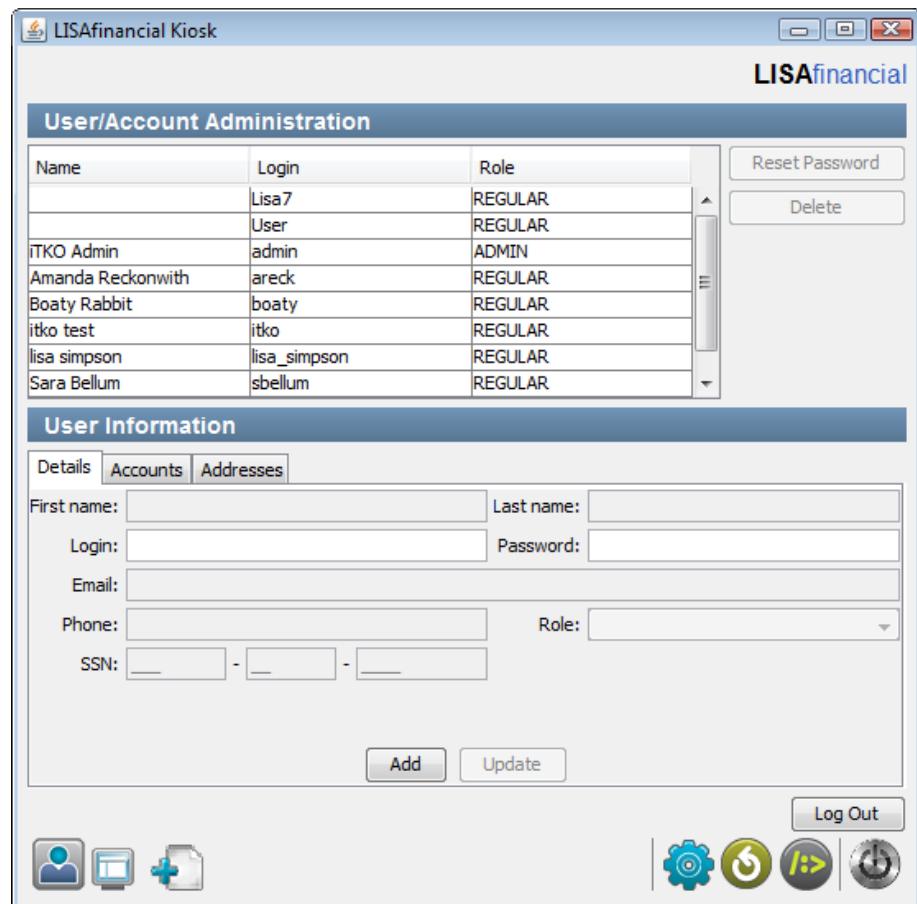
Execute Delete

Etape 8 : vérification de l'exécution de la méthode

A partir de l'application LISA Bank, vous pouvez vérifier que l'utilisateur a été ajouté.

Procédez comme suit:

1. Accédez à l'application LISA Bank.
2. Connectez-vous en tant qu'**administrateur** avec le mot de passe **admin**.
3. Pour confirmer que l'utilisateur Lisa7 a été ajouté, affichez la liste des utilisateurs.

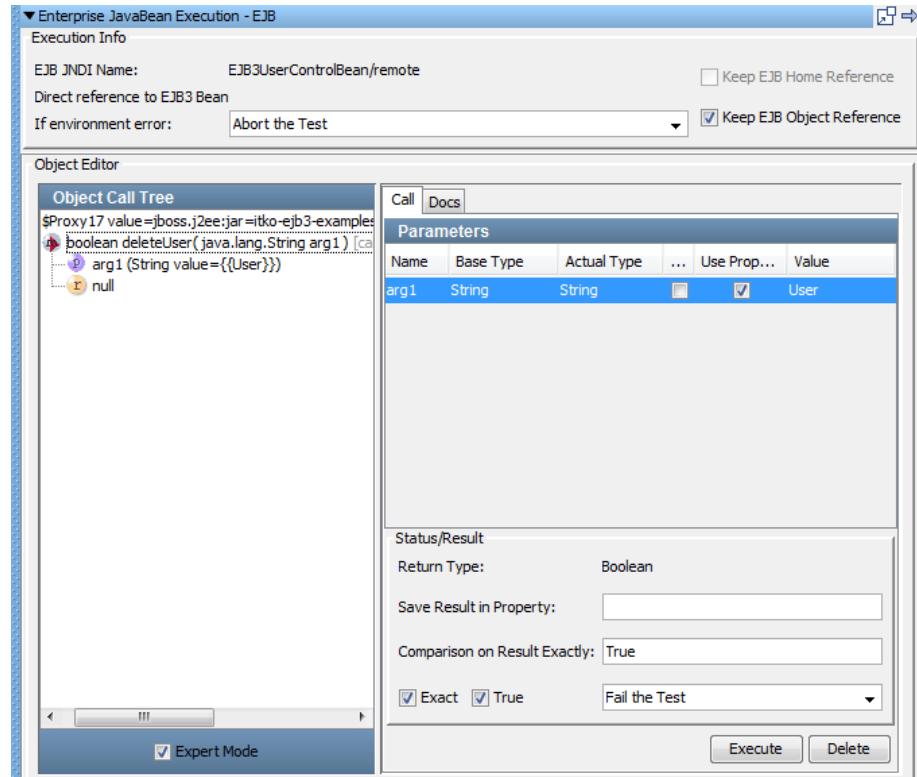


Etape 9 : ajout d'une autre étape de test EJB

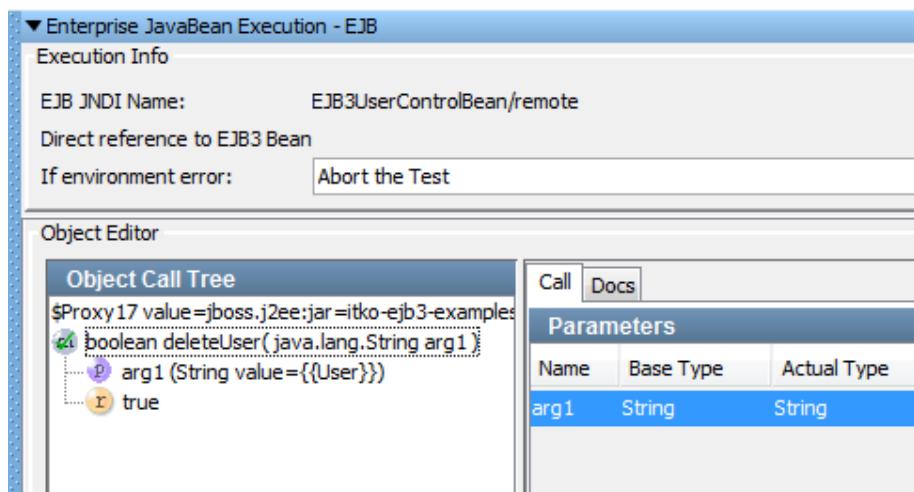
Effectuez les étapes précédentes à nouveau pour appeler la méthode deleteUser.

Procédez comme suit:

1. Répétez le didacticiel en commençant à l'étape 3 pour ajouter une étape EJB nommée DeleteUser.
2. Utilisez la propriété de paramètre de méthode User (Utilisateur).



3. Cliquez sur Execute (Exécuter) pour exécuter la méthode et obtenir des résultats.



4. Cliquez sur Save (Enregistrer).

Didacticiel 7 : résumé

Ce didacticiel vous a permis de :

- Créer un scénario de test consistant en deux étapes de test EJB
L'objet EJB a été chargé à partir de l'exemple d'application sur le serveur de démonstration.
- Créer une étape de test EJB et charger un objet EJB
- Utiliser l'éditeur Complex Object Editor (Editeur d'objets complexes) pour manipuler des objets EJB.

Didacticiel 8 : test d'un service Web

Dans ce didacticiel, utilisez l'étape de test Web Service Execution (XML) (Exécution de service Web XML) pour appeler des opérations de service Web dans un scénario de test. Testez ensuite la demande et la réponse. Ces opérations de service Web fournissent la même fonctionnalité que les appels de méthode équivalents dans l'objet EJB utilisé dans le didacticiel 7.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Ajouter une étape de test Web Service Execution (XML) (Exécution de service Web XML)
- Exécuter une opération de service Web

Conditions préalables

- Vous avez terminé le didacticiel 5.
- DevTest Workstation est ouvert.
- Vous avez accès au serveur de démonstration.

Etape 1 : création d'un scénario de test

Procédez comme suit:

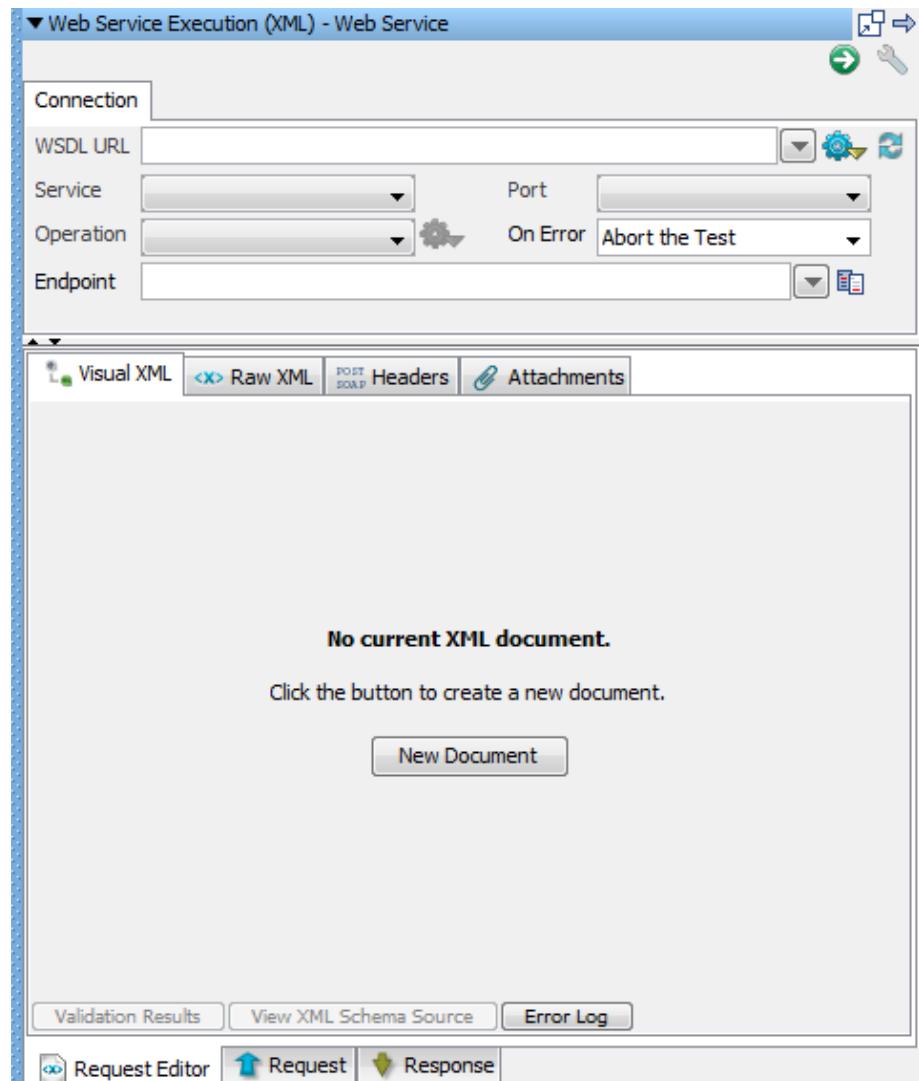
1. Cliquez avec le bouton droit de la souris sur le dossier Tests dans le panneau Project (Projet), puis cliquez sur Create New Test Case (Créer un scénario de test).
2. Définissez le nom de fichier sur **tutorial8** (didacticiel 8).
3. Cliquez sur Save (Enregistrer).
4. Dans le panneau Project cliquez avec le bouton droit de la souris sur project.config et sélectionnez Make active (Activer).

Etape 2 : ajout d'une étape de test Web Service Execution (XML) (Exécution de service Web XML)

L'étape de test Web Service Execution (XML) (Exécution de service Web XML) vous permet d'exécuter une opération sur un service Web SOAP.

Procédez comme suit:

1. Cliquez sur l'onglet Didacticiel 8.
2. Cliquez sur  Add Step (Ajouter une étape).
3. Sélectionnez Web/Web Services (Web/Services Web), puis Web Service Execution (XML).
Une étape de service Web est ajoutée à l'éditeur de modèles.
4. Pour ouvrir l'éditeur de l'étape Web Service Execution (XML), double-cliquez sur l'étape Web Service (Service Web).



5. Cliquez sur New Document (Nouveau document).

Etape 3 : création d'un client de service Web

Spécifiez l'opération à appeler, puis créez un message SOAP à envoyer à l'opération.

Procédez comme suit:

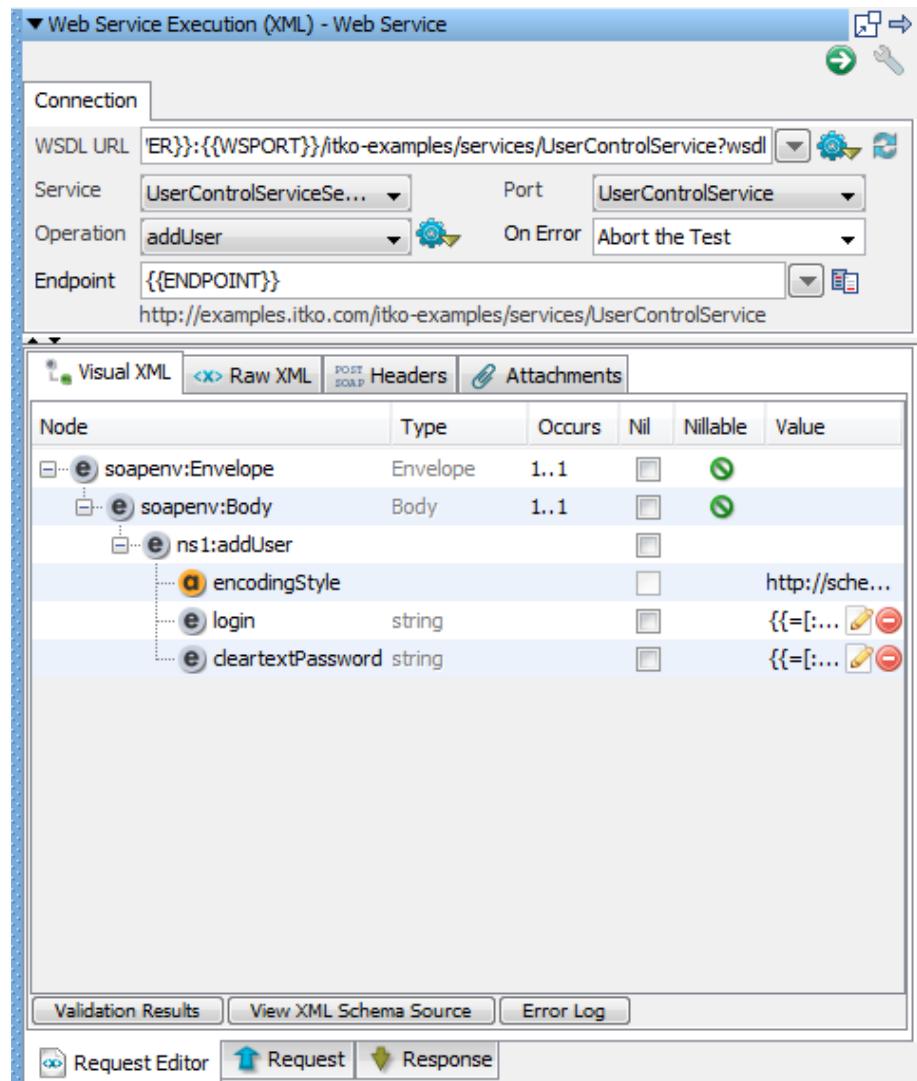
1. Dans le champ WSDL URL (URL du document WSDL), entrez l'emplacement suivant.

Remarque : Les propriétés WSSERVER et WSPORT représentent le serveur et le port.

`http://localhost:8080/itko-examples/services/UserControlService
?wsdl`

2. Dans le champ Service, sélectionnez UserControlServiceService.
3. Dans le champ Port, sélectionnez UserControlService.
4. Dans le champ Operation, sélectionnez l'opération addUser.
5. Dans le champ On Error (En cas d'erreur), sélectionnez Abort the test (Interrompre le test).

DevTest utilise ces critères pour créer le client de service Web. L'éditeur Visual XML (Code visuel XML) affiche une vue graphique du message SOAP.



6. Enregistrez le scénario de test.

Etape 4 : exécution de la demande de service Web

Procédez comme suit:

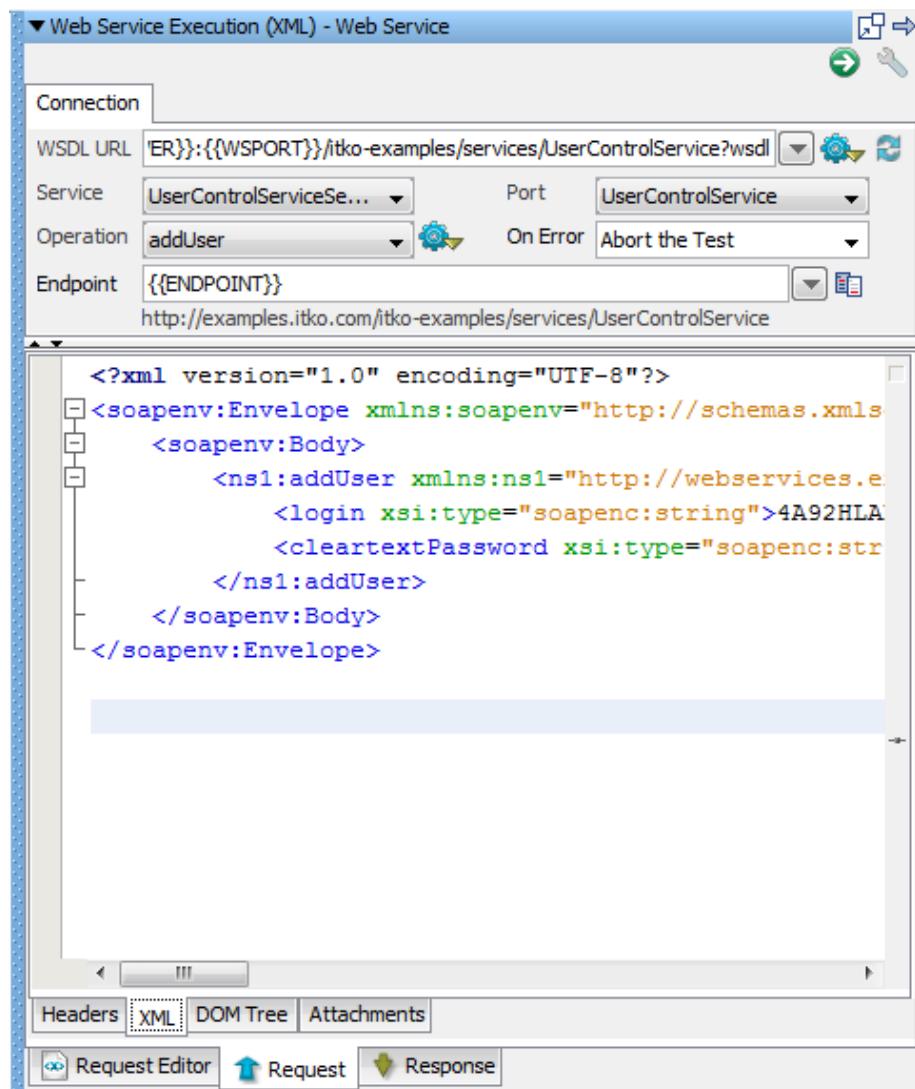
- Cliquez sur Execute WS Request (Exécuter une demande de service Web).
Le test est exécuté.

Etape 5 : affichage de la demande et de la réponse

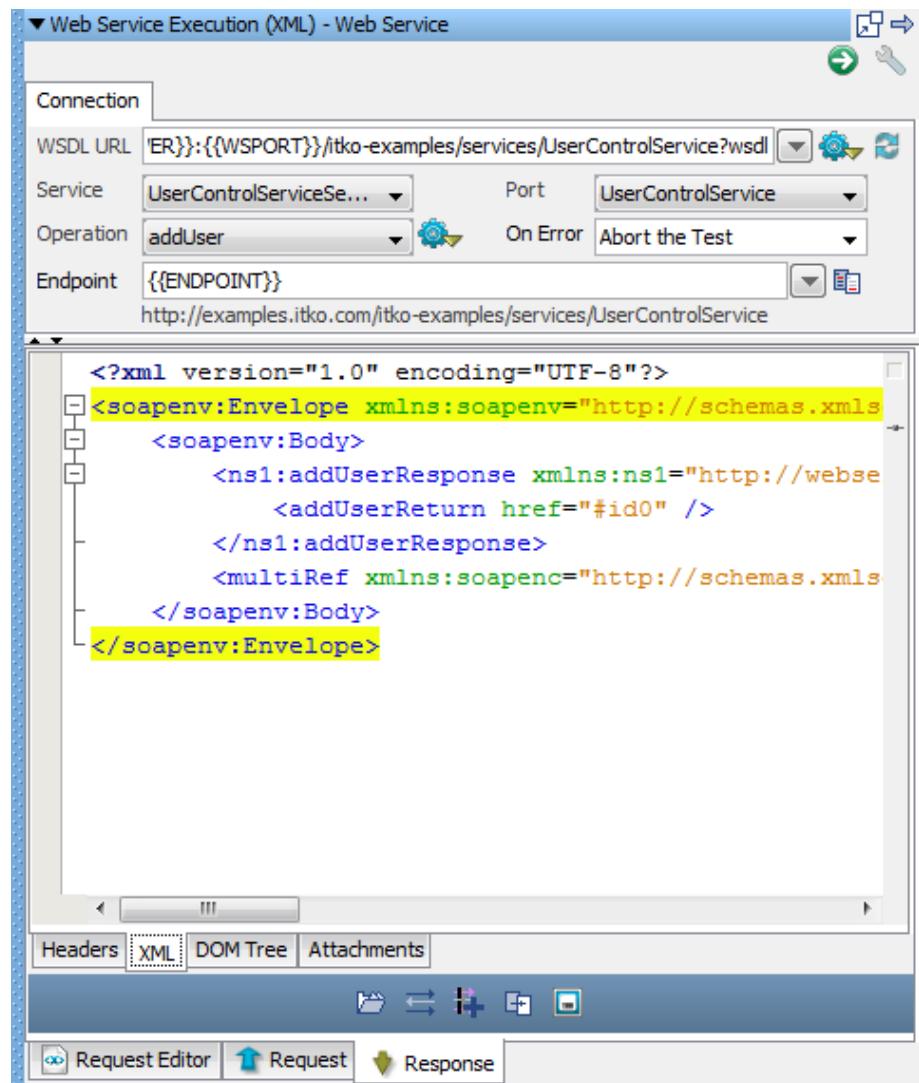
L'onglet Request (Demande) contient les données de demande résultantes envoyées après une tâche de post-traitement (par exemple, la substitution des propriétés DevTest). L'onglet Response (Réponse) affiche les données de la réponse qui ont été reçues.

Procédez comme suit:

1. Pour afficher la demande lors de l'exécution, cliquez sur l'onglet Request (Demande).



2. Pour afficher la réponse lors de l'exécution, sélectionnez l'onglet Response (Réponse).



Didacticiel 8 : résumé

Ce didacticiel vous a permis de :

- Créer un scénario de test avec l'étape de test Web Service Execution (XML) (Exécution de service Web XML)
- Exécuter l'opération addUser
- Afficher la demande et la réponse pour cette opération

Didacticiel 9 : examen et test d'une base de données

Dans ce didacticiel, vous examinez et testez une table de base de données qui fait partie de l'application Web dans le didacticiel 5.

Vous utilisez l'étape SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC) pour interagir avec une base de données dans un scénario de test et pour tester la réponse à l'aide d'une assertion. Vous examinez la table Users (Utilisateurs) à partir d'une base de données Derby qui fait partie de l'application.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Utiliser l'étape SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC)
- Stocker les propriétés d'application dans une configuration
- Ajouter et modifier une assertion
- Ajouter un filtre

Conditions préalables

- Vous avez terminé le didacticiel 5.
- DevTest Workstation est ouvert.
- Vous avez accès au serveur de démonstration.

Etape 1 : création d'un scénario de test

Procédez comme suit:

1. Dans le volet Project (Projet), cliquez avec le bouton droit de la souris sur le dossier Tests et sélectionnez Create New Test Case (Créer un scénario de test).
2. Définissez le nom de fichier sur Didacticiel 9.
3. Cliquez sur Save (Enregistrer).

Etape 2 : ajout de propriétés de base de données à la configuration

Stockez les propriétés nécessaires à la connexion à la base de données dans la configuration. Cette pratique DevTest standard est recommandée pour augmenter la portabilité des scénarios de test.

Procédez comme suit:

1. Si project.config n'est pas la configuration active, cliquez avec le bouton droit de la souris sur project.config dans le volet Project (Projet) et sélectionnez Make Active (Activer).
2. Ouvrez la configuration project.config.
3. Ajoutez les propriétés suivantes :

DBDriver

org.apache.derby.jdbc.ClientDriver

DBConnect

jdbc:derby://localhost:1529/lisa-demo-server.db

DBUserID

sa

DBPwd

sa

Properties Editor		
Key	Value	Encrypt
DBDriver	org.apache.derby.jdbc.ClientDriver	<input type="checkbox"/>
DBConnect	jdbc:derby://localhost:1529/lisa-demo-server.db	<input type="checkbox"/>
DBUserID	sa	<input type="checkbox"/>
DBPwd	sa	<input type="checkbox"/>

4. Cliquez sur Save (Enregistrer).

Etape 3 : ajout d'une étape de test SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC)

L'étape de test SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC) vous permet de vous connecter à une base de données à l'aide de JDBC et d'effectuer des requêtes SQL sur la base de données.

Procédez comme suit:

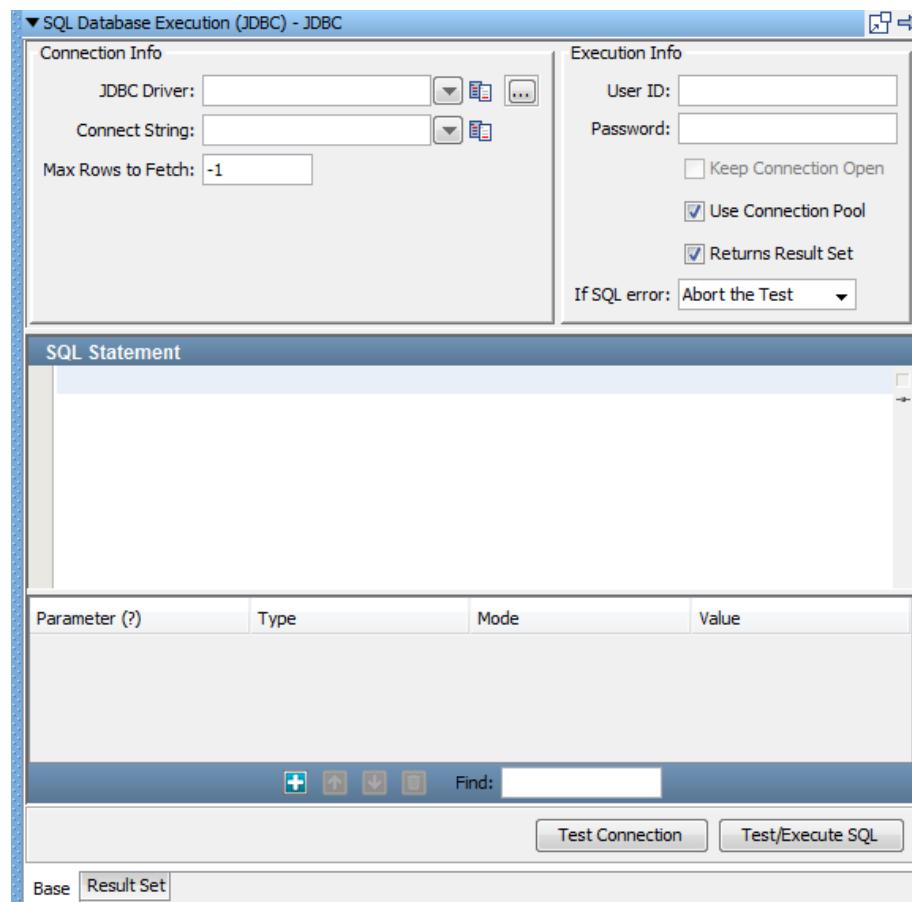
1. Cliquez sur l'onglet Didacticiel 9.



2. Cliquez sur **Add Step (Ajouter une étape)**.
3. Sélectionnez Other Transactions (Autres transactions), puis SQL Database Execution (JDBC).

JDBC est ajouté à l'éditeur de modèles.

4. Pour ouvrir l'éditeur d'étapes, double-cliquez sur l'étape JDBC.



Etape 4 : connexion à la base de données

Pour entrer les informations de connexion, utilisez les propriétés que vous avez ajoutées à la configuration project.config.

Procédez comme suit:

1. Entrez les valeurs suivantes dans les zones Connection Info (Informations sur la connexion) et Execution Info (Informations sur l'exécution) de l'éditeur d'étape. Remarquez que lorsque vous entrez le mot de passe, la valeur est masquée.

JDBC Driver (Pilote JDBC)

`{{DBDriver}}`

Connect String (Chaîne de connexion)

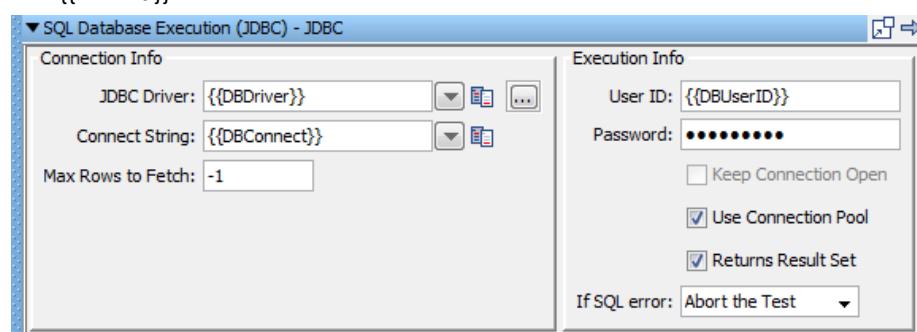
`{{DBConnect}}`

User ID (ID de l'utilisateur)

`{{DBUserID}}`

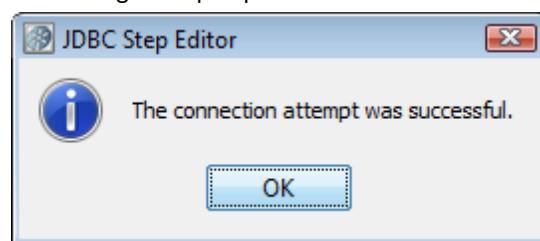
Password (Mot de passe)

`{{DBPwd}}`



2. Cliquez sur Test Connection (Tester la connexion) en bas de l'éditeur d'étapes.

Un message indique que la connexion est valide.



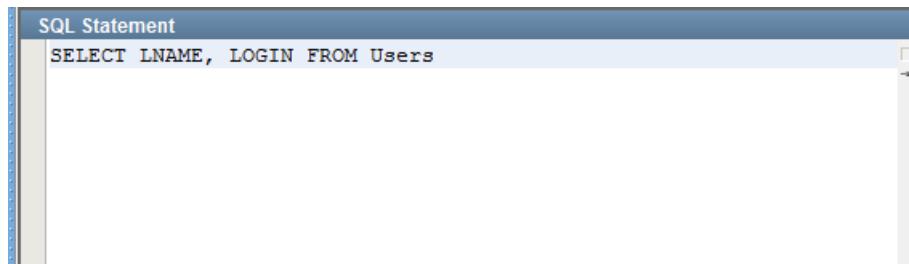
3. Cliquez sur OK (Redéployer/déployer).

Etape 5 : exécution d'une requête SQL

Spécifiez et exécutez une instruction SQL qui récupère les données de la table Users (Utilisateurs).

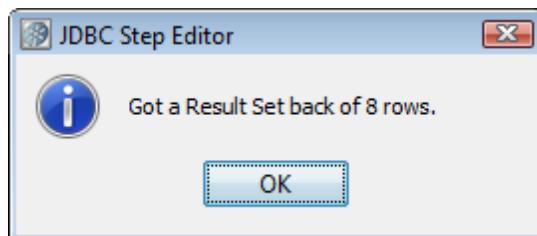
Procédez comme suit:

1. Dans le volet SQL Statement (Instruction SQL), entrez l'instruction suivante :
SELECT LNAME, LOGIN FROM Users



2. Cliquez sur Test/Execute SQL (Tester/exécuter l'expression SQL) en bas de l'éditeur d'étapes.

Un message confirme la validité de la requête et affiche le nombre de lignes renvoyées.



3. Cliquez sur OK.

L'onglet Result Set (Ensemble de résultats) est affiché.

Result Set	
LNAME	LOGIN
	User
Admin	admin
Bellum	sbellum
Piece	wpiece
Reckonwith	areck
Rabbit	boaty
test	itko
simpson	lisa_simpson

Etape 6 : ajout d'une assertion

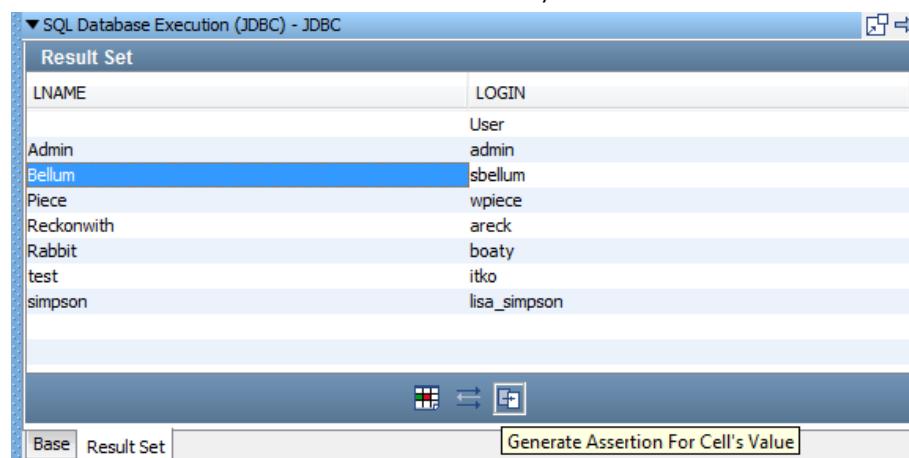
Ajoutez une assertion pour déterminer la présence d'un nom dans l'ensemble de résultats.

Procédez comme suit:

1. Dans l'onglet Result Set (Ensemble de résultats), sélectionnez une cellule de la colonne LNAME.

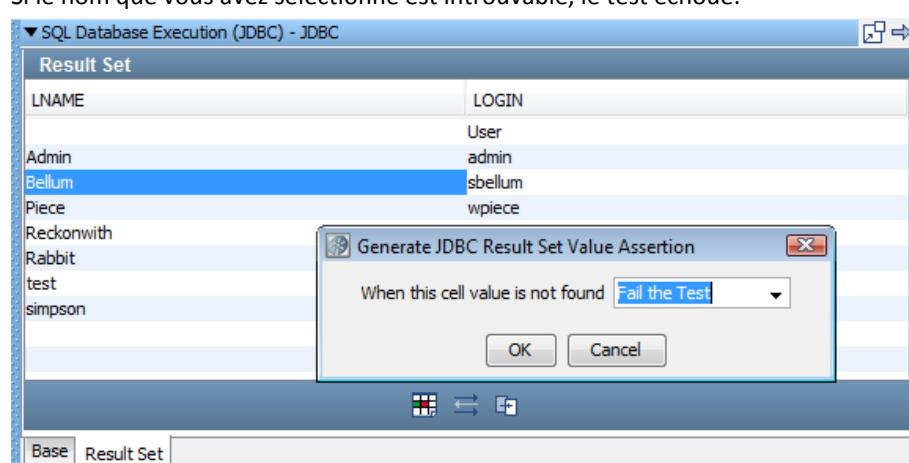
2. Cliquez sur  Generate Assertion for the Value of a Cell (Générer une assertion pour la valeur d'une cellule).

La boîte de dialogue Generate JDBC Result Set Value Assertion (Générer une assertion de valeur d'ensemble de résultats JDBC) s'ouvre.



3. Dans la liste déroulante, sélectionnez l'option Fail the Test (Faire échouer le test).

Si le nom que vous avez sélectionné est introuvable, le test échoue.



4. Cliquez sur OK (Redéployer/déployer).
5. Cliquez sur Save (Enregistrer).

Etape 7 : exécution du scénario de test

Procédez comme suit:



Start Interactive Test Run (Démarrer l'exécution d'un test interactif).

2. Cliquez sur Execute Next Step (Exécuter l'étape suivante).

Le test est exécuté. L'ensemble de résultats est affiché dans l'onglet Response (Réponse).

The screenshot shows the 'Interactive Test Run - 1' window. On the left, there's an 'Execution History' pane with a single entry: 'JDBC'. Below it is a message box stating 'The test is complete.' with an 'OK' button. On the right, there's a 'Result Set' table with two columns: 'LNAME' and 'LOGIN'. The data rows are:

LNAME	LOGIN
Admin	User
Bellum	admin
Piece	sbellum
Reckonwith	wpiece
Rabbit	areck
test	boaty
simpson	itko
	lisa_simpson

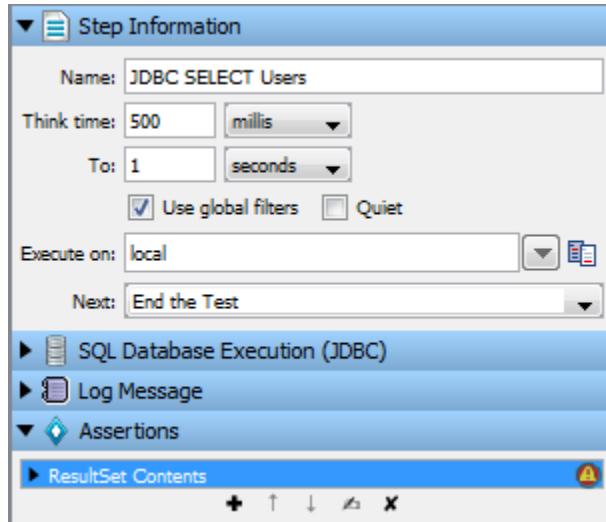
3. Rétractez la barre d'état de l'ITR.

Etape 8 : modification de l'assertion

Modifiez l'assertion de sorte à faire échouer le test.

Procédez comme suit:

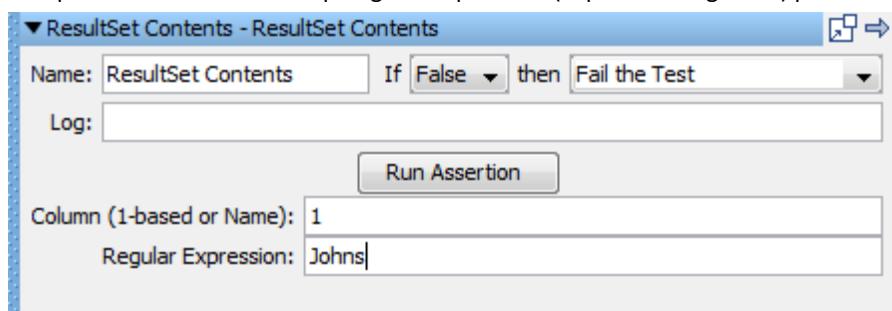
1. Dans l'éditeur de modèles, cliquez sur l'étape de test JDBC Select Users (Sélection d'utilisateurs JDBC).
2. Ouvrez l'onglet Assertions dans l'arborescence Element.



3. Double-cliquez sur l'assertion que vous avez créée antérieurement.

L'éditeur d'assertions est ouvert. La partie inférieure indique que l'assertion vérifie la première colonne de l'ensemble de résultats pour la valeur spécifiée.

4. Remplacez la valeur du champ Regular Expression (Expression régulière) par Johns.



5. Démarrez une nouvelle exécution d'un test interactif et exécutez le scénario de test à nouveau.

Le test échoue.

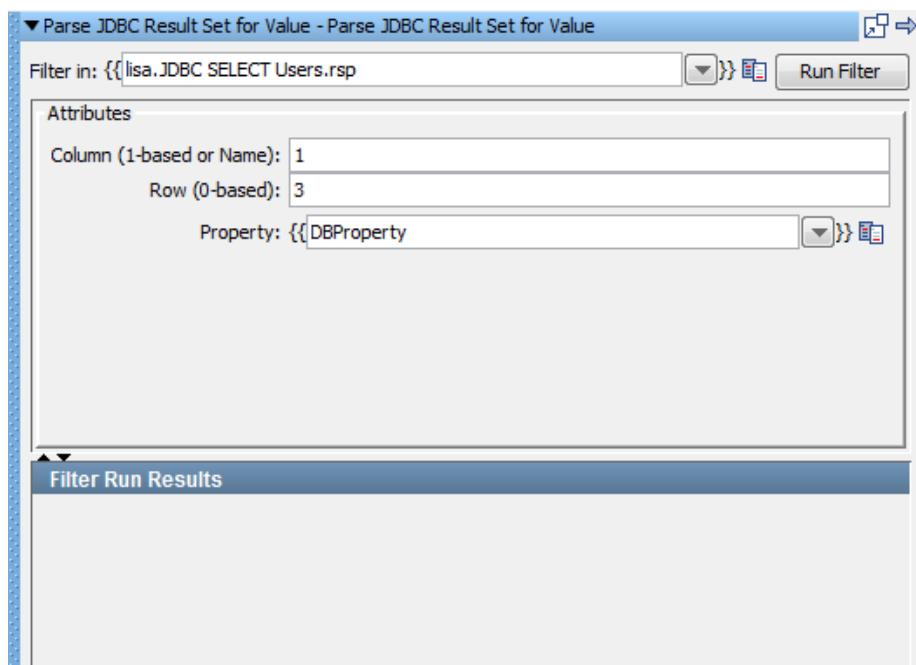
6. Rétractez la barre d'état de l'ITR.

Etape 9 : ajout d'un filtre

Ajoutez un filtre de base de données pour capturer la valeur située dans la première colonne et la quatrième ligne de l'ensemble de résultats. La valeur est stockée dans une propriété.

Procédez comme suit:

1. Dans l'éditeur de modèles, sélectionnez l'étape de test JDBC Select Users (Sélection d'utilisateurs JDBC).
2. Ouvrez l'onglet Filters (Filtres) dans l'arborescence Element.
3. Cliquez sur  Add (Ajouter).
4. A partir du sous-menu Database Filters (Filtres de base de données), sélectionnez Extract Value from JDBC Result Set (Extraire la valeur de l'ensemble de résultats JDBC).
L'éditeur de filtres s'ouvre.
5. Dans le champ Column (Colonne), entrez 1. Vous pouvez également entrer le nom de colonne réel, *LNAME*.
6. Dans le champ Row (Ligne), entrez 3.
Ce champ est en base 0. Par conséquent, la valeur 3 correspond à la quatrième ligne.
7. Dans le champ Property (Propriété), saisissez *DBProperty*.



8. Cliquez sur Save (Enregistrer).

Etape 10 : test du filtre et de l'assertion

Procédez comme suit:

- Démarrez une nouvelle exécution d'un test interactif et exécutez le scénario de test à nouveau.

Le test échoue, car Johns est introuvable dans l'ensemble de résultats.

- Cliquez sur l'onglet Test Events (Événements de test).

- Cliquez sur l'événement Property Set (Propriété définie).

Remarquez que DBProperty a été défini sur la valeur spécifiée par le filtre.

- Cliquez sur l'événement Assertion fired (Assertion déclenchée).

La zone Long Info Field (Informations détaillées) indique que l'assertion s'est déclenchée, car la première colonne de l'ensemble de résultats ne contient pas la valeur Johns.

Timestamp	EventID	Short	Long
14:44:58,293	Step started	JDBC SELECT...	
14:44:58,294	Step target	JDBC SELECT... jdbc:derby://localhost:1529/lisa-demo-server.db	
14:44:58,294	Step request	JDBC SELECT... SELECT LNAME, LOGIN FROM Users	
14:44:58,297	Step response...	JDBC SELECT... 3	
14:44:58,299	Step response	JDBC SELECT... <com.itko.lisa.jdbc.JDBCResultSetCache> <row...>	
14:44:58,300	Property set	DBProperty Piece	
14:44:58,300	Assertion fired	JDBC SELECT... Assert [ResultSet Contents] fired false of type R...	

Long Info Field

```

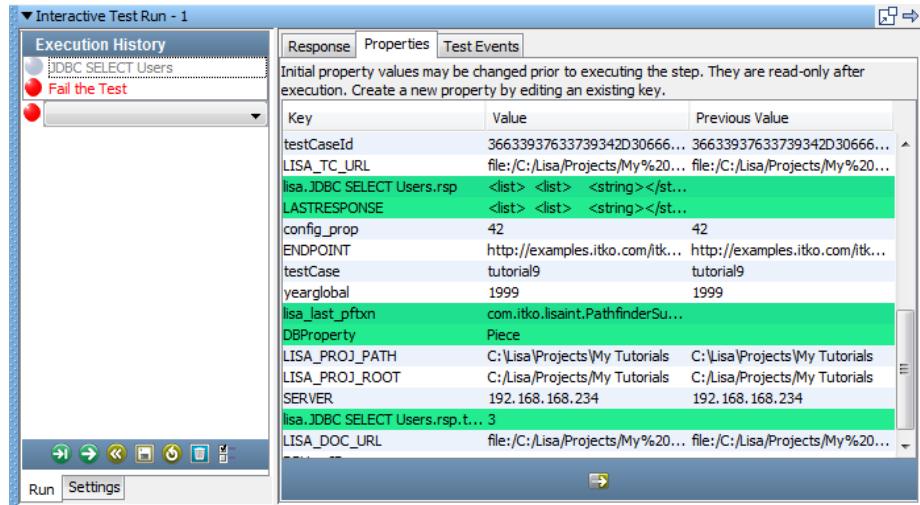
Assert [ResultSet Contents] fired false of type ResultSet
Result set does not contain Johns in column 1 result
<list>
  <list>
    <string></string>
    <string>User</string>
  </list>
</list>

```

- Cliquez sur l'onglet Properties (Propriétés).

- Localisez et vérifiez la ligne DBProperty.

Figure 3: Capture d'écran des résultats ITR Results (Résultats de l'ITR) pour le didacticiel 9



Didacticiel 9 : résumé

Dans ce didacticiel, vous avez créé un scénario de test pour interroger une base de données. Vous avez utilisé la table Users (Utilisateurs) de la base de données Apache Derby qui accompagne les applications sur le serveur de démonstration. Vous avez appris à :

- Vous connecter à la base de données
- Exécuter une requête SQL sur la base de données
- Ajouter des assertions et des filtres

Didacticiel 10 : simulation d'un test rapide

Dans ce didacticiel, vous utilisez l'option de simulation rapide (test rapide) pour apprendre à simuler des tests et à interpréter les rapports qui en découlent. Vous exécutez le test rapide sur l'exemple de test à plusieurs niveaux fourni avec DevTest. Exécuter un test rapide est la façon la plus simple de simuler un test.

Tâches de didacticiel

Ce didacticiel vous permet de :

- Utiliser un scénario de test à plusieurs niveaux
- Utiliser la fonctionnalité de test rapide
- Sélectionner et formater des rapports

Conditions préalables

- Vous avez terminé les didacticiels 5 à 9.
- DevTest Workstation est ouvert.
- Vous avez accès au serveur de démonstration.

Etape 1 : ouverture du scénario de test

Ouvrez un scénario de test à partir du projet Examples (Exemples).

Procédez comme suit:

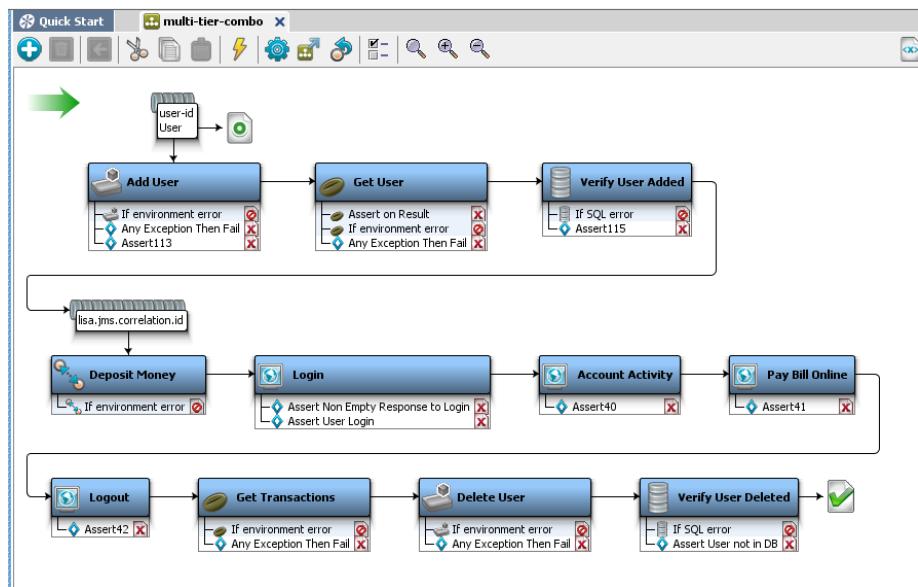
1. Dans le menu principal, sélectionnez File (Fichier), Open (Ouvrir), Test Case (Scénario de test), File System (Système de fichiers).
2. Accédez au dossier LISA_HOME\examples\Tests.
3. Sélectionnez le test à plusieurs niveaux et cliquez sur Open (Ouvrir).

Le scénario de test à plusieurs niveaux s'ouvre dans l'éditeur de modèles.

Etape 2 : vérification du scénario de test

Examinez les divers types d'étapes de test dans ce scénario de test. Par exemple :

- L'étape Add User (Ajouter un utilisateur) est une étape Web Service Execution (Legacy) (Exécution de service Web (héritée)).
- L'étape Get User (Obtenir le nom d'utilisateur) est une étape Enterprise JavaBean Execution (Exécution d'un objet EJB).
- L'étape Verify User (Vérifier l'utilisateur) est une étape SQL Database Execution (JDBC) (Exécution d'une base de données SQL JDBC).
- L'étape Deposit Money (Déposer de l'argent) est une étape JMS Messaging (JNDI) (Messagerie JMS (JNDI)).



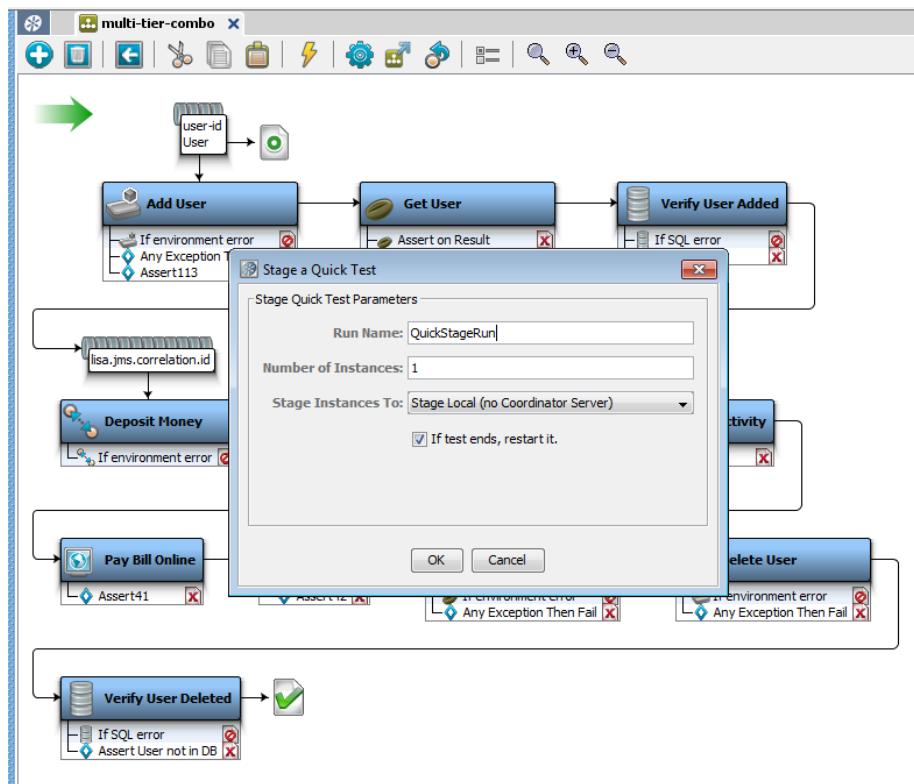
La plupart de ces étapes sont utilisées dans les didacticiels 6 à 9. Dans ce didacticiel, vous utilisez toutes les étapes de test pour créer un scénario de test plus réaliste impliquant plusieurs couches de l'application.

Etape 2, partie A : exécution d'un test rapide

Procédez comme suit:

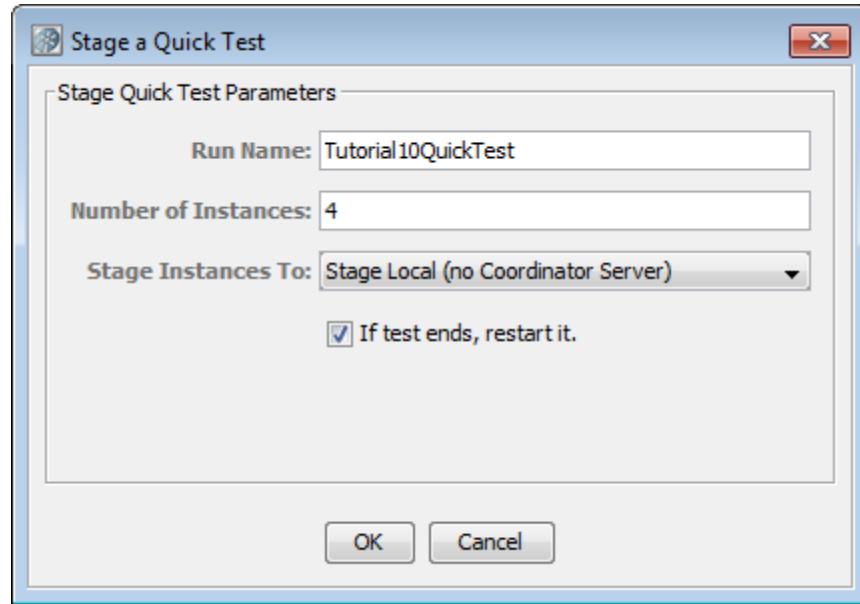
- Dans la barre d'outils du scénario de test, cliquez sur  Stage a Quick Test (Simuler un test rapide).

Pour simuler un test rapide, l'exemple de scénario de test peut être ouvert dans l'éditeur de modèles. Vous pouvez également cliquer avec le bouton droit de la souris sur le test dans le panneau Project (Projet) et y entrer les paramètres pour simuler un test rapide.



- Dans la boîte de dialogue Stage a Quick Test, saisissez les informations requises, comme suit :

- Run Name (Nom de l'exécution) : entrez un nom unique (Tutorial10QuickTest).
- Number of Instances (Nombre d'instances) : entrez le nombre d'utilisateurs qui exécutent le test simultanément (4).
- Stage Instances To (Simuler les instances sur) : sélectionnez le nom du serveur de coordination ou simulez-les localement.
- Pour redémarrer le test, sélectionnez If test ends, restart it (Si le test se termine, redémarrez-le).



- Cliquez sur OK.

La fenêtre Test Run (Exécution de test) s'ouvre, mais le test n'a pas encore démarré.

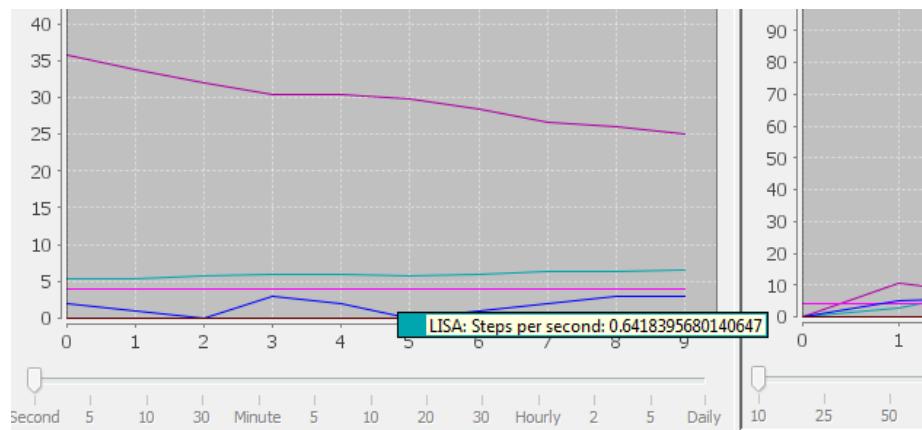


- Pour lancer l'exécution du test, cliquez sur à partir de la barre d'outils principale.

Le test commence et le graphique affiche immédiatement les résultats.



Vous pouvez passer la souris sur les lignes de graphique pour afficher des descriptions.



5. Pour sélectionner les événements à afficher, sélectionnez l'onglet Events (Événements).

multi-tier-combo [Tutorial10QuickTest] ×					
Events to Filter Out		Test Events			
Terse Event Set		Timestamp	Event	Simulator	Instance
<input checked="" type="checkbox"/>	Coordinator server started	2011-08-24 08:30:14,208	Cycle ending	local	3/4
<input checked="" type="checkbox"/>	Coordinator server ended	2011-08-24 08:30:14,205	Cycle started	local	3/4
<input checked="" type="checkbox"/>	Coordinator started	2011-08-24 08:30:14,204	Cycle ending	local	3/3
<input checked="" type="checkbox"/>	Coordinator ended	2011-08-24 08:30:14,200	Cycle started	local	3/3
<input type="checkbox"/>	Test started	2011-08-24 08:30:08,186	Cycle ending	local	3/2
<input type="checkbox"/>	Test ended	2011-08-24 08:30:08,802	Cycle ending	local	0/2
<input type="checkbox"/>	Instance started	2011-08-24 08:29:58,686	Cycle started	local	1/3
<input checked="" type="checkbox"/>	Instance ended	2011-08-24 08:29:58,669	Cycle ending	local	1/2
<input checked="" type="checkbox"/>	Simulator started	2011-08-24 08:29:49,118	Cycle started	local	2/3
<input checked="" type="checkbox"/>	Simulator ended	2011-08-24 08:29:49,408	Cycle ending	local	2/2
<input checked="" type="checkbox"/>	Cycle initialized	2011-08-24 08:29:11,145	Cycle started	local	1/2
<input type="checkbox"/>	Cycle started	2011-08-24 08:29:21,106	Cycle ending	local	1/1
<input checked="" type="checkbox"/>	Cycle ended normally	2011-08-24 08:29:19,388	Cycle started	local	0/2
<input type="checkbox"/>	Cycle ended abnormally	2011-08-24 08:29:19,381	Cycle ending	local	0/1
<input type="checkbox"/>	Cycle initialized	2011-08-24 08:28:18,443	Cycle started	local	3/2
<input type="checkbox"/>	Cycle started	2011-08-24 08:28:18,627	Cycle ending	local	3/1
<input type="checkbox"/>	Cycle ended normally	2011-08-24 08:28:17,36	Cycle started	local	2/2
<input type="checkbox"/>	Cycle ended abnormally	2011-08-24 08:28:10,28	Cycle ending	local	2/1
<input type="checkbox"/>	Cycle ended normally	2011-08-24 08:28:1,687	Cycle started	local	1/1
<input type="checkbox"/>	Cycle ended abnormally	2011-08-24 08:28:1,676	Cycle ending	local	1/0
<input type="checkbox"/>	Cycle initialized	2011-08-24 08:28:30,525	Cycle started	local	0/1
<input type="checkbox"/>	Cycle started	2011-08-24 08:28:30,518	Cycle ending	local	0/0
<input type="checkbox"/>	Cycle ended normally	2011-08-24 08:28:27,594	Cycle started	local	3/1
<input type="checkbox"/>	Cycle ended abnormally	2011-08-24 08:28:27,565	Cycle ending	local	3/0
<input type="checkbox"/>	Cycle initialized	2011-08-24 08:28:26,180	Cycle started	local	2/1
<input type="checkbox"/>	Cycle started	2011-08-24 08:28:26,167	Cycle ending	local	2/0
<input type="checkbox"/>	Cycle ended normally	2011-08-24 08:27:19,731	Cycle started	local	2/0
<input type="checkbox"/>	Cycle ended abnormally	2011-08-24 08:27:19,729	Cycle started	local	0/0
<input type="checkbox"/>	Cycle initialized	2011-08-24 08:27:19,730	Cycle ending	local	1/0
<input type="checkbox"/>	Cycle started	2011-08-24 08:27:19,729	Cycle started	local	3/0
<input type="checkbox"/>	Cycle ended normally	2011-08-24 08:27:19,706	Test started	N/A	0/0

Simulators

Name	Instances	Change
local	4	

Add Simulator

Perf Stats Events

Auto Refresh

Wednesday, August 24, 2011

Etape 2, partie B : affichage des rapports générés

DevTest fournit une visionneuse de rapports permettant d'afficher les rapports.

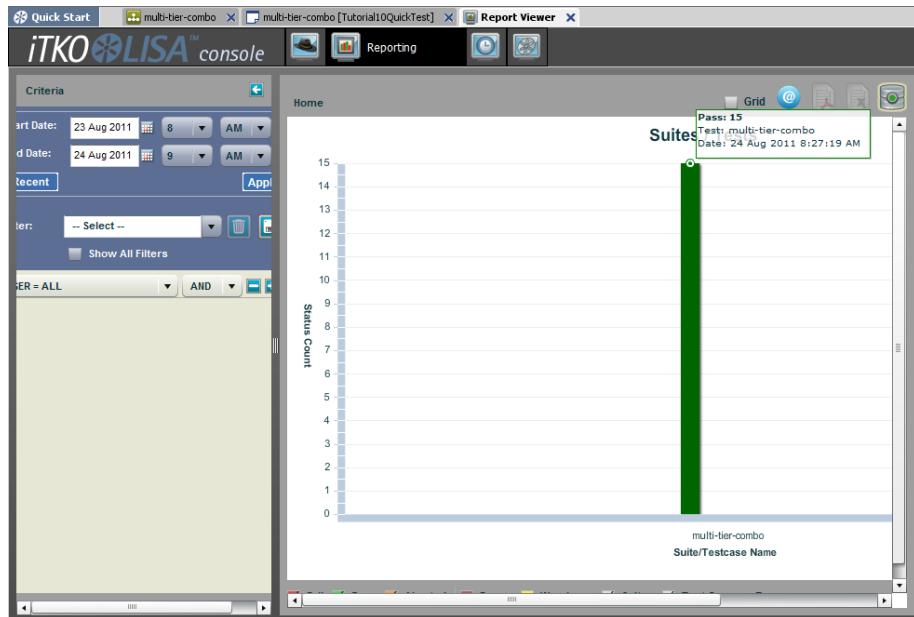
Procédez comme suit:

1. Dans le menu principal, cliquez sur View (Afficher), Reporting Console (Console de



génération de rapports) ou cliquez sur Reports (Rapports)  dans la barre d'outils.

La visionneuse Report Viewer (Visionneuse de rapports) s'ouvre.



2. Pour afficher uniquement les activités récentes, cliquez sur Recent (Récent).

Dans l'illustration précédente, le graphique indique que tous les tests du scénario de test sont terminés.

Vous pouvez cliquer avec le bouton droit de la souris sur le graphique pour afficher des menus différents. Pour plus d'informations sur les rapports, consultez la section Rapports de la rubrique *Utilisation de CA Application Test*.

Didacticiel 10 : résumé

Dans ce didacticiel, vous avez testé l'exemple de test à plusieurs niveaux à l'aide d'un test rapide. Vous avez appris à :

- Parcourir un scénario de test contenant plusieurs types d'étapes de test
- Configurer et exécuter un test dans la fonctionnalité de test rapide
- Examiner un rapport généré à partir de l'exécution de test.

Chapitre 3: Didacticiels sur les tests d'application mobile

Cette section contient une série de didacticiels qui illustrent divers aspects des tests d'application mobile.

Configuration requise pour chaque didacticiel

- Vous avez installé DevTest Workstation et entré vos informations de licence pour DevTest Solutions.
- Vous avez effectué toutes les tâches décrites dans la section Configuration de l'environnement de tests d'application mobile.
- Pour le *didacticiel 1 sur l'enregistrement d'un scénario de test iOS*, téléchargez l'application de didacticiel [UICatalog.app.zip](#) sur votre ordinateur local et décompressez-le.
- Pour le *didacticiel 2 sur l'enregistrement d'un scénario de test Android*, téléchargez l'application de didacticiel [ApiDemos.apk](#) sur votre ordinateur local.

Ce chapitre traite des sujets suivants :

[Didacticiel 1 : enregistrement d'un scénario de test iOS](#) (page 117)

[Didacticiel 2 : enregistrement d'un scénario de test d'application Android](#) (page 124)

Didacticiel 1 : enregistrement d'un scénario de test iOS

Ce didacticiel présente chaque étape requise pour l'enregistrement d'un scénario de test à l'aide d'un simulateur iOS.

Dans ce didacticiel, vous effectuez les tâches suivantes :

- Création d'un actif
- Enregistrement d'un scénario de test d'application mobile
- Exécution du scénario de test dans l'utilitaire d'exécution d'un test interactif

Etape 1 : démarrage de DevTest Workstation

Procédez comme suit:

1. Démarrez le registre.
 - Pour DevTest Server :
 - a. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, Registre (Registre).
 - b. Patientez jusqu'à ce que le message DevTest Registry Ready (Le registre DevTest est prêt) s'affiche.
 - Pour DevTest Workstation, utilisez un registre qui s'exécute sur un autre ordinateur.
2. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, DevTest Workstation.

La boîte de dialogue Set DevTest Registry (Définir le registre DevTest) s'ouvre.
 3. Sélectionnez un registre, puis cliquez sur OK.

Etape 2 : création d'un projet

Le projet que vous créez contient tous les exemples de fichier de scénario de test qui sont requis pour les didacticiels mobiles.

Remarque : Si vous avez créé le projet MyMobileTutorials dans un didacticiel précédent, passez à l'étape 3 sur la [création d'un fichier de configuration pour un simulateur Android](#) (page 125).

Procédez comme suit:

1. A partir du menu principal de DevTest Workstation, sélectionnez File (Fichier), New (Créer), Project (Projet).

La boîte de dialogue Create New Project (Créer un projet) s'affiche.
2. Dans le champ Project Name (Nom du projet), saisissez **MyMobileTutorials**.
3. Acceptez le paramètre par défaut pour créer le projet dans le répertoire LISA_HOME.
4. Cliquez sur Create (Créer).

Le projet MyMobileTutorials a été créé.

Etape 3 : création d'un fichier de configuration pour un simulateur iOS

La configuration par défaut est appelée project.config et est créée automatiquement pour un nouveau projet. Le fichier project.config se trouve dans le dossier Configs (Configurations) du panneau Project (Projet). Vous pouvez créer un fichier de configuration contenant des informations sur l'actif spécifique à l'utilisation d'un simulateur iOS.

Procédez comme suit:

1. Cliquez avec le bouton droit de la souris sur le dossier Configs (Configurations) dans le panneau Project (Projet) et sélectionnez Create New Config (Créer une configuration).
2. Saisissez le nom de la nouvelle configuration : **MobileiOSimulator**.
3. Cliquez sur OK (Redéployer/déployer).
Le fichier de configuration MobileiOSimulator s'ouvre dans l'éditeur de propriétés.
4. Cliquez avec le bouton droit de la souris sur le fichier de configuration MobileiOSimulator dans le panneau Project (Projet) et sélectionnez Make Active (Activer).

Etape 4 : création d'un actif de simulateur iOS

Un actif de session de simulateur associé à une configuration vous permet de spécifier le simulateur mobile à utiliser pour les tests.

Procédez comme suit:

1. Ouvrez le fichier de configuration MobileiOSSimulator que vous avez créé à l'étape 3 ([création d'un fichier de configuration pour un simulateur iOS](#)) (page 119), si ce n'est pas déjà fait.

2. Dans le navigateur d'actifs, cliquez sur Add (Ajouter)  au bas du volet.
3. Cliquez sur Mobile Session (Session mobile).

La boîte de dialogue (Session mobile) s'affiche.

Remarque : Si vous avez déjà défini des valeurs pour l'un de ces champs, vous pouvez sélectionner la valeur dans la liste pour ce champ. Pour ajouter ou supprimer des valeurs de la liste, sélectionnez l'option Maintain List (Gérer la liste).

4. Remplissez les champs suivants :

nom

Saisissez **iOSSimulator**.

Description

Entrez **iOS Simulator for use with Mobile Tutorial 1** (Simulateur iOS à utiliser avec le didacticiel mobile 1).

Plate-forme

Sélectionnez **iOS** dans le menu déroulant.

Application

Accédez au fichier **UICatalog.app** téléchargé et sélectionnez-le.

Family (Famille)

Sélectionnez **iPhone only** (iPhone uniquement) dans le menu déroulant.

Target (Cible)

Sélectionnez **Simulator** (Simulateur) dans le menu déroulant.

Simulator (Simulateur)

Accédez au simulateur iOS sur votre ordinateur et sélectionnez-le.

Version iOS

Sélectionnez 6.1.

5. (Facultatif) Pour vérifier l'actif avant de l'enregistrer, cliquez sur .
6. Cliquez sur OK.

Le nouvel actif de simulateur iOS s'affiche dans le navigateur d'actifs.

7. Cliquez sur Enregistrer.

Etape 5 : enregistrement d'un scénario de test

Procédez comme suit:

1. Vérifiez que le fichier de configuration MobileiOSimulator est actif.
2. Créez un scénario de test nommé **Tutorial_iOS_Simulator** :
 - a. Cliquez avec le bouton droit de la souris sur le dossier Tests dans le panneau Project (Projet), puis cliquez sur Create New Test Case (Créer un scénario de test).
 - b. Dans la boîte de dialogue, accédez au répertoire dans lequel vous voulez stocker le scénario de test.
 - c. Saisissez le nom du nouveau scénario de test.
 - d. Cliquez sur Save (Enregistrer).
 - e. Cliquez sur Create steps by recording or templating (Créer des étapes à l'aide d'un enregistrement ou d'un modèle) .
3. Cliquez sur Record Test Case for User Interface (Enregistrer le scénario de test pour l'interface utilisateur), Mobile Recorder (Enregistreur mobile).
L'enregistreur de test d'application mobile et la fenêtre de simulateur mobile s'ouvrent.
4. Si vous avez défini plusieurs actifs mobiles, sélectionnez l'un d'eux à connecter dans la liste déroulante Choose Mobile asset (Sélectionner un actif mobile), puis cliquez sur OK.
5. Cliquez sur Start Recording (Lancer l'enregistrement) au bas de la fenêtre d'enregistreur.
L'enregistrement est en cours.
6. Effectuez les actions à enregistrer pour votre scénario de test dans la fenêtre de l'enregistreur.
Remarque : N'effectuez pas ces actions directement dans le simulateur mobile. Toutes les actions effectuées dans la fenêtre de l'enregistreur sont envoyées au simulateur automatiquement.
7. Cliquez sur Stop Recording lorsque vous avez capturé toutes les actions pour votre test.
La fenêtre d'enregistreur et le simulateur mobile se ferment. Le nouveau scénario de test est rempli avec des étapes de test qui représentent les actions d'applications mobiles capturées pendant l'enregistrement. Chaque étape de test représente les actions effectuées avec une fenêtre spécifique dans l'application.
8. Pour afficher les détails de chaque étape :
 - a. Cliquez sur l'étape de test à réviser.

- b. Cliquez sur l'étape Mobile testing step (Etape de test d'application mobile) dans l'arborescence d'éléments à droite pour afficher les détails de l'étape.

L'onglet Mobile Testing Step (Etape de test d'application mobile) s'ouvre et affiche une capture d'écran de l'application de test. La section Actions, en haut de l'onglet, indique les actions effectuées dans l'étape de test.

- c. Pour afficher la capture d'écran associée à une action, cliquez sur l'action dans la section Actions.

Pour plus d'informations sur la modification d'une étape de test enregistrée, consultez la rubrique Modification d'étapes de test d'applications mobiles.

Pour plus d'informations sur l'ajout d'assertions à une étape de test, consultez la rubrique Ajout d'une assertion à une étape de test d'applications mobiles.

Etape 6 : exécution du scénario de test dans l'ITR

Procédez comme suit:

1. Ouvrez le scénario de test de l'étape 5 ([enregistrement d'un scénario de test](#) (page 122), si ce n'est pas déjà fait.



2. Dans la barre d'outils, cliquez sur l'icône ITR .

Remarque : Indiquez si démarrer une nouvelle exécution d'un test interactif ou ouvrir une exécution d'un test interactif précédente (si vous avez exécuté l'ITR précédemment).

La fenêtre Interactive Test Run (Exécution d'un test interactif) s'ouvre.



3. Cliquez sur Execute (Exécuter) au bas de la fenêtre de l'ITR.

La fenêtre de simulateur mobile s'ouvre et DevTest exécute les étapes de test. L'application mobile s'affiche dans le simulateur lors de l'exécution du test.

Lorsque le test est terminé, un message s'affiche informant de la fin du test. La fenêtre du simulateur se ferme.

4. Cliquez sur OK.

Pour plus d'informations sur l'utilisation de l'utilitaire d'exécution d'un test interactif, consultez la rubrique Exécution de scénarios de test et de suites.

Didacticiel 2 : enregistrement d'un scénario de test d'application Android

Ce didacticiel présente chaque étape requise pour enregistrer un scénario de test à l'aide d'un simulateur Android.

Dans ce didacticiel, vous effectuez les tâches suivantes :

- Création d'un actif
- Enregistrement d'un scénario de test d'application mobile
- Exécution du scénario de test dans l'utilitaire d'exécution d'un test interactif

Etape 1 : démarrage de DevTest Workstation

Procédez comme suit:

1. Démarrez le registre.
 - Pour DevTest Server :
 - a. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, Registre (Registre).
 - b. Patiencez jusqu'à ce que le message DevTest Registry Ready (Le registre DevTest est prêt) s'affiche.
 - Pour DevTest Workstation, utilisez un registre qui s'exécute sur un autre ordinateur.
2. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, DevTest Workstation.

La boîte de dialogue Set DevTest Registry (Définir le registre DevTest) s'ouvre.
 3. Sélectionnez un registre, puis cliquez sur OK.

Etape 2 : création d'un projet

Le projet que vous créez contient tous les exemples de fichier de scénario de test qui sont requis pour les didacticiels mobiles.

Remarque : Si vous avez créé le projet MyMobileTutorials dans un didacticiel précédent, passez à l'étape 3 sur la [création d'un fichier de configuration pour un simulateur Android](#) (page 125).

Procédez comme suit:

1. A partir du menu principal de DevTest Workstation, sélectionnez File (Fichier), New (Créer), Project (Projet).
La boîte de dialogue Create New Project (Créer un projet) s'affiche.
2. Dans le champ Project Name (Nom du projet), saisissez **MyMobileTutorials**.
3. Acceptez le paramètre par défaut pour créer le projet dans le répertoire LISA_HOME.
4. Cliquez sur Create (Créer).
Le projet MyMobileTutorials a été créé.

Etape 3 : création d'un fichier de configuration pour un simulateur Android

La configuration par défaut est appelée project.config et est créée automatiquement pour un nouveau projet. Le fichier project.config se trouve dans le dossier Configs (Configurations) du panneau Project (Projet). Vous pouvez créer un fichier de configuration contenant des informations sur l'actif spécifique à l'utilisation d'un simulateur iOS.

Procédez comme suit:

1. Cliquez avec le bouton droit de la souris sur le dossier Configs (Configurations) dans le panneau Project (Projet) et sélectionnez Create New Config (Créer une configuration).
2. Saisissez le nom de la nouvelle configuration : **MobileAndroidSimulator**.
3. Cliquez sur OK (Redéployer/déployer).
Le fichier de configuration MobileAndroidSimulator s'ouvre dans l'éditeur de propriétés.
4. Cliquez avec le bouton droit de la souris sur le fichier de configuration MobileAndroidSimulator dans le panneau Project (Projet) et sélectionnez Make Active (Activer).

Etape 4 : création d'un actif d'émulateur Android

Un actif de session de simulateur associé à une configuration vous permet de spécifier le simulateur mobile à utiliser pour les tests.

Procédez comme suit:

1. Ouvrez le fichier de configuration MobileAndroidSimulator que vous avez créé à l'étape 3 ([création d'un fichier de configuration pour un simulateur Android](#)) (page 125), si ce n'est pas déjà fait.
2. Dans le navigateur d'actifs, cliquez sur Add (Ajouter)  au bas du volet.
3. Cliquez sur Mobile Session (Session mobile).
La boîte de dialogue (Session mobile) s'affiche.
4. Renseignez les champs suivants :

nom

Entrez **AndroidSimulator**.

Description

Entrez **Android Simulator for use with Mobile Tutorial 2** (Simulateur Android utilisé pour le didacticiel mobile 2).

Plate-forme

Sélectionnez **Android** dans le menu déroulant.

Application

Accédez au fichier **ApiDemos.apk** téléchargé et sélectionnez-le.

Target (Cible)

Sélectionnez **Emulator** (Emulateur) dans le menu déroulant.

AVD (Appareil virtuel Android)

Saisissez le nom de l'appareil virtuel Android défini lors de la configuration de test d'application mobile.

SDK Version (Version SAML)

Sélectionnez **4.2.2**.

5. Cliquez sur  pour vérifier l'actif.

6. Cliquez sur OK.

Le nouvel actif de simulateur Android s'affiche dans le navigateur d'actifs.

7. Cliquez sur Enregistrer.

Etape 5 : enregistrement d'un scénario de test

Procédez comme suit:

1. Vérifiez que le fichier de configuration MobileAndroidSimulator est actif.
2. Créez un scénario de test nommé **Tutorial_Android_Simulator**.
 - a. Cliquez avec le bouton droit de la souris sur le dossier Tests dans le panneau Project (Projet), puis cliquez sur Create New Test Case (Créer un scénario de test).
 - b. Dans la boîte de dialogue, accédez au répertoire dans lequel vous voulez stocker le scénario de test.
 - c. Saisissez le nom du nouveau scénario de test.
 - d. Cliquez sur Save (Enregistrer).
3. Cliquez sur Create steps by recording or templating (Créer des étapes à l'aide d'un enregistrement ou d'un modèle) .
4. Cliquez sur Record Test Case for User Interface (Enregistrer le scénario de test pour l'interface utilisateur), Mobile Recorder (Enregistreur mobile).
L'enregistreur de test d'application mobile et la fenêtre de simulateur mobile s'ouvrent.
5. Si vous avez défini plusieurs actifs mobiles, sélectionnez l'un d'eux à connecter dans la liste déroulante Choose Mobile asset (Sélectionner un actif mobile), puis cliquez sur OK.
6. Cliquez sur Start Recording (Lancer l'enregistrement) au bas de la fenêtre d'enregistreur.
L'enregistrement est en cours.
7. Effectuez les actions à enregistrer pour votre scénario de test dans la fenêtre de l'enregistreur.
Remarque : N'effectuez pas ces actions directement dans le simulateur mobile. Toutes les actions effectuées dans la fenêtre de l'enregistreur sont envoyées au simulateur automatiquement.
8. Cliquez sur Stop Recording lorsque vous avez capturé toutes les actions pour votre test.
La fenêtre d'enregistreur et le simulateur mobile se ferment. Le nouveau scénario de test est rempli avec des étapes de test qui représentent les actions d'applications mobiles capturées pendant l'enregistrement. Chaque étape de test représente les actions effectuées avec une fenêtre spécifique dans l'application.
9. Pour afficher les détails de chaque étape :
 - a. Cliquez sur l'étape de test à réviser.

- b. Cliquez sur l'étape Mobile testing step (Etape de test d'application mobile) dans l'arborescence d'éléments à droite pour afficher les détails de l'étape.

L'onglet Mobile Testing Step (Etape de test d'application mobile) s'ouvre et affiche une capture d'écran de l'application de test. La section Actions, en haut de l'onglet, indique les actions effectuées dans l'étape de test.

- c. Pour afficher la capture d'écran associée à une action, cliquez sur l'action dans la section Actions.

Pour plus d'informations sur la modification d'une étape de test enregistrée, consultez la rubrique Modification d'étapes de test d'applications mobiles.

Pour plus d'informations sur l'ajout d'assertions à une étape de test, consultez la rubrique Ajout d'une assertion à une étape de test d'applications mobiles.

Etape 6 : exécution du scénario de test dans l'ITR

Procédez comme suit:

1. Ouvrez le scénario de test de l'étape 5 ([enregistrement d'un scénario de test](#) (page 122), si ce n'est pas déjà fait.



2. Dans la barre d'outils, cliquez sur l'icône ITR .

Remarque : Indiquez si démarrer une nouvelle exécution d'un test interactif ou ouvrir une exécution d'un test interactif précédente (si vous avez exécuté l'ITR précédemment).

La fenêtre Interactive Test Run (Exécution d'un test interactif) s'ouvre.



3. Cliquez sur Execute (Exécuter) au bas de la fenêtre de l'ITR.

La fenêtre de simulateur mobile s'ouvre et DevTest exécute les étapes de test. L'application mobile s'affiche dans le simulateur lors de l'exécution du test.

Lorsque le test est terminé, un message s'affiche informant de la fin du test. La fenêtre du simulateur se ferme.

4. Cliquez sur OK.

Pour plus d'informations sur l'utilisation de l'utilitaire d'exécution d'un test interactif, consultez la rubrique Exécution de scénarios de test et de suites.

Chapitre 4: Didacticiel sur CA Service Virtualization

Ce didacticiel vous permet de :

- Créer et d'activer un fichier de configuration
- Configurer l'enregistreur de VSE
- Enregistrement d'un scénario de test
- Déployer un modèle de service virtuel
- Tester au moyen d'un modèle de service virtuel

Ce chapitre traite des sujets suivants :

[Conditions préalables](#) (page 129)

[Procédure de création et de test d'une image de service virtuel](#) (page 130)

Conditions préalables

Les étapes suivantes représentent des conditions préalables au didacticiel :

- La station de travail DevTest Workstation et l'environnement VSE sont installés.
- Vous devez consulter les rubriques suivantes :
 - [Glossaire](#) (page 149)
 - Présentation de la virtualisation
 - Introduction à CA Service Virtualization

Procédure de création et de test d'une image de service virtuel

Une fois les conditions préalables appliquées, procédez comme suit :

- [Etape 1 : démarrage de DevTest Workstation](#) (page 130)
- [Etape 2 : démarrage de VSE](#) (page 131)
- [Etape 3 : démarrage du serveur de démonstration](#) (page 131)
- [Etape 4 : exécution d'un scénario de test](#) (page 132)
- [Etape 5 : création d'un fichier de configuration](#) (page 133)
- [Etape 6 : activation du fichier de configuration](#) (page 134)
- [Etape 7 : configuration de l'enregistreur de VSE](#) (page 135)
- [Etape 8 : enregistrement du scénario de test](#) (page 137)
- [Etape 9 : déploiement du modèle de service virtuel](#) (page 139)
- [Etape 10 : exécution du test au niveau du modèle de service virtuel](#) (page 141)
- [Visualisation du didacticiel](#) (page 142)

Etape 1 : démarrage de DevTest Workstation

Procédez comme suit:

1. Vérifiez que le registre est en cours d'exécution.
 - Si le DevTest Server est installé sur votre ordinateur :
 - a. Démarrez le registre en cliquant sur le menu Démarrer, Tous les programmes, DevTest Solutions, EnterpriseDashboard (Tableau de bord d'entreprise). Patientez jusqu'à ce le message Enterprise Dashboard started (Tableau de bord d'entreprise démarré) s'affiche.
 - b. Démarrez le registre en cliquant sur le menu Démarrer, Tous les programmes, DevTest Solutions, Registry (Registre).
 - Si DevTest Workstation est installé sur votre ordinateur, utilisez un registre exécuté sur un autre ordinateur.
2. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, Workstation (Station de travail).
La boîte de dialogue Set DevTest Registry (Définir le registre DevTest) s'ouvre.
3. Sélectionnez un registre, puis cliquez sur OK.
4. La boîte de dialogue Login (Connexion) s'ouvre. Entrez un nom d'utilisateur et un mot de passe valides et cliquez sur Login (Connexion).
5. Poursuivez avec l'[étape 2 : démarrage du VSE](#) (page 131).

Etape 2 : démarrage du VSE

Procédez comme suit:

1. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, Virtual Service Environment (Environnement de service virtuel).

Lorsque la ligne Virtual service environment is now ready (L'environnement de service virtuel est prêt) s'affiche, le VSE est initialisé.

2. Poursuivez avec l'[étape 3 : démarrage du serveur de démonstration](#) (page 131).

Etape 3 : démarrage du serveur de démonstration

Une fois que la station de travail DevTest Workstation et l'environnement VSE sont en cours d'exécution, vous pouvez démarrer le serveur de démonstration utilisé pour finaliser ce didacticiel.

Procédez comme suit:

1. Cliquez sur Démarrer, Tous les programmes, DevTest Solutions, DevTest Demo Server (Serveur de démonstration de DevTest).

Lorsque la ligne Started in XXs:YYms (Démarré dans XXs:YYms) s'affiche, le serveur de démonstration a démarré.

2. Poursuivez avec l'[étape 4 : exécution d'un scénario de test](#) (page 132).

Etape 4 : exécution d'un scénario de test

Pour vérifier que LISA Bank est disponible, exécutez un scénario de test dans l'utilitaire ITR (Interactive Test Run, exécution d'un test interactif).

Procédez comme suit:

1. Dans le panneau Project (Projet), double-cliquez sur **webservices-xml.tst**.
Le scénario de test webservices s'ouvre dans un onglet.
2. Pour ouvrir l'utilitaire d'exécution d'un test interactif, cliquez sur ITR .
3. Pour exécuter le scénario de test sur le serveur de démonstration, cliquez sur Automatically Execute Test (Exécuter automatiquement le test) .
A l'issue du test, vous saurez que le serveur de démonstration est disponible et que le scénario de test s'exécute correctement.
4. Fermez la fenêtre ITR.
5. Pour fermer le scénario de test, cliquez sur X dans l'onglet de l'éditeur.
6. Continuez à l'[étape 5 : création d'un fichier de configuration](#) (page 133).

Etape 5 : création d'un fichier de configuration

Le scénario de test **webservices** est le pilote d'application source. Pour écouter et enregistrer les transactions de ce scénario de test, utilisez l'enregistreur de VSE dans DevTest Workstation. Insérez l'enregistreur de VSE entre l'application cliente source et le système dynamique. Le service Web du système dynamique LISA Bank est sur le port 8080 de localhost (hôte local). Configurez votre terminal de sorte à ce qu'il utilise le port d'écoute pour l'enregistreur de VSE : 8001. L'enregistreur intercepte la demande et la réponse sur le chemin vers et à partir de l'application de service Web dynamique sur le port 8080.

Dans chacune des étapes du scénario de test, un champ permet de spécifier un terminal. Par défaut, DevTest utilise la propriété {{ENDPOINT1}} pour définir ce terminal. Pour modifier rapidement le terminal pour chaque étape en changeant la configuration du projet pour la valeur {{ENDPOINT1}}, utilisez une propriété sans coder le terminal de manière irréversible.

La configuration par défaut est appelée project.config et est créée automatiquement pour un nouveau projet. Le fichier project.config se trouve dans le dossier Configs (Configurations) du panneau Project (Projet). Vous pouvez également créer un fichier de configuration.

Procédez comme suit:

1. Cliquez avec le bouton droit de la souris sur le dossier Configs (Configurations) dans le panneau Project (Projet) et sélectionnez Create New Config (Créer une configuration).
2. Saisissez le nom de la nouvelle configuration : **VSRecorder**.
3. Cliquez sur OK.

L'éditeur de propriétés ouvre le fichier de configuration VSRecorder.

Pour ajouter une propriété au fichier de configuration :

1. Pour ajouter une ligne, cliquez sur Add (Ajouter)  au bas de l'éditeur de propriétés.
2. Sélectionnez ENDPOINT1 dans le champ Key (Clé).
3. Entrez la valeur suivante dans le champ Value (Valeur) :

`http://localhost:8001/itkoExamples/EJB3UserControlBean?wsdl`

Remarque : Le port issu du scénario de test (8080) est remplacé par le port d'écoute de l'enregistreur du VSE : 8001. L'enregistreur peut intercepter la demande et la réponse sur le chemin vers et à partir de l'application de service Web dynamique sur le port 8080.

4. Cliquez sur Save (Enregistrer) dans la barre d'outils principale.
5. Pour fermer le fichier de configuration, cliquez sur X dans l'onglet de l'éditeur.

6. Poursuivez avec l'[étape 6 : activation du fichier de configuration](#) (page 134).

[Etape suivante](#) (page 134)

Etape 6 : activation du fichier de configuration

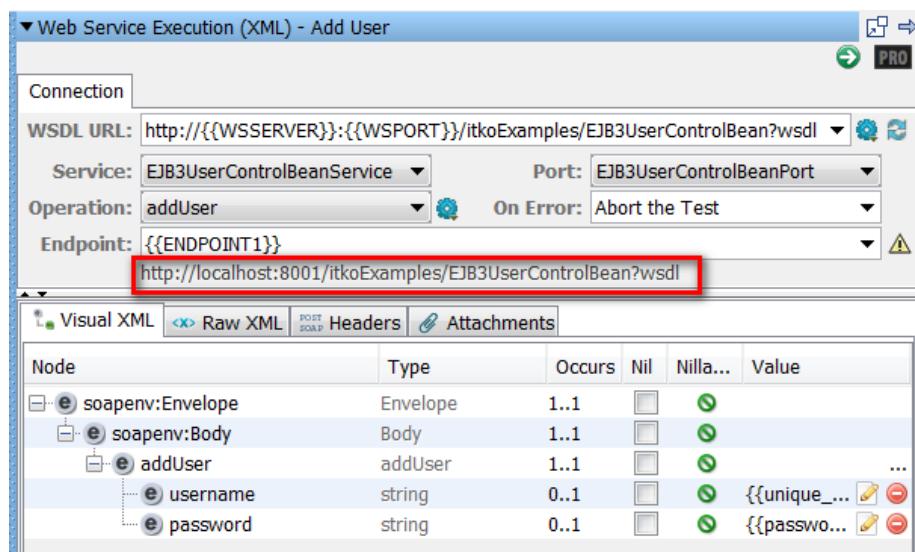
Pour que DevTest utilise le fichier de configuration tout juste créé comme fichier de configuration principal ou actif, activez-le.

Pour activer le fichier de configuration :

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur le fichier de configuration VSRecorder.
2. Sélectionnez Make Active (Activer).

Pour confirmer que la valeur ENDPOINT1 dans le fichier de configuration VSRecorder est utilisée :

1. Double-cliquez sur le scénario de test **webservices** dans le panneau Project (Projet). L'éditeur s'ouvre.
2. Double-cliquez sur l'étape de test **Add User** (Ajouter un utilisateur) et vérifiez que le terminal est défini sur le port 8001.



3. Passez à l'[étape 7 - configuration de l'enregistreur du VSE](#) (page 135).

Etape 7 : configuration de l'enregistreur du VSE

Une fois le terminal défini, vous pouvez configurer l'enregistreur de VSE.

Procédez comme suit:

1. Dans la barre d'outils principale, cliquez sur VSE Recorder (Enregistreur de VSE)

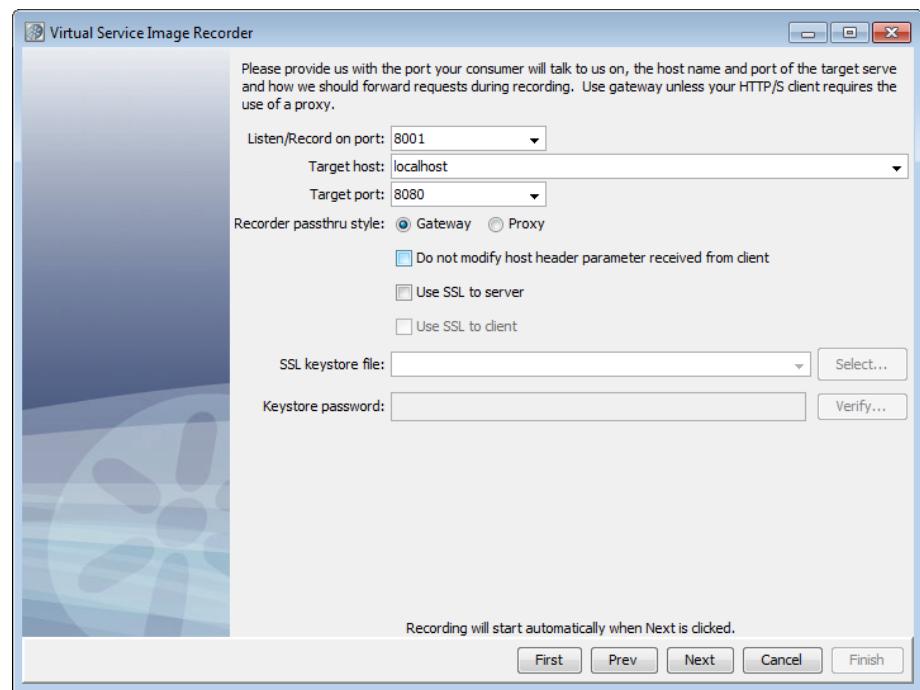


Dans l'onglet Basics (Paramètres de base), configurez les paramètres de l'image de service virtuel, qui stocke les données. Configurez le paramètre du modèle de service virtuel, qui crée le modèle de flux de travaux de transaction.

2. Remplacez le nom d'image de service par défaut, *newimage.vsi*, dans le champ Write image to (Emplacement d'écriture de l'image) par *VSETutorial.vsi*.
3. Laissez le champ Import traffic (Importer un fichier de trafic) vide.
Dans ce didacticiel, vous n'importerez pas de fichier de trafic existant.
4. Sélectionnez le protocole de transport HTTP/S.

Dans ce didacticiel, vous ne désensibilisez pas les données, car vous ne transférez aucune donnée sensible, comme des numéros de sécurité sociale ou des numéros de compte bancaire. Continuez jusqu'au champ Export to (Emplacement d'exportation), qui est utilisé pour exporter un fichier de trafic brut de l'enregistrement. Le fichier exporté sert de la sauvegarde brute de l'ensemble de la session d'enregistrement.

5. Cliquez sur Browse (Parcourir), nommez le fichier **rt_VSETutorial** et enregistrez-le dans le dossier VServices.
Le préfixe **rt_** indique qu'il s'agit d'un fichier de trafic brut.
6. Pour créer le fichier de modèle de service virtuel, cliquez sur Browse (Parcourir).
7. Saisissez le nom **VSETutorial**, puis cliquez sur **Save** (Enregistrer).
8. Laissez les sélections VS Model style (Style du modèle de service virtuel) telles quelles et cliquez sur Next (Suivant).
Dans la fenêtre suivante, désignez les ports de l'enregistreur.
9. Par défaut, l'enregistreur est défini pour écouter et enregistrer sur le port 8001, défini comme terminal dans le fichier de configuration VSRecorder. Ajoutez **localhost (hôte local)** au champ Target host (Hôte cible).
10. Mettez à jour Target port (Port cible), sinon le port des transactions du système dynamique LISA Bank sera le port **8080**.



11. Pour Recorder passthru style (Style d'authentification directe de l'enregistreur), sélectionnez l'option Gateway (Passerelle).
Une fois ces paramètres définis, vous pouvez lancer l'enregistrement.
12. Cliquez sur Next (Suivant).
13. Poursuivez avec l'[étape 8 : exécution d'un scénario de test](#) (page 137).

Etape 8 : enregistrement du scénario de test

Procédez comme suit:

1. Pour démarrer l'enregistrement, cliquez sur Next (Suivant).

Comme vous pouvez l'observer dans la fenêtre, l'enregistreur écoute le port cible 8080 sur le port 8001. La fenêtre indique également le nombre de sessions et de transactions. Le scénario de test n'ayant pas été exécuté, le nombre est défini sur 0. A l'issue de l'exécution du scénario de test, cette fenêtre contient trois transactions qui sont comptées.

2. Dans la station de travail DevTest Workstation, cliquez sur l'onglet du scénario de test, ouvrez l'utilitaire ITR (Exécuter un test interactif) et exéutez les étapes automatiquement.

3. Accédez à l'enregistreur et vérifiez que les trois transactions ont été enregistrées.

Une fois le scénario de test enregistré, vous pouvez finaliser la session d'enregistrement en cliquant sur Next (Suivant). DevTest traite l'enregistrement et présente une liste de transactions. Pour obtenir des transactions de service virtuel plus avancées, modifiez le modèle et les données post-enregistrement dans la fenêtre suivante. Pour votre scénario de test, vous pouvez continuer ; cliquez sur Next (Suivant).

Dans la fenêtre suivante, les paramètres de chemin de base, de liaison et de logique sont corrects.

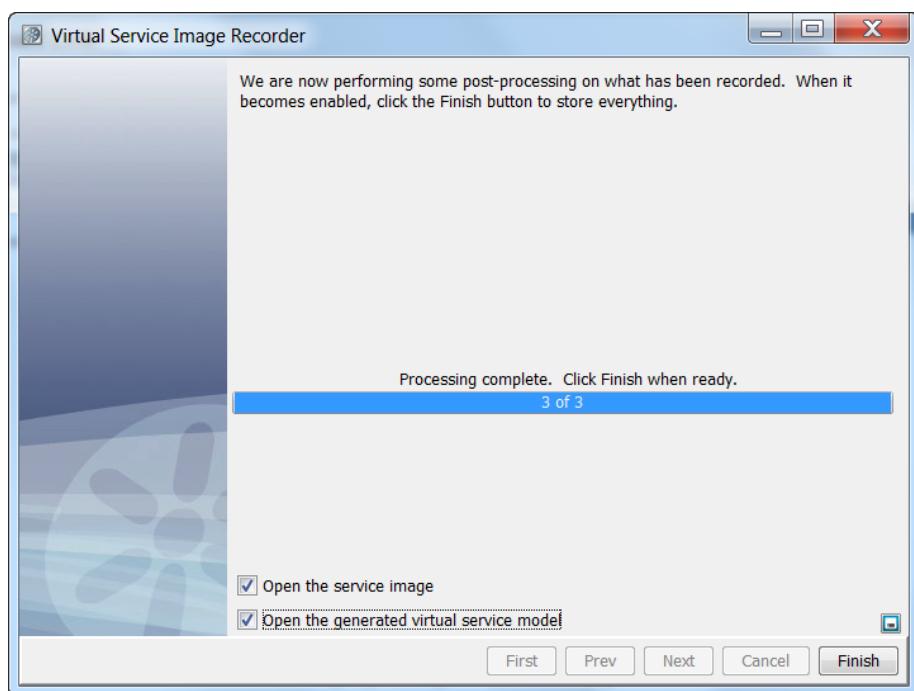
4. Cliquez sur Next (Suivant).

La fenêtre suivante indique qu'un gestionnaire de protocoles de données de services Web (SOAP) a été sélectionné pour le côté demande.

5. Laissez cette sélection telle quelle et cliquez sur Next (Suivant).

6. Ne modifiez pas la fenêtre suivante et appuyez sur Entrée.

DevTest traite l'enregistrement une dernière fois, puis crée le modèle de service virtuel et les fichiers d'image de service virtuel.



Vous pouvez ouvrir et modifier chaque fichier dans DevTest Workstation.

7. Pour afficher le modèle de service virtuel et les fichiers d'image de service virtuel, sélectionnez les cases à cocher Open the service image (Ouvrir l'image de service) et Open the generated virtual service (Ouvrir le modèle de service virtuel généré).
8. Cliquez sur Finish (Terminer).
DevTest ferme l'enregistreur et ouvre les fichiers sélectionnés dans la station de travail DevTest Workstation.
9. Passez à l'[étape 9 : déploiement du modèle de service virtuel](#) (page 139).

Etape 9 : déploiement du modèle de service virtuel

Pour déployer le modèle de service virtuel :

1. Pour fermer la fenêtre Demo Server (Serveur de démonstration), cliquez sur le X coin supérieur droit.

La fermeture de cette fenêtre arrête le serveur de démonstration, qui héberge LISA Bank. L'arrêt laisse le scénario de test sans système testé et permet d'utiliser le service virtuel créé.

2. Ouvrez DevTest Workstation et ouvrez la console de serveur, qui fournit l'interface utilisateur de VSE pour la gestion des services virtuels.
3. Dans le volet de navigation gauche, cliquez sur VSE.
4. Dans DevTest Workstation, cliquez avec le bouton droit de la souris sur le modèle de service virtuel VSETutorial dans le panneau Project (Projet) et sélectionnez Deploy/Redeploy to VSE (Déployer/redéployer vers l'environnement de service virtuel).

Dans la boîte de dialogue Deploy Virtual Service (Déployer un service virtuel), le modèle de service virtuel et le fichier de configuration VSRecorder sont sélectionnés. La balise de groupe est vide. La Concurrent capacity (Capacité simultanée) est définie sur 1, le Think time scale (Pourcentage de délai de réflexion) est défini sur 100 % et If service ends, automatically restart it (Redémarrer automatiquement le service) est sélectionné. La case à cocher Start the service on deployment (Lancer le service au déploiement) étant sélectionnée, le service virtuel démarre une fois déployé.

Pour partager le modèle et les configurations ou le déployer à distance, cliquez sur Save as MAR (Enregistrer en tant que fichier MAR) pour créer un fichier de MARS et distribuer le modèle selon les besoins. Toutes les configurations préremplies sont correctes, vous pouvez cliquer sur Deploy (Déployer) ; le service virtuel est déployé et s'affiche dans le tableau de bord de VSE.

Le service virtuel est déployé vers le VSE.

5. Pour revenir à la console de VSE, cliquez sur l'onglet DevTest Console dans votre navigateur.

Le modèle de service virtuel VSETutorial s'affiche dans l'onglet Services.

The screenshot shows the VSE interface with the 'Services' tab selected. A single service, 'VSETutorial', is listed in the table:

Name	Resource / Type	Status	Up-Time	Txn Count	Errors
VSETutorial	8001 /tkoExamples/EJB3UserControlBean	Running	0:00:51	0	

Below the table, the 'Virtual Service Details' panel is open for the 'VSETutorial' service. It displays the following configuration parameters:

Model Name :	VSETutorial	Config Name :	C:\Users\hoan02\lisatmp_6.1.14\ads\B1A2D4BED9102898D4A1F2E8275611E2\Configs\VSRecorder.config
Execution mode :	Most Efficient	Auto-Restart :	<input checked="" type="checkbox"/>
Last Start :	2012-11-05 08:41:14 AM	Last End :	0
Transaction Count :	0	Error Count :	0
Current txns/s :	0	Peak txns/s :	0
Capacity :	1	Think Scale :	100

Le service virtuel est démarré et prêt à traiter des transactions, qui sont suivies dans la colonne Txn Count (Nombre de transactions). Le service indique initialement qu'il a traité 0 transaction.

6. Exécutez le scénario de test et revenez à cette fenêtre.
7. Continuez à l'[étape 10 : test au niveau du modèle de service virtuel](#) (page 141).

Etape 10 : test au niveau du modèle de service virtuel

Pour effectuer le test au niveau du modèle de service virtuel :

1. Ouvrez le scénario de test **webservices** et exécutez-le dans l'ITR (Utilitaire d'exécution d'un test interactif).
2. A l'issue de l'exécution du test interactif, revenez au service virtuel.

Le nombre de transactions est désormais de 3, confirmant le test au niveau du service virtuel.

Name	Resource / Type	Status	Up-Time	Txn Count	Execution Mode	Errors
VSETutorial	8001:/tkoExamples/EJB3UserControlBean	Running	0:01:50	3	Most Efficient	

Virtual Service Details	
Model Name :	Config Name :
Execution mode :	Auto Restart :
Last Start :	Last End :
Transaction Count :	Error Count :
Current txn/s :	Peak txns/s :
Capacity :	Think Scale :

Remarque : Vous pouvez également simuler et exécuter un scénario de test ou une suite de tests au niveau du service virtuel.

3. Cliquez avec le bouton droit de la souris sur le scénario de test **webservices** dans le volet Project (Projet) et sélectionnez Stage a Quick Test (Simuler un test rapide).
4. Saisissez **10** pour Number of instances (Nombre d'instances).
5. Désactivez la case à cocher If test ends, restart it (Si le test se termine, redémarrez-le.) et cliquez sur OK.
6. Cliquez sur Play (Lire) dans la fenêtre Test Monitor (Moniteur de tests) ouverte et laissez le test s'exécuter.

Lorsque vous revenez au tableau de bord du VSE après l'exécution du scénario de test avec 10 instances, le nombre de transactions est passé à 33.

Name	Resource / Type	Status	Up-Time	Txn Count
VSETutorial	8001:/tkoExamples/EJB3UserControlBean	Running	1:28:15	33

7. Poursuivez à la section [Visualisation du didacticiel](#) (page 142).

Visualisation du didacticiel

Dans ce didacticiel, vous avez créé et testé un service virtuel pour vous familiariser avec les fonctionnalités de base de VSE.

Ce didacticiel vous a permis de :

- Créer et activer un fichier de configuration
- Configurer l'enregistreur de VSE
- Enregistrer un scénario de test
- Déployer un modèle de service virtuel
- Utiliser l'utilitaire d'exécution d'un test interactif et simuler un test sur un modèle de service virtuel

Chapitre 5: Didacticiel sur CA Continuous Application Insight

Dans ce didacticiel, vous générerez des transactions à l'aide de l'application de démonstration LISA Bank. Puis, vous afficherez les transactions dans la fenêtre Analyze Transactions (Analyser des transactions) du portail de DevTest.

Conditions préalables

- Les composants suivants sont exécutés : le tableau de bord Enterprise Dashboard, le registre, le serveur de démonstration, l'intermédiaire et le portail.

Ce chapitre traite des sujets suivants :

[Etape 1 - augmentation des niveaux de capture par défaut \(page 144\)](#)

[Etape 2 - génération de transactions à partir de l'application de démonstration \(page 145\)](#)

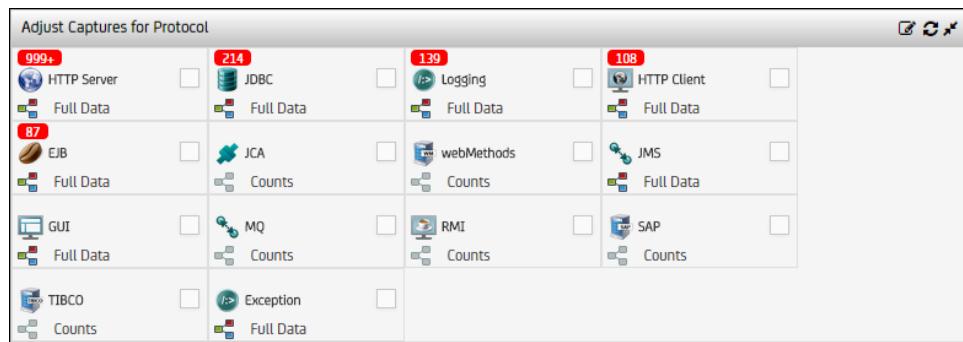
[Etape 3 - affichage des transactions dans la fenêtre d'analyse des transactions \(page 146\)](#)

Etape 1 - augmentation des niveaux de capture par défaut

Chaque protocole que l'agent Java de DevTest peut capturer a un niveau de capture.

Le niveau de capture par défaut est défini sur Comptes (Nombres). Pour afficher les transactions ultérieurement dans ce didacticiel, vous devez définir le niveau de capture pour les protocoles pertinents sur Full Data (Données complètes).

Le graphique suivant illustre le volet Adjust Captures for Protocol (Ajuster les captures de protocole) dans la fenêtre Agents.



Procédez comme suit:

1. Ouvrez le portail DevTest. Si le portail est exécuté sur un ordinateur local, l'URL sera **http://localhost:1507/devtest**.
2. Dans le menu de navigation gauche, sélectionnez Settings (Paramètres), Agents.
La fenêtre Agents s'affiche.
3. Dans le volet Agents, sélectionnez l'agent **JBoss_LISABank**.
4. Si le volet Adjust Captures for Protocol (Ajuster les captures de protocole) est réduit, développez-le.
5. Si certains protocoles n'apparaissent pas, cliquez sur Edit (Modifier) et sélectionnez Show All Protocols (Afficher tous les protocoles).
6. Vérifiez que les niveaux de capture pour les protocoles suivants sont définis sur Full Data (Données complètes) :
 - HTTP Server
 - JDBC
 - EJB
 - Logging (Journalisation)

Pour modifier un niveau de capture, sélectionnez la case à cocher du protocole et sélectionnez Full Data (Données complètes) dans la liste déroulante de niveau de capture.

Etape 2 - génération de transactions à partir de l'application de démonstration

DevTest Solutions inclut une application de démonstration nommée LISA Bank.

Dans la procédure suivante, vous vous connecterez à l'application, créerez un compte, puis vous déconnecterez.

Procédez comme suit:

1. Dans un navigateur Web, entrez **http://localhost:8080/lisabank**. Si le serveur de démonstration est exécuté sur un autre ordinateur, remplacez **localhost** par le nom d'hôte ou l'adresse IP de l'ordinateur distant.
La boîte de connexion s'ouvre.
2. Dans le champ Name (Nom), saisissez **lisa_simpson**.
3. Dans le champ Password (Mot de passe), saisissez **golisa**.
4. Cliquez sur Login (Connexion).
La page de bienvenue s'affiche. Le côté gauche contient des boutons permettant de réaliser diverses actions.
5. Pour créer un compte, procédez comme suit :
 - a. Cliquez sur New Account (Nouveau compte).
 - b. Dans le champ Account Name (Nom du compte), entrez **Mes économies**.
 - c. Dans le champ Initial Balance (Solde initial), remplacez la valeur par défaut par **500**.
 - d. Cliquez sur Add Account (Ajouter un compte).
6. Cliquez sur Log Out (Déconnexion).

Etape 3 - affichage des transactions dans la fenêtre d'analyse des transactions

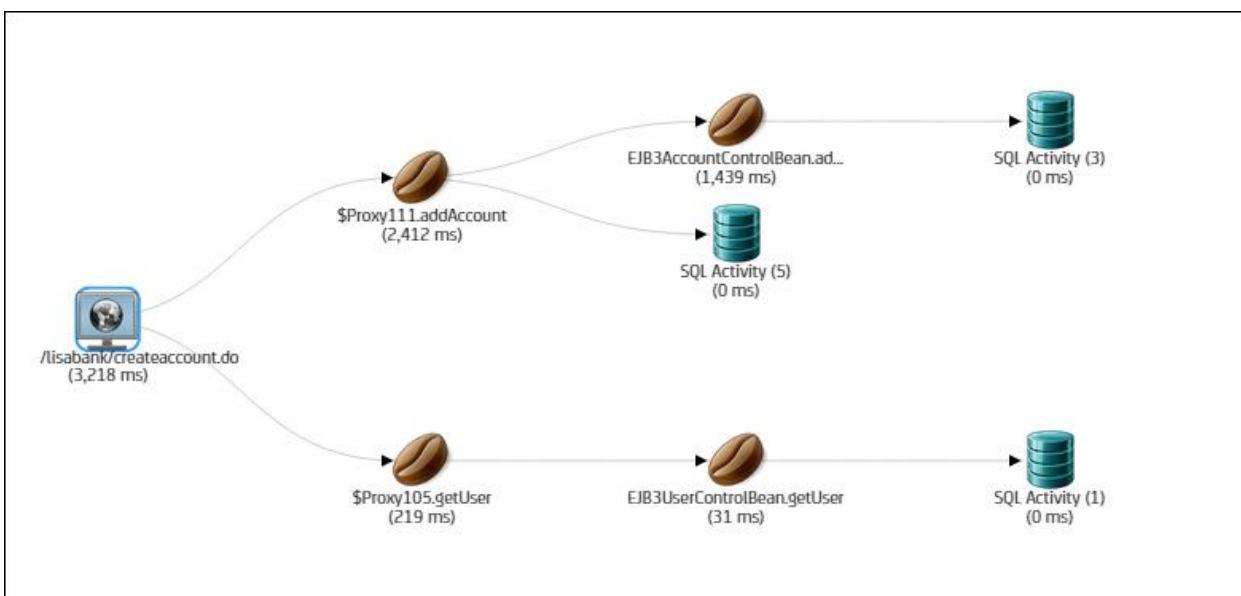
La fenêtre Analyze Transactions (Analyser des transactions) permet d'effectuer les tâches suivantes :

- Afficher des transactions
- Rechercher et filtrer des transactions
- Afficher les détails de composant

Le graphique suivant illustre un ensemble de transactions dans la fenêtre Analyze Transactions (Analyser des transactions). Les transactions apparaissent dans la vue List. Les transactions apparaissent dans l'ordre chronologique inverse.

Results: 7						Export/Import	Repeated Paths
	Name	Start Time	Wall Time	CPU Time	Agent	Actions	
1	/lisabank/logout.do	2014-10-10 16:43:41 722	844 ms	187 ms	JBoss_LISABank		
2	/lisabank/createaccount.do	2014-10-10 16:43:31 507	1330 ms	514 ms	JBoss_LISABank		
3	/lisabank/buttonclick.do	2014-10-10 16:43:09 846	1757 ms	561 ms	JBoss_LISABank		
4	/lisabank/login.do	2014-10-10 16:42:46 044	5706 ms	1981 ms	JBoss_LISABank		
5	/lisabank/home.do	2014-10-10 16:42:19 331	8210 ms	2917 ms	JBoss_LISABank		
6	/lisabank/	2014-10-10 16:42:09 704	9442 ms	3088 ms	JBoss_LISABank		
7	/lisabank/	2014-10-10 16:42:08 039	40 ms	0 ms	JBoss_LISABank		

Le schéma suivant illustre un graphique de chemin pour l'une des transactions.



Procédez comme suit:

1. Revenez au portail DevTest.
2. Sélectionnez Application Insight, Analyze Transactions (Analyser des transactions) dans le menu de navigation gauche.
La fenêtre Analyze Transactions (Analyser des transactions) s'ouvre.
3. Recherchez la transaction nommée **/lisabank/createaccount.do**.
4. Cliquez sur le bouton Open transaction details (Ouvrir les détails de transaction) dans la colonne Actions.
La boîte de dialogue des détails de transactions s'affiche. La zone supérieure contient le graphique de chemin. La zone inférieure contient des informations sur le noeud sélectionné.
5. Cliquez sur un noeud EJB dans le graphique de chemin.
6. Examinez les informations qui s'affichent dans chaque onglet.
7. Fermer la boîte de dialogue.

Glossaire

Archive de modèle (MAR)

Une *archive de modèle (MAR)* est le principal artefact de déploiement dans DevTest. Les fichiers MAR contiennent un actif principal, tous les fichiers secondaires qui sont requis pour exécuter l'actif principal, un fichier d'informations et un fichier d'audit. Pour plus d'informations, consultez la section Utilisation des archives de modèle (MAR) de la rubrique *Utilisation de CA Application Test*.

assertion

Une *assertion* est un élément qui s'exécute après l'exécution d'une étape et de tous ses filtres. Les assertions vérifient que les résultats de l'étape sont conformes aux prévisions. Une assertion est généralement utilisée pour modifier le flux d'un scénario de test ou le modèle de service virtuel. Les assertions globales s'appliquent à chaque étape d'un scénario de test ou d'un modèle de service virtuel. Pour plus d'informations, consultez la section Assertions de la rubrique *Utilisation de CA Application Test*.

Audit document (Document d'audit)

Un *document d'audit* permet de définir des critères de réussite d'un test, ou d'un ensemble de tests dans une suite. Pour plus d'informations, consultez la section Génération de documents d'audit dans la rubrique *Utilisation de CA Application Test*.

companion (compagnon)

Un *Companion (Compagnon)* est un élément exécuté avant et après chaque exécution de scénario de test. Les compagnons sont comparables à des filtres applicables à l'ensemble du scénario de test, par opposition à des étapes de test spécifiques. Les compagnons sont utilisés pour configurer le comportement global dans le scénario de test. Pour plus d'informations, consultez la section Compagnons de la rubrique *Utilisation de CA Application Test*.

Configuration

Une *configuration* est une collection nommée de propriétés qui spécifient en général des valeurs propres à un environnement pour le système testé. La suppression de données d'environnement codées de manière irréversible permet d'exécuter un scénario de test ou un modèle de service virtuel au niveau d'environnements différents en modifiant simplement des configurations. La configuration par défaut dans un projet est appelée `project.config`. Un projet peut contenir de nombreuses configurations, mais une seule configuration peut être active à la fois. Pour plus d'informations, consultez la section Configurations de la rubrique *Utilisation de CA Application Test*.

Conversation tree (Arborescence des conversations)

Une *arborescence des conversations* est un ensemble de noeuds liés qui représentent des chemins de conversation pour les transactions avec état dans une image de service virtuel. Chaque noeud porte une étiquette de nom d'opération ; par exemple : withdrawMoney. Exemple de chemin de conversation pour un système d'opérations bancaires : getToken, getAccount, withdrawMoney, deleteToken. Pour plus d'informations, reportez-vous à la rubrique *Utilisation de CA Service Virtualization*.

coordinator (coordinateur)

Le *coordinateur* reçoit les informations d'exécution de test sous forme de document et coordonne les tests exécutés sur un ou plusieurs serveurs de simulation. Pour plus d'informations, consultez la section Coordinator Server (Serveur de coordination) de la rubrique *Utilisation de CA Application Test*.

data protocol (protocole de données)

Un *protocole de données* est également appelé gestionnaire de données. Dans CA Service Virtualization, un protocole de données est responsable de la gestion de l'analyse des demandes. Certains protocoles de transport autorisent (ou requièrent) un protocole de données auquel le job de création de demandes est délégué. C'est pourquoi le protocole doit connaître la charge utile de la demande. Pour plus d'informations, consultez la section Utilisation de protocoles de données dans la rubrique *Utilisation de CA Service Virtualization*.

Data set (Ensemble de données)

Un *Data set (Ensemble de données)* est une collection de valeurs que vous pouvez utiliser pour définir des propriétés dans un scénario de test ou un modèle de service virtuel lors de l'exécution. Les ensembles de données fournissent un mécanisme permettant d'introduire des données de test externes dans un scénario de test ou un modèle de service virtuel. Vous pouvez créer des ensembles de données internes à DevTest, ou de manière externe ; par exemple, dans un fichier ou une table de base de données. Pour plus d'informations, consultez la section Ensembles de données de la rubrique *Utilisation de CA Application Test*.

desensitize (désensibiliser)

La *désensibilisation* consiste à convertir des données sensibles par des valeurs de substitution définies par l'utilisateur. Par exemple, les numéros de carte de crédit et les numéros de sécurité sociale sont des données sensibles. Pour plus d'informations, consultez la section Désensibilisation de données de la rubrique *Utilisation de CA Service Virtualization*.

Event (Événement)

Un *Event (Événement)* est un message sur une action qui s'est produite. Vous pouvez configurer des événements au niveau d'un scénario de test ou d'un modèle de service virtuel. Pour plus d'informations, consultez la section Introduction aux événements de la rubrique *Utilisation de CA Application Test*.

Filter (Filtre)

Un *filtre* est un élément exécuté avant et après une étape. Les filtres permettent de traiter les données des résultats ou de stocker des valeurs dans des propriétés. Les filtres globaux s'appliquent à chaque étape d'un scénario de test ou d'un modèle de service virtuel. Pour plus d'informations, consultez la section Filtres de la rubrique *Utilisation de CA Application Test*.

Groupe

Un *groupe* ou un *groupe de services virtuels* est une collection de services virtuels portant la même balise de groupe. Cela permet de les surveiller ensemble dans la console VSE.

Interactive Test Run (ITR) (Exécuter un test interactif)

L'utilitaire *Interactive Test Run (ITR)* (Exécuter un test interactif) permet d'exécuter un scénario de test ou un modèle de service virtuel étape par étape. Vous pouvez changer le scénario de test ou le modèle de service virtuel lors de l'exécution et le réexécuter pour vérifier les résultats. Pour plus d'informations, consultez la section Utilisation de l'utilitaire Interactive Test Run (ITR) (Exécuter un test interactif) de la rubrique *Utilisation de CA Application Test*.

Lab (Laboratoire)

Un *laboratoire* est un conteneur logique pour un ou plusieurs membres de laboratoire. Pour plus d'informations, consultez la section Laboratoires et membres de Laboratoire dans la rubrique *Utilisation de CA Application Test*.

magic date (date magique)

Pendant un enregistrement, un analyseur de dates analyse les demandes et les réponses. Une valeur correspondant à une définition étendue de formats de la date est convertie en *date magique*. Les dates magiques permettent de vérifier que le modèle de service virtuel fournit des valeurs de date explicites dans des réponses. Exemple de date magique : {{=doDateDeltaFromCurrent("yyyy-MM-dd","10");/*2012-08-14*/}. Pour plus d'informations, consultez la section Chaînes et dates magiques de la rubrique *Utilisation de CA Service Virtualization*.

Magic string (Chaîne magique)

Une *chaîne magique* est une chaîne générée pendant la création d'une image de service. Une chaîne magique est utilisée pour vérifier que les réponses fournies par le modèle de service virtuel contiennent des valeurs de chaîne explicites. Exemple de chaîne magique : {{=request_fname;/chris/}}. Pour plus d'informations, consultez la section Chaînes et dates magiques de la rubrique *Utilisation de CA Service Virtualization*.

Match tolerance (Tolérance de correspondance)

La *tolérance de correspondance* est un paramètre qui permet de contrôler la méthode utilisée par CA Service Virtualization pour comparer une demande entrante avec les demandes dans une image de service. Les options disponibles sont EXACT, SIGNATURE et OPERATION. Pour plus d'informations, consultez la section Tolérance de correspondance de la rubrique *Utilisation de CA Service Virtualization*.

Metrics (Mesures)

Les mesures permettent d'appliquer des méthodes et des mesures quantitatives aux performances et aux aspects fonctionnels de vos tests, ainsi qu'au système testé. Pour plus d'informations, consultez la section Génération de mesures de la rubrique *Utilisation de CA Application Test*.

Model Archive (MAR) Info (Fichier d'informations d'archive de modèle (MAR))

Un *Model Archive (MAR) Info (Fichier d'informations d'archive de modèle (MAR))* est un fichier qui contient des informations requises pour la création d'une archive de modèle MAR. Pour plus d'informations, consultez la section Utilisation des archives de modèle (MAR) de la rubrique *Utilisation de CA Application Test*.

navigation tolerance (tolérance de navigation)

La tolérance de navigation est un paramètre qui permet de contrôler le méthode utilisée par CA Service Virtualization pour rechercher la transaction suivante dans une arborescence des conversations. Les options disponibles sont CLOSE, WIDE et LOOSE. Pour plus d'informations, consultez la section Tolérance de navigation de la rubrique *Utilisation de CA Service Virtualization*.

Network graph (Graphique du réseau)

Le graphique de réseau est une zone de la console de serveur qui contient une représentation graphique du composant DevTest Cloud Manager et des laboratoires associés. Pour plus d'informations, consultez la section Démarrage d'un laboratoire de la rubrique *Utilisation de CA Application Test*.

Node (Noeud)

Une étape de test interne à DevTest peut également être appelée *node* (noeud), ce qui explique l'intégration du terme node dans l'ID de certains événements.

Path (Chemin)

Un *chemin* contient des informations sur une transaction capturée par l'agent Java. Pour plus d'informations, reportez-vous à la rubrique *Utilisation de CA Continuous Application Insight*.

Path graph (Graphique de chemin)

Un *graphique de chemin* contient une représentation graphique d'un chemin et ses trames. Pour plus d'informations, consultez la section Graphique de chemin de la rubrique *Utilisation de CA Continuous Application Insight*.

Project (Projet)

Un *projet* est une collection de fichiers DevTest liés. Les fichiers peuvent inclure des scénarios de test, des suites, des modèles de service virtuel, des images de service, des configurations, des documents d'audit, des documents de simulation, des ensembles de données, des moniteurs et des fichiers d'informations MAR. Pour plus d'informations, consultez la section Panneau Project (Projet) de la rubrique *Utilisation de CA Application Test*.

Property (Propriété)

Une *propriété* est une paire clé-valeur que vous pouvez utiliser comme variable d'exécution. Les propriétés peuvent stocker plusieurs types de données différents. Exemple de propriétés communes : LISA_HOME, LISA_PROJ_ROOT et LISA_PROJ_NAME. Une configuration est une collection nommée de propriétés. Pour plus d'informations, consultez la section Propriétés de la rubrique *Utilisation de CA Application Test*.

quick test (test rapide)

La fonctionnalité *Quick test (Test rapide)* permet d'exécuter un scénario de test avec une installation minimale. Pour plus d'informations, consultez la section Simulation d'un test rapide du *Utilisation de CA Application Test*.

Registry (Registre)

Le *registre* fournit un emplacement central pour l'enregistrement de tous les composants de DevTest Server et de DevTest Workstation. Pour plus d'informations, consultez la section Registre de la rubrique *Utilisation de CA Application Test*.

ressource

Un *actif* est un ensemble de propriétés de configuration groupées dans une unité logique. Pour plus d'informations, consultez la section Actifs de la rubrique *Utilisation de CA Application Test*.

service image (image de service)

Une *image de service* est une version normalisée de transactions enregistrées dans CA Service Virtualization. Chaque transaction peut être avec état (conversationnel) ou sans état. Une image de service peut être créée à l'aide de l'enregistreur d'image de service virtuel. Les images de service sont stockées dans un projet. Une image de service est également appelée *virtual service image (image de service virtuel, VSI)*. Pour plus d'informations, consultez la section Images de service de la rubrique *Utilisation de CA Service Virtualization*.

Simulator (Simulateur)

Un *simulateur* exécute les tests sous la surveillance du serveur de coordination. Pour plus d'informations, consultez la section Simulator Server (Serveur de simulation) de la rubrique *Utilisation de CA Application Test*.

Staging document (Document de simulation)

Un *document de simulation* contient les informations sur l'exécution d'un scénario de test. Pour plus d'informations, consultez la section Génération de documents de simulation dans la rubrique *Utilisation de CA Application Test*.

Subprocess (Sous-processus)

Un *sous-processus* est un scénario de test appelé par un autre scénario de test. Pour plus d'informations, consultez la section Génération de sous-processus de la rubrique *Utilisation de CA Application Test*.

Tableau de bord Continuous Service Validation (CVS) (Service de validation en continu)

Le *Continuous Service Validation (CVS) Dashboard* (*Tableau de bord du service de validation en continu*) permet de planifier l'exécution régulière de scénarios de test et des suites de test, sur une période étendue. Pour plus d'informations, consultez la section Service de validation en continu dans la rubrique *Utilisation de CA Application Test*.

test case (scénario de test)

Un *scénario de test* est une spécification de la procédure de test d'un composant métier dans le système testé. Chaque scénario de test contient une ou plusieurs étapes de test. Pour plus d'informations, consultez la section Génération de scénarios de test dans la rubrique *Utilisation de CA Application Test*.

test step (étape de test)

Une *étape de test* est un élément du flux de travaux de scénario de test qui représente une action de test unique à réaliser. Exemples d'étapes : Services Web, JavaBeans, JDBC et messagerie JMS. Une étape de test peut contenir des éléments de DevTest, tels que des filtres, des assertions et des ensembles de données liés. Pour plus d'informations, consultez la section Génération d'étapes de test de la rubrique *Utilisation de CA Application Test*.

Test suite (Suite de tests)

Une *suite de tests* est un groupe de scénarios de test, d'autres suites de tests, ou les deux, planifiés pour être exécutés l'un après l'autre. Un document de suite de tests spécifie le contenu de la suite, les rapports à générer et les mesures à collecter. Pour plus d'informations, consultez la section Génération de suites de tests de la rubrique *Utilisation de CA Application Test*.

Think time (Délai de réflexion)

Le *temps de réflexion* est la durée d'attente d'un scénario de test patiente avant d'exécuter une étape de test. Pour plus d'informations, consultez les sections Exemple d'ajout d'étape de test et Editeur de documents de simulation - Onglet Base de la rubrique *Utilisation de CA Application Test*.

transaction frame (Trame de transaction)

Une *trame de transaction* contient des données sur un appel de méthode intercepté par l'agent Java de DevTest ou un agent Light de CAI. Pour plus d'informations, consultez la section Transactions organisationnelles et trames de transaction de la rubrique *Utilisation de CA Continuous Application Insight*.

Virtual Service Environment (Environnement de service virtuel, VSE)

Le *Virtual Service Environment* (Environnement de service virtuel, VSE) est une application de DevTest Server qui permet de déployer et d'exécuter des modèles de service virtuel. VSE est également appelé CA Service Virtualization. Pour plus d'informations, reportez-vous à la rubrique *Utilisation de CA Service Virtualization*.

virtual service model (VSM, modèle de service virtuel)

Un *modèle de service virtuel* reçoit des demandes de service auxquelles il répond en l'absence du fournisseur de services réel. Pour plus d'informations, consultez la section Modèle de service virtuel de la rubrique *Utilisation de CA Service Virtualization*.