

DevTest Solutions

Utilisation de CA Service Virtualization

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: CA Service Virtualization 11

Présentation de la virtualisation	11
Types de virtualisation des services	12
Présentation de la virtualisation des services	12
Etapes de virtualisation de haut niveau	13
Virtualisation de systèmes de messagerie	15

Chapitre 2: installation 19

Procédure d'installation de CA Service Virtualization	19
Configuration système requise	20
Procédure de configuration de CA Service Virtualization	21
Configuration des utilisateurs et surveillance de l'utilisation	22
Définition du proxy pour l'hôte local	22
Définition de paramètres supplémentaires dans le fichier local.properties.....	22
Installation du simulateur de base de données	23
Autres propriétés de démarrage.....	25
Fichiers DDL pour les bases de données principales	25
Propriétés EclipseLink pour une session	26
Propriétés EclipseLink pour un schéma	30
agent APPC	34
Informations sur le déploiement technique d'un agent APPC.....	35
Installation de l'agent APPC	36
Configuration de l'agent APPC	39

Chapitre 3: Introduction à CA Service Virtualization 43

Utilisation de CA Service Virtualization	43
Composants de CA Service Virtualization	44
Modèles de service virtuel	45
Images de service	46
Importation de transactions	47
Fonctionnement de la virtualisation	48
Procédure de gestion des demandes conversationnelles.....	49
Chaînes et dates magiques.....	51
Chaînes magiques	52
Dates magiques.....	57
Introduction aux transactions de VSE	58

Transactions sans état et conversationnelles du VSE	58
Tolérance de navigation	61
Transactions logiques.....	62
Tolérance de correspondance	63
Opérateurs de correspondance d'arguments	64
Métatransactions et réponses spécifiques	65
Procédure de débogage des échecs de correspondance	66
Suivi des transactions.....	66
 Chapitre 4: Utilisation du portail DevTest avec CA Service Virtualization	 67
Ouverture du portail DevTest	67
 Chapitre 5: Création d'un service virtuel	 69
Enregistrement d'un site Web (HTTP ou HTTP/S)	70
Configuration d'un service virtuel.....	74
Enregistrement d'un service virtuel	76
 Chapitre 6: Modification des services virtuels	 81
Ouverture d'un service virtuel	81
Vue de transactions sans état	82
Vue de conversation	87
Réponses inconnues.....	88
Recherche de texte dans un service virtuel	90
Ajout d'une note à une signature	91
Ajout d'une étiquette à une transaction spécifique	91
Mise à jour manuelle d'un service virtuel	92
Recherche de transactions correspondant à une demande	101
URL de service virtuel.....	102
Déploiement d'un service virtuel	103
 Chapitre 7: Utilisation de la Workstation (Station de travail) et de la console avec CA Service Virtualization	 107
 Chapitre 8: Création d'images de service	 109
Ouverture d'une image de service	109
Combine Service Images (Combiner les images de service)	110
Suppression d'une image de service	111
Création d'une image de service.....	111
Création d'une image de service à partir de zéro	113

Création d'une image de service à partir d'un fichier WSDL.....	113
Création d'une image de service à partir d'un objet WADL.....	117
Création d'une image de service à partir d'un objet RAML	119
Création d'une image de service à partir de Layer 7.....	121
Création d'une image de service à partir d'une paire réponse-demande	122
Création d'une image de service à partir d'un objet PCAP	127
Création d'une image de service à partir d'un enregistrement	130
Création et déploiement d'un service virtuel avec VSEasy	226
Utilisation des modèles de service virtuel	227
Utilisation de protocoles de données	229

Chapitre 9: Modification d'images de service 295

Images de service héritées	295
Ouverture d'une image de service à modifier	296
Onglet Service Image (Image de service)	297
Onglet Transactions	301
Onglet Transactions pour les transactions sans état	302
Onglet transactions pour les conversations.....	315
Editeur de conversations	317
Images de service pour le protocole de transport JMS.....	328

Chapitre 10: Modification d'un modèle de service virtuel 329

Etape Virtual Service Router (Routeur de service virtuel)	331
Etape Virtual Service Tracker (Outil de suivi de service virtuel)	332
Etape Virtual Conversational/Stateless Response Selector (Sélecteur virtuel de réponse conversationnelle/sans état)	333
Etape Virtual HTTP/S Listener (Ecouteur virtuel de DRDA).....	334
Etape Virtual HTTP/S Live Invocation (Appel dynamique DRDA virtuel).....	336
Etape Virtual HTTP/S Responder (Répondeur virtuel HTTP/S)	338
Etape Virtual JDBC Listener (Ecouteur virtuel de JDBC)	339
Etape Virtual JDBC Responder (Répondeur virtuel JDBC)	340
Etape Socket Server Emulator (Emulateur de serveur de socket)	341
Etape Messaging Virtualization Marker (Marqueur de virtualisation de messagerie)	343
Etape Compare Strings for Response Lookup (Comparer les chaînes pour rechercher des réponses)	344
Etape Compare Strings for Next Step Lookup (Comparer les chaînes pour rechercher l'étape suivante)	346
Etape Virtual Java Listener (Ecouteur virtuel Java)	348
Etape Virtual Java Live Invocation (Appel dynamique virtuel via Java)	350
Etape Virtual Java Responder (Répondeur virtuel Java)	350
Etape Virtual TCP/IP Listener (Ecouteur virtuel de DRDA)	351
Etape Virtual TCP/IP Live Invocation (Appel dynamique DRDA virtuel).....	353
Etape Virtual TCP/IP Responder (Répondeur virtuel TCP/IP).....	354

Etape Virtual CICS Listener (Ecouteur virtuel CICS)	355
Etape Virtual CICS Responder (Répondeur virtuel CICS)	355
Etape CICS Transaction Gateway Listener (Ecouteur de la passerelle de transaction CICS)	356
Etape CICS Transaction Gateway Live Invocation (Appel dynamique de la passerelle de transaction CICS)	358
Etape CICS Transaction Gateway Responder (Répondeur de la passerelle de transaction CICS)	359
Etape Virtual DRDA Listener (Ecouteur virtuel de DRDA)	360
Etape Virtual DRDA Response Builder (Générateur de réponses virtuelles de DRDA)	360
Etape Virtual DRDA Live Invocation (Appel dynamique DRDA virtuel)	361
Etape IMS Connect Listener (Ecouteur d'IMS Connect)	362
Etape IMS Connect Live Invocation (Appel dynamique IMS Connect)	363
Etape Virtual IMS Connect Responder (Répondeur de connexion au service IMS virtuel)	364
Etapes JMS du VSE	365
Etape JCo IDoc Listener (Ecouteur de RFC via JCo)	370
Etape JCo IDoc Live Invocation (Appel dynamique de fichier IDoc via JCo)	371
Etape JCo IDoc Responder (Répondeur IDoc via JCo)	372
Etape JCo RFC Listener (Ecouteur de RFC via JCo)	372
Etape JCo RFC Live Invocation (Appel dynamique de RFC via JCo)	373
Etape JCo RFC Responder (Répondeur RFC via JCo)	374

Chapitre 11: Désensibilisation des données 375

Désensibilisation dynamique	376
Désensibilisation statique	376
Gestionnaire de protocoles de données Data Desensitizer (Désensibilisation de données)	377

Chapitre 12: Exécution de la virtualisation 379

Préparation de la virtualisation	380
Déploiement et exécution d'un service virtuel	382
Exécution de demandes dynamiques	385
Affichage de session et réparation de modèle	398
Mesures du VSE	401

Chapitre 13: Gestionnaire de VSE : gestion et déploiement de services virtuels 411

Installation du gestionnaire de VSE	411
Utilisation du gestionnaire de VSE	412

Chapitre 14: Commandes du VSE 417

Commande du gestionnaire du VSE - gestion des environnements de service virtuel	418
---	-----

Commandes ServiceManager - Gestion des services.....	419
Commande ServiceImageManager - Gestion des images de service.....	420
Commande VirtualServiceEnvironment.....	425

Chapitre 15: Propriétés VSE de l'agent Java	427
--	------------

Glossaire	431
------------------	------------

Chapitre 1: CA Service Virtualization

Ce chapitre traite des sujets suivants :

[Présentation de la virtualisation](#) (page 11)

[Types de virtualisation des services](#) (page 12)

[Présentation de la virtualisation des services](#) (page 12)

[Etapas de virtualisation de haut niveau](#) (page 13)

Présentation de la virtualisation

En règle générale, la *virtualisation* fait référence à la virtualisation de matériel, au niveau de laquelle le comportement d'un actif physique, par exemple un serveur ou une application dans un émulateur de logiciel, est simulé et l'émulateur est hébergé dans un environnement virtuel. L'environnement virtuel fournit la même communication avec l'actif émulé que l'environnement physique.

La virtualisation offre les avantages suivants :

- Amélioration de la gestion des actifs physiques et par conséquent, de la gestion des modifications et des configurations
- Amélioration de l'utilisation de la capacité physique et par conséquent, de la capacité des actifs physiques
- Amélioration de l'agilité, évitant des retards coûteux dus à l'attente de la reconfiguration ou du basculement des serveurs effectué par le service informatique

CA Service Virtualization permet la virtualisation des services. Le processus est en principe identique à celui de la virtualisation de matériel.

Types de virtualisation des services

CA Service Virtualization fournit deux configurations de produit, optimisées pour des applications clientes spécifiques :

- CA Service Virtualization
- CA Service Virtualization for Performance

CA Service Virtualization est plus adapté aux scénarios d'utilisation pour le développement, l'intégration, les tests et l'acceptation des utilisateurs. Les instances de ces produits traitent jusqu'à 10 transactions parallèles simultanément, ou environ 10 transactions par seconde.

CA Service Virtualization for Performance est spécifiquement destiné aux applications de test de performances et est évolutif, se trouvant uniquement limité par le matériel et le réseau sous-jacents.

Présentation de la virtualisation des services

La *virtualisation des services* consiste en l'imagerie du comportement de service logiciel et la modélisation d'un service virtuel pour représenter le service réel lors du développement et du test. La virtualisation des services complète la virtualisation de matériel et résout ses restrictions. Dans cette documentation, le terme *virtualisation* se réfère à la virtualisation des services.

Si vous ne souhaitez ou ne pouvez pas rester connecté au serveur pour vos tâches d'assurance qualité, CA Service Virtualization permet d'émuler le comportement du serveur.

Dans CA Service Virtualization, vous pouvez utiliser un environnement de service virtuel (VSE) comme client pour permettre l'enregistrement du service. Toutefois, vous utiliserez probablement davantage le service pour l'enregistrement dans votre client (par exemple, un navigateur ou une application interne).

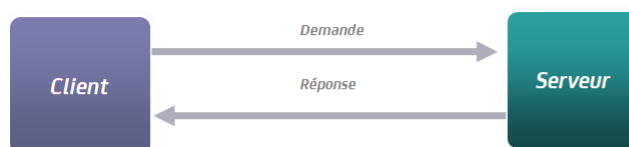
Etapes de virtualisation de haut niveau

Les étapes de haut niveau dans CA Service Virtualization sont les suivantes :

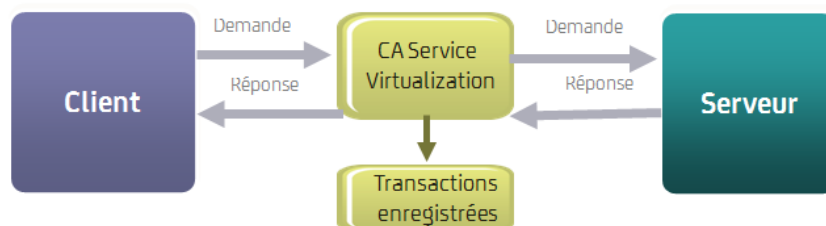
1. Prise d'une image du comportement du service (image de service) ; enregistrement des transactions gérées par le serveur
2. Création du service virtuel à partir du comportement (modèle de service virtuel)
3. Déployez le modèle de service virtuel dans l'environnement de services virtuels (VSE). Le modèle de service virtuel recherche dans les images de service capturées les réponses appropriées aux demandes parvenant au VSE.

Les graphiques suivants montrent que lors de l'enregistrement de l'image, le VSE fait office de mécanisme d'authentification directe entre le client et le serveur. Lors de la transmission des demandes et des réponses par le VSE, les transactions sont enregistrées.

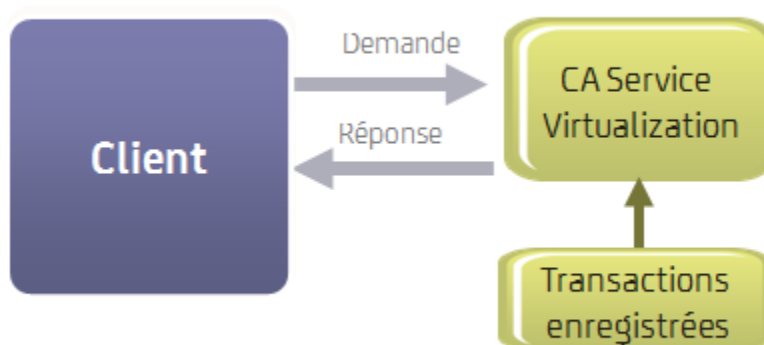
Fonctionnement normal



Enregistrement



Au moment de la virtualisation, en l'absence du serveur, le VSE répond aux demandes clientes en consultant les transactions enregistrées.



Virtualisation de systèmes de messagerie

Les intergiciels asynchrones (Message-Oriented Middleware, MOM), ou systèmes de messagerie, sont des services qui fournissent une méthode d'activation de la communication asynchrone entre deux ou plusieurs applications logicielles. Cette communication se produit toujours sous forme de messages. Les messages sont publiés sur des destinations de message configurées dans le MOM.

Types de destinations de message :

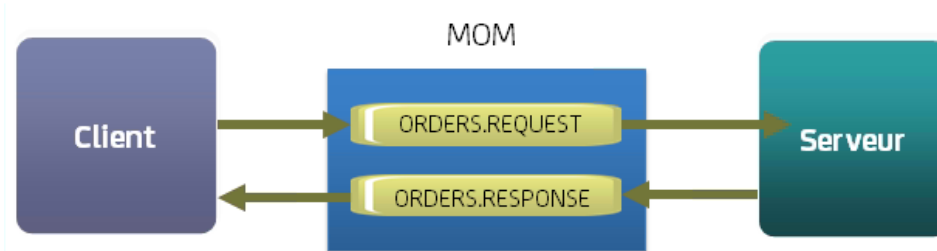
Files d'attente

Un éditeur ajoute un message à la file d'attente, tandis qu'un abonné extrait des messages de la file d'attente au fil de leur arrivée.

Rubriques

Un éditeur publie un message dans une rubrique et tous les abonnés à cette rubrique reçoivent le message.

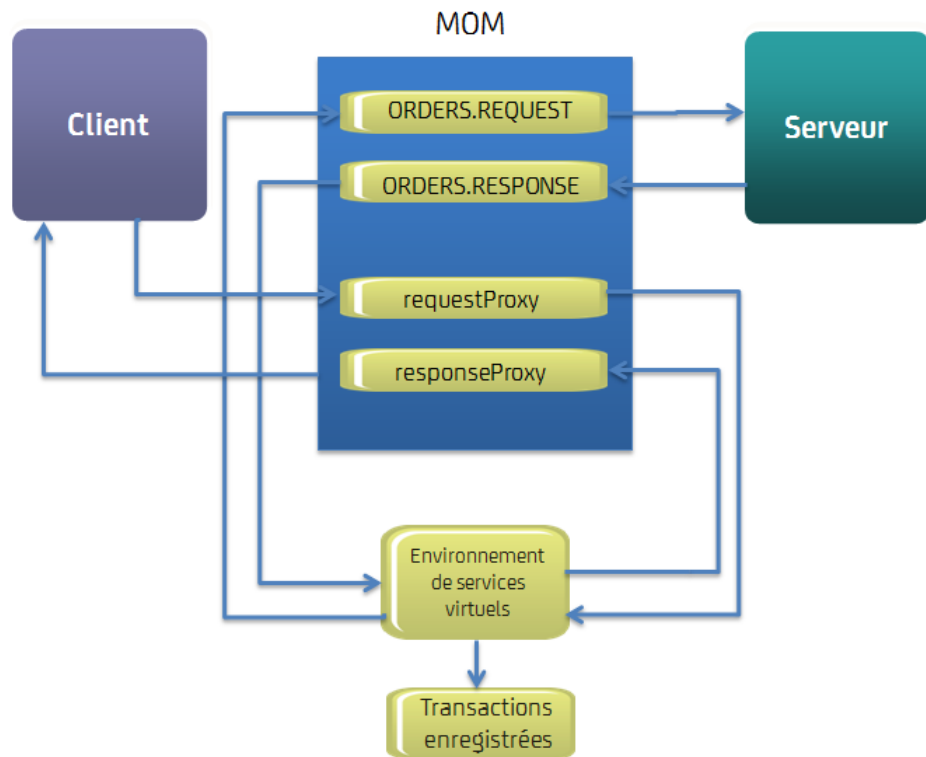
Le graphique suivant illustre un service simple basé sur un message. Dans ce scénario, le client ajoute des messages dans une file d'attente (**ORDERS.REQUEST**), récupérée par le serveur. La réponse du serveur est au format de messages ajoutés à une autre file d'attente (**ORDERS.RESPONSE**). Puis, le client les récupère.



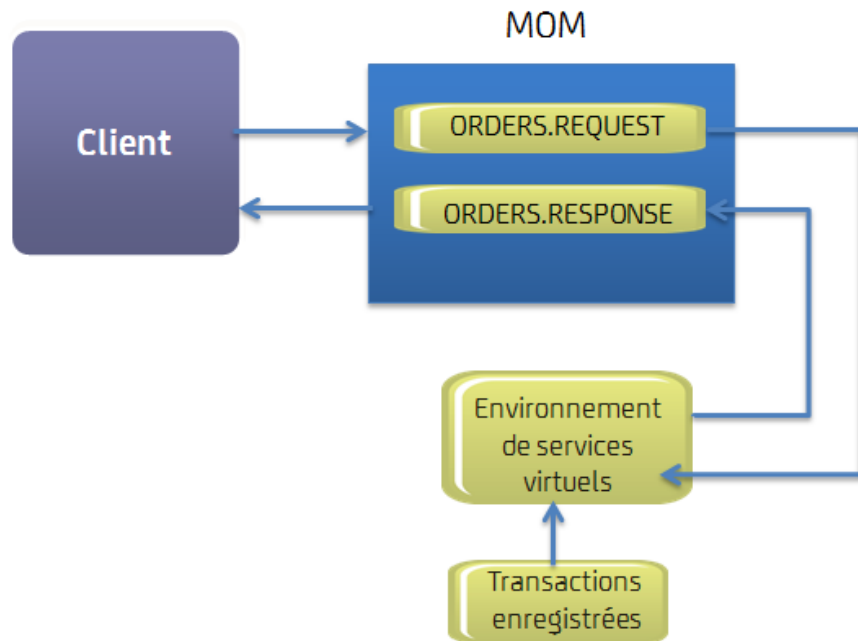
Les variations possibles sont les suivantes :

- Utilisation de rubriques au lieu de files d'attente
- Plusieurs réponses à une demande unique, qui peut potentiellement cibler différentes destinations

CA Service Virtualization tente de virtualiser le serveur. En mode d'enregistrement, le VSE requiert des destinations de proxy supplémentaires (files d'attente **requestProxy** et **responseProxy** dans le graphique suivant) que le client utilise au lieu de leurs équivalents. Le serveur effectue toujours des écoutes et des publications dans des destinations réelles. VSE agit comme une authentification directe entre ces destinations de proxy et réelles. VSE enregistre le trafic pour créer le modèle de service virtuel et l'image de service requise pour la virtualisation.



Ultérieurement, lorsque le VSE virtualise le serveur, il fonctionne simplement avec les destinations réelles. Le VSE ne requiert pas de destinations de proxy.



Chapitre 2: installation

Ce chapitre traite des sujets suivants :

[Procédure d'installation de CA Service Virtualization](#) (page 19)

[Procédure de configuration de CA Service Virtualization](#) (page 21)

[Installation du simulateur de base de données](#) (page 23)

[Fichiers DDL pour les bases de données principales](#) (page 25)

[agent APPC](#) (page 34)

Procédure d'installation de CA Service Virtualization

Le logiciel CA Service Virtualization est installé avec le serveur DevTest. Pour plus d'informations sur l'installation et la configuration de DevTest Solutions, reportez-vous à la rubrique *Installation*.

Pour utiliser CA Service Virtualization, les processus (ou services) suivants doivent être exécutés :

- Registry (Registre - Service de registre DevTest)
- VirtualServiceEnvironment (Environnement de service virtuel - service VSE)

Lorsqu'il est défini en tant que service de niveau serveur, CA Service Virtualization peut coexister avec un registre auquel un coordinateur et un simulateur sont connectés. Le simulateur et le coordinateur ne sont pas requis pour l'exécution de CA Service Virtualization.

Configuration système requise

Les ressources système suivantes pour CA Service Virtualization constituent la configuration requise de référence uniquement.

- **UC** : 2 GHz ou supérieur
- **RAM** : 2 Go ou plus
- **Espace disque** : espace disque disponible de 5 Go
- **Système d'exploitation** recommandé : système d'exploitation 64 bits. Système pris en charge : Windows 2008, 7, 8, Linux, Solaris, AIX 6.1 (LISA 5.0 et version ultérieure)

Pour des déploiements de CA Service Virtualization à grande échelle, les ressources suivantes sont recommandées :

- 256 threads de service virtuel par instance de VSE
- Cœur de processeur unique et 2 Go de RAM par instance de VSE

Exemple : 1 000 000 transactions par jour

- 1 thread pour la prise en charge de tests fonctionnels par chaque service
- Environ 6 threads par service pour la prise en charge de la virtualisation des tests de charge et de performance
- 8 noyaux correspondent à 2 048 threads de service virtuel simultanés
- 16 Go de mémoire RAM (pour DevTest)

Pour plus d'informations sur d'autres configurations système requises, reportez-vous à la section Configuration système requise et Conditions requises de la rubrique *Installation*.

Structure et fichiers des répertoires de logiciel

La liste suivante décrit la structure de répertoires de DevTest Solutions. Les répertoires sont disponibles dans le dossier d'installation racine de DevTest.

bin

Contient des fichiers exécutables, par exemple les fichiers TestRegistry.exe, Workstation.exe, VirtualServiceEnvironment.exe et VSEManager.exe.

DemoServer

Contient le serveur de démonstration DevTest.

doc

Contient la documentation de DevTest.

exemples/vse

Ce répertoire contient des exemples liés au VSE.

tmp

Ce répertoire contient l'espace de travail du VSE dans lequel il stocke temporairement des conversations et des transactions sans état.

vseDeploy

Ce répertoire contient des modèles de service virtuel déployés et des données connexes.

Procédure de configuration de CA Service Virtualization

Pour configurer CA Service Virtualization, effectuez les tâches suivantes :

1. [Configuration des utilisateurs et surveillance de l'utilisation](#) (page 22)
2. [Définissez le proxy pour l'hôte local](#) (page 22).
3. (Facultatif) [Définition de paramètres supplémentaires dans le fichier local.properties](#) (page 22)

Configuration des utilisateurs et surveillance de l'utilisation

Votre licence DevTest Solutions s'applique au produit complet. Le contrat de licence prévoit un nombre maximum spécifique d'utilisateurs simultanés de type SV Power User (Utilisateur avancé SV) (parmi d'autres types d'utilisateur). Un administrateur accorde à des utilisateurs de CA Service Virtualization diverses autorisations associées au type d'utilisateur SV Power User (Utilisateur avancé SV). Un administrateur génère de manière régulière un rapport d'audit d'utilisation pour surveiller la conformité de l'utilisation simultanée maximum. Consultez la rubrique *Administration*.

Définition du proxy pour l'hôte local

Lorsque DevTest sert de client HTTP au VSE, le trafic HTTP issu de DevTest doit être transmis au VSE. La configuration du VSE comme proxy Web est une méthode couramment utilisée pour transmettre le trafic HTTP. Toutefois, l'utilisation de proxy est désactivée par défaut pour des noms simples tels que localhost (hôte local).

Vous pouvez modifier ce comportement dans le fichier **local.properties**.

Procédez comme suit:

1. Ouvrez le fichier local.properties dans le dossier d'installation racine de DevTest.
2. Supprimez les commentaires de la propriété **lisa.http.webProxy.nonProxyHosts.excludeSimple**.
3. Définissez la valeur de cette propriété sur false.
`lisa.http.webProxy.nonProxyHosts.excludeSimple=false`
4. Enregistrez et fermez le fichier.

Définition de paramètres supplémentaires dans le fichier local.properties

Vous pouvez également définir les configurations supplémentaires suivantes dans le fichier local.properties :

- **lisa.vseName=VSENAME**
Pour renommer le serveur du VSE, ajoutez cette propriété et modifiez la valeur pour laquelle VSENAME est le nom du serveur du VSE.
- **lisa.registryName=REGISTRY** ou
- **lisa.registryName=tcp://111.666.11.198:2010/REGISTRY**

Pour vous connecter à un autre registre de test :

1. Ajoutez la propriété **lisa.registryName** au fichier **local.properties**.
2. Remplacez les valeurs de REGISTRY par le nom du nouveau registre de test.

Installation du simulateur de base de données

Pour enregistrer le trafic JDBC, installez le pilote JDBC de simulation de DevTest sur le client de base de données. Le client de base de données utilise le pilote DevTest au lieu du pilote réel.

Procédez comme suit:

1. Copiez le fichier JAR **lisajdbcsim.jar** du répertoire LISA_HOME\lib dans la variable classpath de votre base de données.

Le fichier JAR contient le pilote JDBC de simulation de DevTest. Pour simplifier l'utilisation du serveur de démonstration comme client de base de données, le pilote est déjà copié sur le serveur de démonstration dans le répertoire DEMO_HOME\jboss\server\default\lib.

2. Définissez la classe de pilotes sur **com.itko.lisa.vse.jdbc.driver.Driver**. Définissez la classe de pilotes, selon le type de JDBC utilisé par le client de base de données :

DriverManager style (Style de gestionnaire de pilotes)

Dans un serveur d'applications, il est peu probable que le client de base de données utilise le DriverManager (Gestionnaire de pilotes) Java pour acquérir des connexions. Si c'est le cas du serveur d'applications, ajoutez la commande suivante à la commande de démarrage du client de base de données :

```
-Djdbc.drivers=com.itko.lisa.vse.jdbc.driver.Driver
```

Si cette propriété est déjà utilisée, ajoutez le pilote DevTest au début. Séparez le pilote DevTest des autres noms de classe de pilotes par des deux-points (:).

DataSource style (Style de gestionnaire de pilotes)

Si le client de base de données utilise le style DataSource pour obtenir une connexion (tel que le serveur de démonstration), mettez à jour sa configuration. A l'emplacement de spécification d'une définition de source de données, spécifiez **com.itko.lisa.vse.jdbc.driver.Driver** comme pilote JDBC à utiliser et une URL de connexion.

3. Pour obtenir une authentification directe, modifiez l'URL de connexion de sorte que le pilote JDBC de simulation DevTest puisse identifier les informations réelles du pilote. Formatez l'URL de connexion comme suit :

```
nom=valeur[;nom=valeur...]
```

name

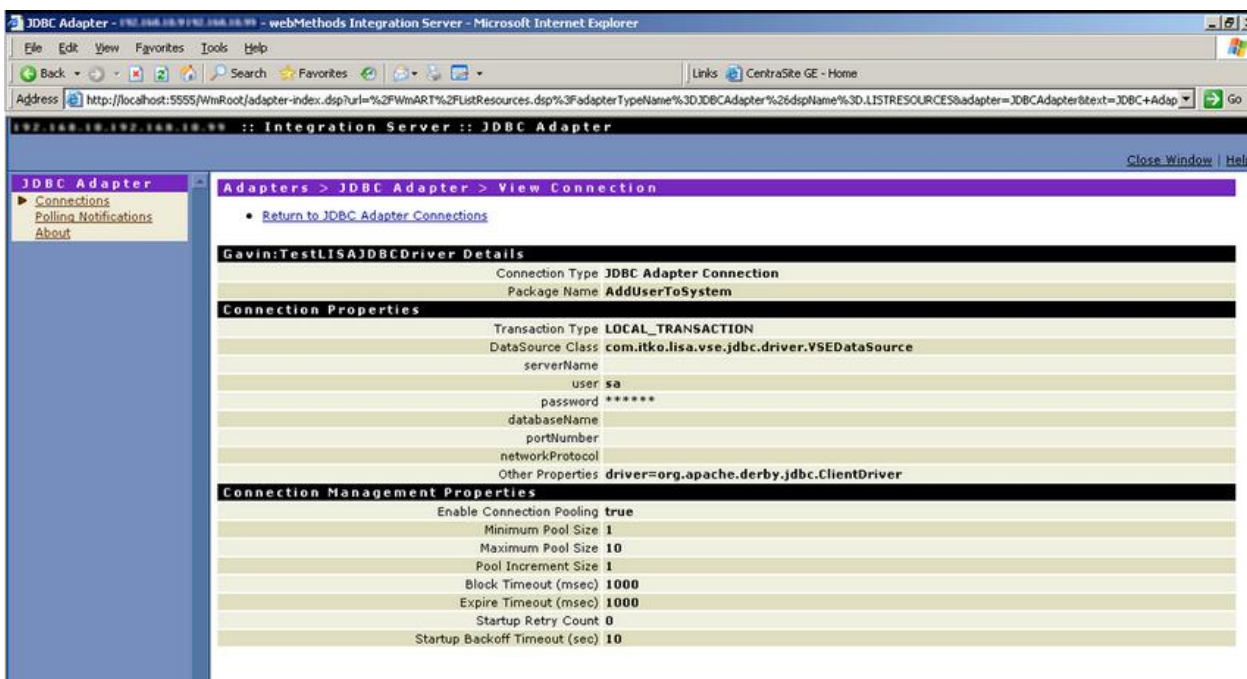
jdbc:lisasim:driver : la valeur doit être le nom complet de la classe du pilote JDBC réel à utiliser.

URL : la valeur doit être définie sur l'URL de connexion attendu par le pilote réel. Elle doit être définie comme dernière propriété de sorte à contenir des points-virgules.

Remarque : VSE ne prend pas en charge le pilote léger Oracle comme pilote d'authentification directe. Le pilote léger Oracle ne fournit pas l'implémentation JDBC complète. Si vous utilisez une base de données Oracle, utilisez d'autres pilotes JDBC pour la virtualisation.

Pour obtenir un exemple de définition de l'URL de connexion, reportez-vous au fichier **DEMO_HOME\jboss\server\default\deploy\itko-example-ds.xml**.

Le graphique suivant illustre un exemple de virtualisation de la base de données dans un environnement WebMethods.



Pour utiliser les pilotes de simulation et de CA Continuous Application Insight, convertissez le pilote de simulation en pilote extérieur. Spécifiez la classe et l'URL CAI. Pour obtenir un exemple de définition d'une source de données DevTest sur JBoss pour le serveur de démonstration DevTest, consultez le fichier **itko-example-ds.xml** dans le dossier **DEMO_HOME\jboss\server\default\deploy**.

Autres propriétés de démarrage

Quel que soit le style de connexion du client de base de données, vous pouvez ajouter les propriétés suivantes à la commande de démarrage de sorte que le client de base de données affecte le pilote de simulation.

`lisa.jdbc.sim.require.remote`

Spécifie si l'exécution du pilote requiert une connexion active à un serveur DevTest Workstation ou VSE. Il s'agit de la meilleure méthode pour synchroniser un client de base de données avec le VSE et enregistrer ou lire une activité de base de données de démarrage effectuée par le serveur.

Valeurs :

- Si elle est définie sur **true**, le pilote sera bloqué jusqu'à l'établissement d'une connexion active avec un serveur DevTest Workstation ou VSE.
- Si elle est définie sur **false**, le pilote ne requiert aucune connexion active.

Valeur par défaut : false

`lisa.jdbc.sim.port`

Définit le port IP sur lequel le pilote écoute les connexions à partir d'un enregistreur ou d'un modèle de service virtuel en cours d'exécution.

Valeur par défaut : 2999

Fichiers DDL pour les bases de données principales

Pour que DevTest génère un DDL pour un fichier, ajoutez les lignes suivantes au fichier **local.properties** :

```
eclipselink.ddl-generation=create-tables
eclipselink.ddl-generation.output-mode=sql-script
eclipselink.target-database=Oracle
```

Le démarrage de DevTest avec ces propriétés permet de créer les fichiers suivants qui contiennent le DDL nécessaire :

- `createDDL.jdbc`
- `dropDDL.jdbc`

Vous pouvez générer des DDL pour un SGBD différent en modifiant la valeur `target-database`. Pour plus d'informations, consultez la section :

- [Propriétés de l'unité de persistance EclipseLink pour la session](#) (page 26)
- [Propriétés de l'unité de persistance EclipseLink pour la génération de schémas](#) (page 30)

Propriétés EclipseLink pour une session

Les propriétés de l'unité de persistance EclipseLink JPA que vous pouvez définir dans un fichier **persistence.xml** pour configurer des extensions EclipseLink pour une session, ou comme base de données cible et serveur d'applications, se présentent comme suit.

eclipseLink.session-name

Spécifiez le nom sous lequel la session EclipseLink est stockée dans le gestionnaire de session statique. Utilisez cette option si vous devez accéder à la session partagée EclipseLink en dehors du contexte de la programmation de persistance (JPA). Cette option permet également d'utiliser une session EclipseLink existante configurée dans un fichier EclipseLink **sessions.xml**.

Valeurs valides : nom de session EclipseLink valide et unique dans un déploiement de serveur

Exemple :

```
persistence.xml file<property value="MySession"/>
```

Exemple :

```
property Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
t(PersistenceUnitProperties.SESSION_NAME, "MySession");
```

Valeur par défaut : nom unique généré par EclipseLink

eclipseLink.sessions-xml

Spécifiez les informations de persistance qui sont chargées à partir du fichier de configuration de session EclipseLink **sessions.xml**.

Vous pouvez utiliser cette option comme alternative aux annotations et au fichier XML de déploiement. Si vous spécifiez cette propriété, EclipseLink remplacera toutes les annotations de classe et le mappage relationnel d'objet dans les fichiers persistence.xml, ORM.xml et d'autres fichiers de mappage.

Pour indiquer la session, définissez la propriété **eclipseLink.session-name**.

Remarque : Si vous ne spécifiez pas la valeur de cette propriété, le fichier **sessions.xml** ne sera pas utilisé.

Valeurs valides : nom de ressource du fichier XML de sessions

Exemple :

```
persistence.xml file<property value="mysession.xml"/>Example: property
Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put(Persist
enceUnitProperties.SESIONS_XML, "mysession.xml");
```

eclipseLink.session-event-listener

Spécifiez un écouteur d'événements du descripteur à ajouter lors de l'amorçage.

Valeurs valides : nom complet d'une classe qui implémente l'interface org.eclipse.persistence.sessions.SessionEventListener.

Exemple :

```
Persistence.xml file<property value="mypackage.MyClass.class"/>Example:
property Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put(Persist
enceUnitProperties.SESSION_EVENT_LISTENER_CLASS, "mypackage.MyClass.class");
```

eclipselink.session.include.descriptor.queries

Activez ou désactivez la copie par défaut de toutes les requêtes nommées des descripteurs vers la session. Ces requêtes incluent celles définies à l'aide de l'API EclipseLink, des méthodes de modification de descripteur, etc.

Valeurs valides :

- **True** : active la copie par défaut de toutes les requêtes nommées des descripteurs vers la session.
- **False** : désactive la copie par défaut de toutes les requêtes nommées des descripteurs vers la session.

Exemple :

```
Persistence.xml file<property value="false"/>Example: property Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
(PersistenceUnitProperties.INCLUDE_DESCRIPTOR_QUERIES, "false");
```

Valeur par défaut : true**eclipselink.target-database**

Spécifiez le type de base de données utilisée par votre application JPA.

Valeurs valides :

Les valeurs valides à utiliser dans un fichier **persistence.xml** et pour **org.eclipse.persistence.config.TargetDatabase** sont les suivantes :

- **Attunity** : configurez le fournisseur de persistance pour utiliser une base de données Attunity.
- **Auto** : EclipseLink accède à la base de données et utilise les métadonnées fournies par le pilote JDBC pour déterminer la base de données cible (valide pour les pilotes JDBC qui prennent en charge ces métadonnées).
- **Cloudscape** : configurez le fournisseur de persistance pour utiliser une base de données Cloudscape.
- **Base de données** : si votre base de données cible n'est pas répertoriée à cet emplacement et que votre pilote JDBC prend en charge l'utilisation des métadonnées requises par l'option Auto, configurez le fournisseur de persistance pour utiliser une sélection générique.
- **DB2** : configurez le fournisseur de persistance pour utiliser une base de données DB2.
- **DB2Mainframe** : configurez le fournisseur de persistance pour utiliser une base de données DB2Mainframe.

- **DBase** : configurez le fournisseur de persistance pour utiliser une base de données DBase.
- **Derby** : configurez le fournisseur de persistance pour utiliser une base de données Derby.
- **HSQL** : configurez le fournisseur de persistance pour utiliser une base de données HSQL.
- **Informix** : configurez le fournisseur de persistance pour utiliser une base de données Informix.
- **JavaDB** : configurez le fournisseur de persistance pour utiliser une base de données Java DB.
- **MySQL** : configurez le fournisseur de persistance pour utiliser une base de données MySQL.
- **Oracle** : configurez le fournisseur de persistance pour utiliser une base de données Oracle.
- **PointBase** : configurez le fournisseur de persistance pour utiliser une base de données PointBase.
- **PostgreSQL** : configurez le fournisseur de persistance pour utiliser une base de données PostgreSQL.
- **SQLAnywhere** : configurez le fournisseur de persistance pour utiliser une base de données SQLAnywhere.
- **SQLServer** : configurez le fournisseur de persistance pour utiliser une base de données SQLServer.
- **Sybase** : configurez le fournisseur de persistance pour utiliser une base de données Sybase.
- **TimesTen** : configurez le fournisseur de persistance pour utiliser une base de données TimesTen. Vous pouvez également définir la valeur sur le nom complet d'une sous-classe de la classe `org.eclipse.persistence.platform.DatabasePlatform`.

Exemple :

```
Persistence.xml file<property value="Oracle"/>Example: property Mapimport
org.eclipse.persistence.config.TargetDatabase;import
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
t(PersistenceUnitProperties.TARGET_DATABASE, TargetDatabase.Oracle);
```

Valeur par défaut : Auto

eclipselink.target-server

Spécifiez le type de serveur d'applications utilisé par votre application JPA.

Valeurs valides :

Les valeurs suivantes sont les valeurs valides à utiliser dans le fichier **persistence.xml** et dans **org.eclipse.persistence.config.TargetServer** :

- **None (Aucun)** : configurez le fournisseur de persistance de sorte à n'utiliser aucun serveur d'applications.
- **WebLogic** : configurez le fournisseur de persistance pour utiliser un serveur Oracle WebLogic. Ce serveur définit cette propriété de manière automatique. Définissez-la uniquement si elle est désactivée.
- **WebLogic_9** : configurez le fournisseur de persistance pour utiliser un serveur Oracle WebLogic version 9.
- **WebLogic_10** : configurez le fournisseur de persistance pour utiliser un serveur Oracle WebLogic version 10.
- **OC4J** : configurez le fournisseur de persistance pour utiliser OC4J.
- **SunAS9** : configurez le fournisseur de persistance pour utiliser un serveur Sun Application Server version 9. Ce serveur définit cette propriété de manière automatique. Définissez-la uniquement si elle est désactivée.
- **WebSphere** : configurez le fournisseur de persistance pour utiliser un serveur WebSphere Application Server.
- **WebSphere_6_1** : configurez le fournisseur de persistance pour utiliser un serveur WebSphere Application Server version 6.1.
- **JBoss** : configurez le fournisseur de persistance pour utiliser un serveur d'applications JBoss.
- Nom complet d'une classe de serveur personnalisée qui implémente l'interface **org.eclipse.persistence.platform.ServerPlatform**

Exemple :

```
persistence.xml file<property value="OC4J_10_1_3"/>Example: property
Mapimport org.eclipse.persistence.config.TargetServer;import
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
(PersistenceUnitProperties.TARGET_SERVER, TargetServer.OC4J_10_1_3);
```

Valeur par défaut : None (Aucun)

Propriétés EclipseLink pour un schéma

Vous pouvez définir les propriétés d'unité de persistance EclipseLink JPA dans un fichier `persistence.xml` pour configurer la génération de schémas.

`eclipselink.ddl-generation`

Permet de spécifier l'action de génération DDL (Data Definition Language) à utiliser avec vos entités JPA. Pour spécifier la cible de génération DDL, consultez **`eclipselink.ddl-generation.output-mode`**.

Valeurs :

Un fichier `persistence.xml` peut contenir les valeurs suivantes :

- **none (aucun)** : EclipseLink ne génère pas de fichier DDL ; aucun schéma ne sera généré.
- **create-tables** : EclipseLink tente d'exécuter la commande `CREATE TABLE SQL` pour chaque table. Lorsque vous émettez une commande `CREATE TABLE SQL` pour une table existante, EclipseLink applique le comportement par défaut de votre combinaison spécifique de base de données et pilote JDBC. Dans la plupart des cas, une exception est renvoyée, la table n'est pas créée et EclipseLink traite l'instruction suivante.
- **drop-and-create-tables** : EclipseLink tente de supprimer toutes les tables, puis de les créer. En cas de problème, EclipseLink applique le comportement par défaut de votre combinaison spécifique de base de données et pilote JDBC. EclipseLink passera alors à l'instruction suivante.

Les valeurs suivantes sont valides pour

`org.eclipse.persistence.config.PersistenceUnitProperties` :

- **NONE**
- **CREATE_ONLY**
- **DROP_AND_CREATE**

Si vous utilisez la persistance dans un environnement Java SE et que vous voulez créer les fichiers DDL sans créer de tables, définissez une propriété système Java **`INTERACT_WITH_DB`** et définissez sa valeur sur **`False`**.

Valeur par défaut : une des valeurs suivantes.

- Aucun
- `PersistenceUnitProperties.NONE`

Exemple :

```
persistence.xml file<property value="create-tables"/>Example: property
Mapimport
```

```
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put(Persist
enceUnitProperties.DDL_GENERATION, PersistenceUnitProperties.CREATE_ONLY);
```

`eclipselink.application-location`

Permet de spécifier l'emplacement d'écriture par EclipseLink des fichiers DDL générés. Les fichiers seront écrits si la propriété **eclipselink.ddl-generation** est définie sur une valeur autre que **none (aucun)**.

Value (Valeur) : spécification de fichier vers un répertoire pour lequel vous disposez de droits d'accès en écriture. La spécification de fichier peut être une valeur relative à votre répertoire de travail actuel ou une valeur absolue. Si elle ne se termine pas par un séparateur de fichiers, EclipseLink ajoutera un séparateur valide pour votre système d'exploitation.

Valeur par défaut : une des valeurs suivantes.

"."+File.separator

ou

<tt>PersistenceUnitProperties.DEFAULT_APP_LOCATION</tt>

Exemple :

```
persistence.xml file<property value="C:\ddl\"/>Example: property Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put(Persist
enceUnitProperties.APP_LOCATION, "C:\ddl\");
```

eclipselink.create-ddl-jdbc-file-name

Spécifiez le nom du fichier DDL généré par EclipseLink et qui contient des instructions SQL pour créer des tables destinées à des entités JPA. Ce fichier est écrit à l'emplacement spécifié par eclipselink.application-location lorsque la propriété eclipselink.ddl-generation est définie sur create-tables ou sur drop-and-create-tables.

Valeurs valides : un nom de fichier valide pour votre système d'exploitation. Vous pouvez également ajouter un chemin d'accès au fichier comme préfixe au nom de fichier, à condition que la concaténation eclipselink.application-location + eclipselink.create-ddl-jdbc-file-name soit une spécification de fichier valide pour votre système d'exploitation.

Exemple :

```
persistence.xml file<property value="create.sql"/> Example: property
Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
(PersistenceUnitProperties.CREATE_JDBC_DDL_FILE, "create.sql");
```

Valeur par défaut : createDDL.jdbc ou

PersistenceUnitProperties.DEFAULT_CREATE_JDBC_FILE_NAME

eclipselink.drop-ddl-jdbc-file-name

Spécifiez le nom du fichier DDL généré par EclipseLink et qui contient des instructions SQL pour supprimer des tables des entités JPA. Ce fichier est écrit à l'emplacement spécifié par eclipselink.application-location lorsque la propriété eclipselink.ddl-generation est définie sur drop-and-create-tables.

Valeurs valides : un nom de fichier valide pour votre système d'exploitation. Vous pouvez également ajouter un chemin d'accès au fichier comme préfixe au nom de fichier, pourvu que la concaténation `eclipselink.application-location + eclipselink.drop-ddl-jdbc-file-name` soit une spécification de fichier valide pour votre système d'exploitation.

Exemple :

```
persistence.xml file<property value="drop.sql"/>Example: property Mapimport
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put
t(PersistenceUnitProperties.DROP_JDBC_DDL_FILE, "drop.sql");
```

Valeur par défaut : `dropDDL.jdbc` ou
`PersistenceUnitProperties.DEFAULT_DROP_JDBC_FILE_NAME`

`eclipselink.ddl-generation.output-mode`

Pour spécifier la cible de génération de fichiers DDL, utilisez cette propriété.

Valeurs valides :

Les valeurs valides à utiliser dans le fichier **`persistence.xml`** sont les suivantes :

`both` (les deux actions)

Générez les fichiers SQL et exécutez-les sur la base de données.

Si **`eclipselink.ddl-generation`** est définie sur `create-tables`,
`eclipselink.create-ddl-jdbc-file-name` sera écrite dans
`eclipselink.application-location` et exécutée sur la base de données.

Si **`eclipselink.ddl-generation`** est définie sur `drop-and-create-tables`,
`eclipselink.create-ddl-jdbc-file-name` et **`eclipselink.drop-ddl-jdbc-file-name`**
seront écrites dans **`eclipselink.application-location`**. Les deux fichiers SQL sont
exécutés sur la base de données.

`database` (base de données)

Exécutez SQL sur la base de données uniquement (sans générer de fichiers
SQL).

`sql-script`

Générez les fichiers SQL uniquement (sans les exécuter sur la base de
données).

Si **`eclipselink.ddl-generation`** est définie sur `create-tables`,
`eclipselink.create-ddl-jdbc-file-name` sera écrite dans
`eclipselink.application-location`. La commande ne sera pas exécutée sur la
base de données.

Si **`eclipselink.ddl-generation`** est définie sur `drop-and-create-tables`,
`eclipselink.create-ddl-jdbc-file-name` et **`eclipselink.drop-ddl-jdbc-file-name`**
seront écrites dans **`eclipselink.application-location`**, sans être exécutées sur la
base de données. Les valeurs suivantes sont valides pour
`org.eclipse.persistence.config.PersistenceUnitProperties` :

- **`DDL_BOTH_GENERATION`** : consultez les deux propriétés.

- DDL_DATABASE_GENERATION : consultez la base de données.
- DDL_SQL_SCRIPT_GENERATION : consultez sql-script.

Exemple :

```
persistence.xml file<property value="database"/>Example: property Mapimport  
org.eclipse.persistence.config.PersistenceUnitProperties;propertiesMap.put  
(PersistenceUnitProperties.DDL_GENERATION_MODE,  
PersistenceUnitProperties.DDL_DATABASE_GENERATION);
```

Valeur par défaut du mode de conteneur ou Java EE : {{database}},

Remarque : Remplacez ce paramètre par des conteneurs avec prise en charge spécifique d'EclipseLink. Pour plus d'informations, reportez-vous à la documentation du conteneur. Mode d'amorçage ou Java SE : **both (les deux actions)** ou **PersistenceUnitProperties.DDL_BOTH_GENERATION**.

agent APPC

L'agent APPC permet de virtualiser des programmes de transaction APPC. L'agent est un programme Java basé sur l'API APPC CPI-C qui sert de proxy entre les programmes de client et serveur APPC. L'agent transmet les messages envoyés par un client APPC à un serveur APPC et vice versa.

Vous pouvez configurer l'agent pour virtualiser un programme de transaction défini par un nom de destination APPC CPI-C. Vous pouvez utiliser l'agent avec un enregistreur du VSE pour enregistrer les demandes et les réponses échangées entre les programmes de transactions client et serveur APPC. En mode d'enregistrement, il communique avec le client dynamique et le serveur et enregistre tous les messages échangés. L'agent crée un service virtuel APPC que vous pouvez déployer dans le VSE.

En mode de lecture, lorsque vous déployez le service virtuel APPC enregistré, le client dynamique continue de communiquer avec l'agent APPC. Les réponses proviennent du service virtuel APPC exécuté dans le VSE au lieu du serveur dynamique. Si l'enregistrement ou la lecture n'est pas en cours, l'agent passe en mode d'authentification directe. En mode d'authentification directe, l'agent transmet simplement les messages entre le client APPC et le serveur.

L'agent APPC induit les restrictions suivantes :

- L'agent APPC prend en charge un nom de destination pour chaque agent.
- Le protocole APPC ne prend pas en charge le mode d'appel dynamique et ni la réparation de modèle.

Pour utiliser le protocole APPC :

1. Installez et configurez l'agent APPC, à l'aide de la procédure décrite dans la section [Installation de l'agent APPC](#) (page 36).
2. Configurez SNA pour l'agent APPC à l'aide de la procédure de [configuration de l'agent APPC](#) (page 39).
3. Enregistrez un service virtuel APPC à l'aide du processus décrit dans la section Enregistrement d'images de service APPC.
4. Pour utiliser la lecture de service APPC, suivez la procédure standard de déploiement et d'exécution d'un service virtuel.

Les rubriques suivantes sont incluses.

[Informations sur le déploiement technique d'un agent APPC](#) (page 35)

[Installation de l'agent APPC](#) (page 36)

[Configuration de l'agent APPC](#) (page 39)

Informations sur le déploiement technique d'un agent APPC

Conditions préalables

Serveur IBM de communication

Ressources système

Mémoire

512 Mo minimum de mémoire RAM disponible

UC

La [configuration requise relative à UC recommandée par IBM](#) est adéquate pour l'exécution de l'agent APPC.

Espace de stockage sur le disque dur

L'agent APPC génère des journaux de système de fichiers. 500 Mo minimum d'espace disque sont nécessaires pour assurer un espace suffisant pour les journaux dans le temps. Supprimez les archives de journal tous les 30 à 90 jours.

Version de Java

L'agent APPC requiert le SDK d'Oracle 1.7.0 ou version ultérieure.

Système d'exploitation

L'agent APPC est pris en charge sur toutes les plates-formes Linux prises en charge par IBM pour le serveur de communication.

Réseau

- L'agent APPC est lié au port 9000 (par défaut) pour la communication interne de DevTest.
- L'agent APPC communique avec le registre DevTest (port par défaut : 2010).
- L'agent APPC communique avec l'enregistreur du VSE exécuté dans la station de travail DevTest Workstation via AMQ, à travers le registre de DevTest (port par défaut : 2010).
- L'agent APPC communique avec le VSE via AMQ, à travers le registre de DevTest (port par défaut : 2010).

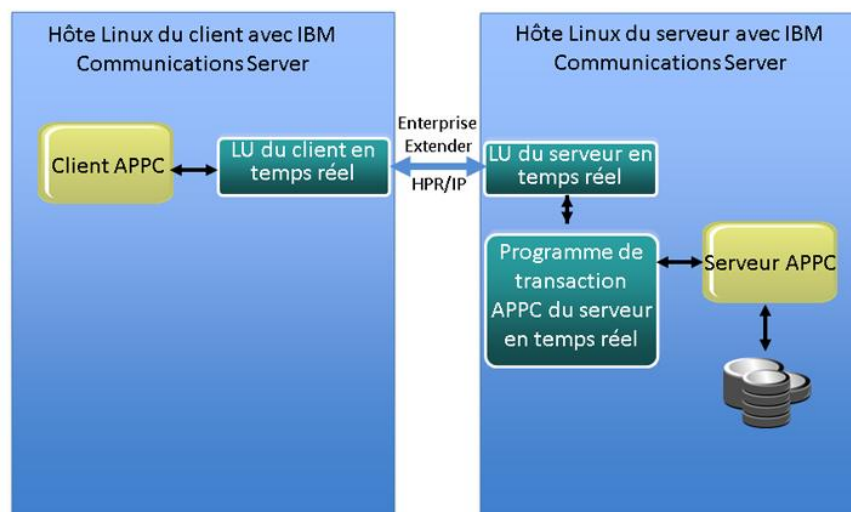
Installation de l'agent APPC

L'agent APPC est uniquement pris en charge sous UNIX et doit être installé sur un système sur lequel le serveur IBM Communications Server est exécuté. Le fonctionnement de l'agent requiert la bibliothèque APPC CPI-C.

L'agent s'exécute sur le client APPC, un serveur APPC, ou un système d'exploitation indépendant. L'installation sur le client APPC est recommandée, car il requiert une configuration moindre et permet de déconnecter complètement le serveur APPC pendant la lecture.

Le graphique suivant illustre l'architecture de l'agent APPC.

Exemple d'architecture de transaction APPC



Pour installer l'agent :

1. Créez un répertoire dans LISA_HOME appelé appc-agent.

L'agent est disponible dans le répertoire
LISA_HOME/addons/appc-agent/appc-agent.zip.

Remarque : L'agent APPC ne requiert pas l'installation de DevTest Solutions sur l'ordinateur cible. Vous pouvez copier le fichier .zip de l'agent à partir d'une installation de DevTest existante et l'installer sur l'ordinateur de votre choix.

2. Décompressez l'agent dans le répertoire appc-agent.

Une fois que le fichier **appc-agent.zip** extrait, le fichier **lisa-sna-appc.properties** sera disponible au niveau supérieur de LISA_HOME/appc-agent/.

3. Copiez le fichier cpic.jar dans le répertoire appc-agent/bin.
4. Pour configurer l'agent à l'aide de variables d'environnement, modifiez le script Shell **set-appc-env.sh** dans le répertoire appc-agent.

Les variables d'environnement dans le script Shell définissent les propriétés de configuration. Les propriétés de configuration remplacent toute propriété définie dans le fichier de propriétés. Si les propriétés **LIVE_DESTINATION**, **VSE_REGISTRY**, **VSE_LAB**, **APPC_AGENT_PORT**, ou **lisa.sna.appc.agent.registry.ping.timeout.millis** ne sont pas définies dans le script Shell, les valeurs du fichier lisa-sna-appc.properties seront utilisées.

Pour configurer l'agent à l'aide d'un fichier de propriété, modifiez **lisa-sna-appc.properties** dans le répertoire bin. Si le script Shell ne les fournit pas pour les champs, indiquez les paramètres à utiliser : **LIVE_DESTINATION**, **VSE_REGISTRY**, **VSE_LAB**, **APPC_AGENT_PORT** et **lisa.sna.appc.agent.registry.ping.timeout.millis**.

L'agent APPC utilise les variables d'environnement suivantes :

LD_LIBRARY_PATH

Spécifie le répertoire dans lequel la bibliothèque partagée SNA est disponible.

LD_PRELOAD

Spécifie les objets partagés de la bibliothèque SNA sur votre système.

LIVE_DESTINATION

Spécifie la destination du programme de transaction virtualisé par l'agent. Pour les exemples de programme inclus avec l'agent APPC, la destination dynamique est VSE_SAM. VSE_SAM est un nom de destination IPC-C pour les exemples.

VSE_REGISTRY=tcp://hôte:port/Registre

Définit l'URL du registre à laquelle l'environnement VSE et la station de travail DevTest Workstation se connectent. Saisissez l'adresse IP et le port sur lequel le registre DevTest est exécuté. Confirmez que le nom d'hôte peut être résolu et qu'il n'existe aucun problème de DNS.

VSE_LAB

Définit le nom du laboratoire pour l'installation de l'agent. La valeur par défaut convient aux installations standard.

Valeur par défaut : Default (Par défaut)

APPC_AGENT_PORT

Spécifie le port auquel le service d'agent est lié.

Valeur par défaut : 9000

lisa.sna.appc.agent.registry.ping.timeout.seconds

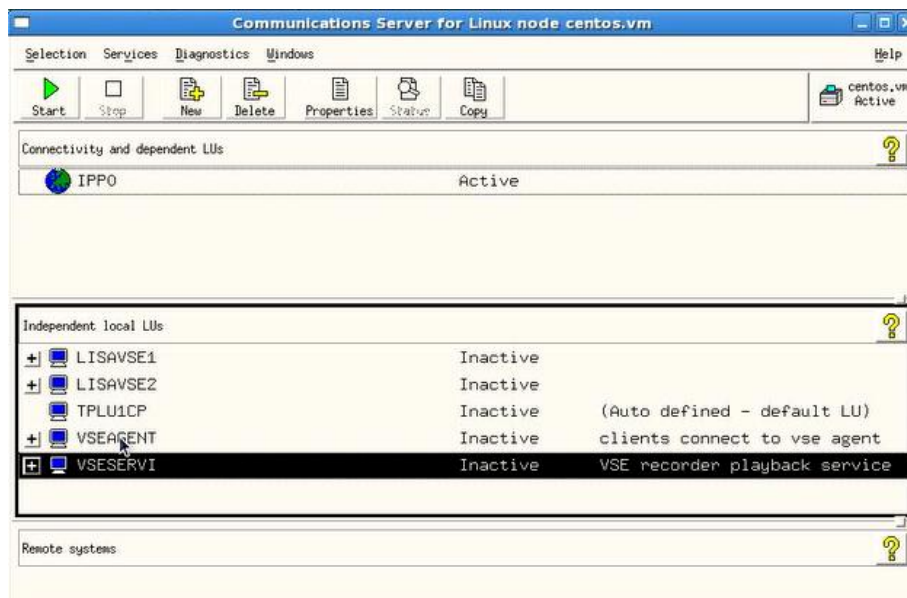
Définit le délai d'expiration (en secondes) après lequel l'agent n'attend plus la connexion au registre DevTest et continue de fonctionner en mode d'authentification directe.

Par défaut : 5

Configuration de l'agent APPC

Procédez comme suit:

1. Pour spécifier des options de journalisation, modifiez le fichier **appc-agent-logging.properties** dans le répertoire bin de l'agent APPC.
Pour plus d'informations sur la configuration des options de journalisation pour [log4j](#), consultez la documentation de log4j.
2. Configurez les propriétés SNA sur le système. Pour accéder au gestionnaire de communications SNA, saisissez *snaconf* dans la ligne de commande.
3. Configurez une paire d'unités logiques locale que vous pouvez utiliser pour configurer le programme de transactions de l'agent et la destination. Consultez l'administrateur SNA pour obtenir les valeurs appropriées pour votre système.



4. Sélectionnez Services, APPC, Transaction program (Programme de transaction), Properties (Propriétés).
5. Configurez le programme de transaction de sorte à pointer sur le répertoire d'accueil de l'agent `directory/bin/run-appc-agent.sh`.
6. Configurez les chemins de sortie standard (stdout) et d'erreur standard (stderr) vers quelques fichiers dans le répertoire de l'agent de sorte que tous les fichiers journaux soient disponibles au même emplacement.
7. Saisissez une description qui indique qu'il s'agit du programme de transactions de l'agent.

TP Invocation

TP name

Application TP

Service TP (hex)

LU

Parameters are for invocation on any LU ☐

Parameters are for invocation on a specific LU ☐

TP invocation

☒ Queue incoming Allocates

Full path to TP executable

Arguments

User ID

Group ID

stdin Path

stdout Path

stderr Path

Environment

Description

OK Cancel Help

8. Lorsque vous avez configuré le programme de transactions de l'agent, sélectionnez Services, APPC, IPC-C, Properties (Propriétés) pour définir une destination IPC-C.

The screenshot shows the 'CPI-C destination' dialog box with the following fields and options:

- Name:** VSEAGENT
- Local LU:**
 - ☒ Specify local LU alias: VSEAGENT
 - ☒ Use default LU
- Partner LU and mode:**
 - ☒ Use PLU alias
 - ☒ Use PLU full name: JTKO . VSESERVI
 - Mode:** LOCODE
- Partner TP:**
 - ☒ Application TP: VSE-AGENT-TP
 - ☒ Service TP (Hex)
- Security:**
 - ☒ None ☒ Same ☒ Program ☒ Program strong
 - User ID:**
 - Password:**
- Description:** VSE Agent Service Destination
- Buttons:** OK, Cancel, Help

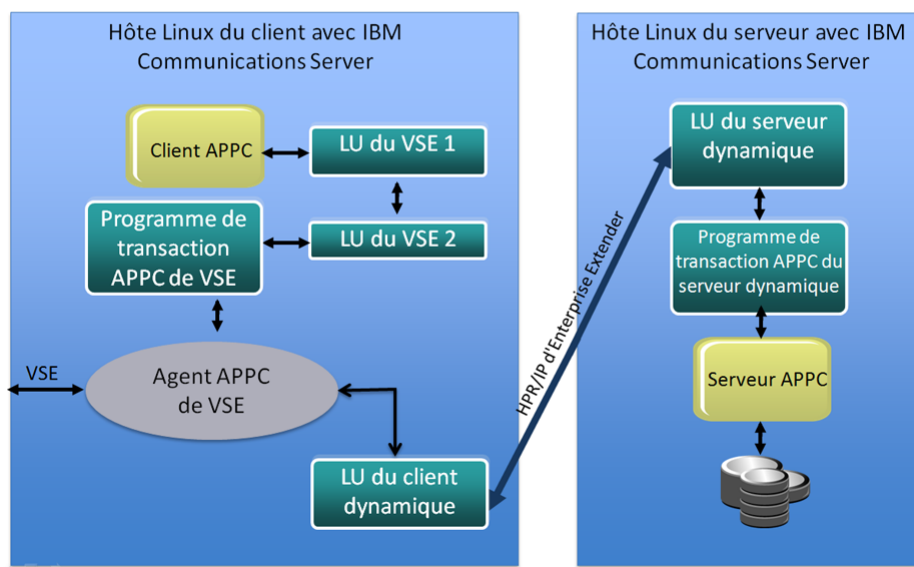
Une fois l'agent configuré, vous pouvez lancer l'enregistrement ou déployer un service virtuel enregistré dans le VSE. Pour pointer vers le nom de destination de l'agent, utilisez vos programmes clients.

Si votre programme client pointe vers un nom de destination dynamique, vous pouvez le basculer vers VSEAGENT. Selon le mode de transfert du nom de destination vers votre programme client, vous pouvez modifier des paramètres de ligne de commande client, un fichier de configuration, ou le code client.

Si un enregistrement a été lancé à l'aide de la station de travail DevTest Workstation, l'agent effectuera les enregistrements en mode d'enregistrement. Si le service est déployé en fonction de l'enregistrement par cet agent, il effectue une lecture. Si l'enregistrement ou la lecture n'est pas déployé, le service fonctionnera en mode d'authentification directe. En mode d'authentification directe, il communique avec la destination dynamique et envoie toutes les demandes et les réponses entre le client dynamique et le serveur dynamique.

Le graphique suivant illustre l'architecture de l'agent APPC avec le VSE.

Exemple d'architecture de transaction APPC avec VSE



Chapitre 3: Introduction à CA Service Virtualization

Ce chapitre traite des sujets suivants :

[Utilisation de CA Service Virtualization](#) (page 43)

[Composants de CA Service Virtualization](#) (page 44)

[Modèles de service virtuel](#) (page 45)

[Images de service](#) (page 46)

[Fonctionnement de la virtualisation](#) (page 48)

[Chaînes et dates magiques](#) (page 51)

[Introduction aux transactions de VSE](#) (page 58)

[Tolérance de correspondance](#) (page 63)

[Suivi des transactions](#) (page 66)

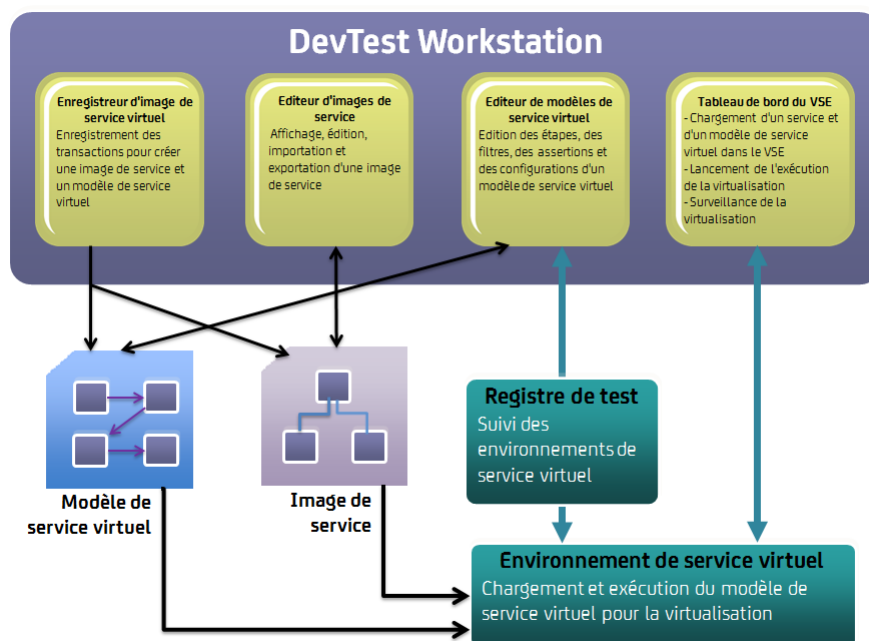
Utilisation de CA Service Virtualization

Le processus général du fonctionnement de CA Service Virtualization comprend les étapes suivantes :

1. Démarrez l'enregistreur d'image de service virtuel.
2. Dans l'enregistreur d'image de service virtuel, fournissez les informations de base sur les éléments à enregistrer. Sélectionnez les protocoles appropriés. Fournissez toutes les informations requises par les protocoles.
3. Dans l'enregistreur d'image de service virtuel, démarrez l'enregistrement.
4. Utilisez la communication cliente avec le serveur routé via le VSE.
VSE enregistre le trafic.
5. Dans l'enregistreur d'image de service virtuel, terminez l'enregistrement.
6. Dans la console de VSE, déployez le modèle de service virtuel et démarrez le service virtuel.
7. Exécutez les demandes dynamiques au niveau de CA Service Virtualization.

Composants de CA Service Virtualization

Le diagramme suivant présente la relation entre les composants dans CA Service Virtualization.



L'enregistreur d'image de service virtuel crée une image de service et un modèle de service virtuel. L'éditeur d'images de service et l'éditeur de modèles de service virtuel permettent de les afficher et de les modifier.

Lors de la virtualisation, le modèle de service virtuel et l'image de service sont chargés et exécutés dans un VSE exécuté en tant que service. Le service de registre de test est utilisé pour suivre un ou plusieurs VSE exécutés et la station de travail DevTest Workstation utilise ce service pour se connecter au VSE. Le tableau de bord du VSE est l'interface utilisateur Web utilisée pour surveiller et contrôler les modèles de service virtuel et les images de service chargés dans les VSE.

Modèles de service virtuel

Vous pouvez conceptualiser un modèle de service virtuel sous forme d'une série d'étapes à exécuter lorsqu'une demande est reçue. Les étapes du modèle de service virtuel créent et renvoient une réponse à la demande. Un modèle de service virtuel est stocké dans un fichier .vsm.

Pour être déployé dans un VSE, un modèle de service virtuel doit contenir au moins une étape du VSE figurant dans la liste d'étapes de l'environnement de service virtuel dans DevTest Workstation.

Une fois que vous avez enregistré une image de service, le VSE génère automatiquement les étapes de protocole dans le modèle de service virtuel. Vous pouvez modifier les étapes générées. Vous pouvez également effectuer les opérations suivantes :

- Remplir les réponses à partir d'une feuille de calcul Excel ou à l'aide de références croisées avec une table de base de données et de calculs dans des entrées
- Ajouter des étapes de types différents
- Manipuler les étapes de demande et de réponse avant de continuer
- Spécifier l'image de service à utiliser à partir de l'étape de sélection d'une réponse

Après la virtualisation, un modèle de service virtuel doit être déployé sur le VSE. Le modèle de service virtuel définit l'utilisation des modèles de comportement et interroge l'image de service pour déterminer le mode de réponse. Le modèle de service virtuel peut parcourir l'image de service. En général, les modèles de service virtuel peuvent être déployés uniquement vers des VSE, tandis que les scénarios de test peuvent être simulés sur des coordinateurs, mais pas sur des VSE.

Images de service

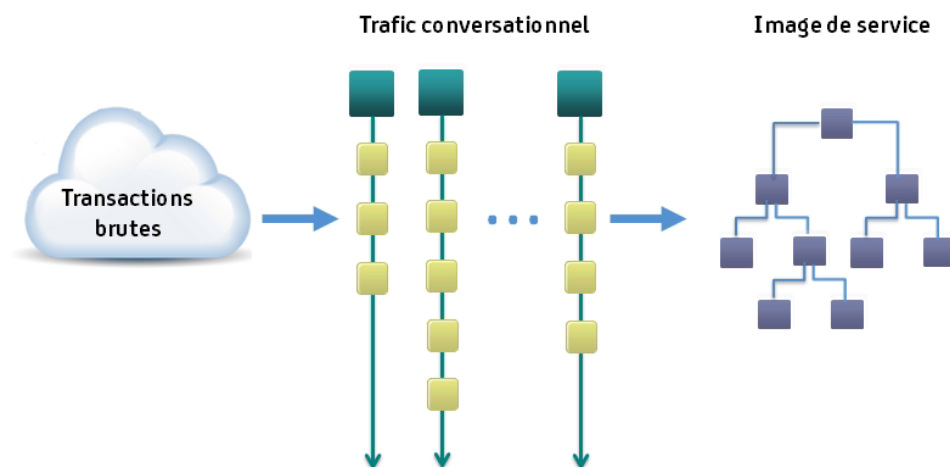
Une *image de service* est l'enregistrement de l'interaction entre le client et le serveur telle que créée par CA Service Virtualization. Un modèle de service virtuel référence une image de service. Après son enregistrement, l'image de service est utilisée pour envoyer la réponse appropriée au client en l'absence du serveur.

Une image de service contient les ensembles d'informations suivants :

- Une liste de conversations (demandes et réponses) enregistrée sous forme d'arborescence de conversations
- Une liste de transactions sans état (demandes et réponses)
- Les réponses à envoyer en cas de demandes conversationnelles ou sans état inconnues

Vous pouvez afficher, modifier ou créer une image de service à l'aide de l'éditeur d'images de service.

Vous pouvez afficher une conversation sous forme d'une série de transactions avec état. Toutefois, vous pouvez enregistrer plusieurs conversations (à partir de plusieurs sessions) dans la même image de service. Les structures de demande similaires sont fusionnées en une seule transaction, créant ainsi une arborescence comme dans le graphique suivant.



Par exemple, si plusieurs utilisateurs se connectent au système à l'aide de transactions **login()**, elles seront toutes fusionnées en une transaction unique. Cependant, si un utilisateur se connecte à l'aide d'une transaction **login()** et qu'un autre utilisateur se connecte avec une transaction **acquireAuthToken()**, ces transactions ne seront pas fusionnées.

Importation de transactions

Vous pouvez importer les documents XML suivants dans une image de service en cours d'enregistrement :

Transactions brutes

Le document XML représente le trafic brut comme s'il provenait du réseau, caractérisé par l'élément racine de <rawTraffic>.

Pour obtenir un exemple, consultez le fichier
LISA_HOME\examples\VServices\raw-traffic.xml.

Trafic conversationnel

Le document XML représente le trafic réorganisé en conversations, en ensembles de transaction sans état, ou les deux. Le trafic est organisé sous forme de listes linéaires de transactions, chacune d'elles représentant une conversation réelle et contenant un élément racine de <traffic>.

Pour obtenir un exemple, consultez le fichier
LISA_HOME\examples\VServices\traffic.xml.

Fonctionnement de la virtualisation

En l'absence de serveur, CA Service Virtualization simule le comportement du serveur pour son client. Ce processus est appelé virtualisation de serveur. Il requiert le chargement de l'image de service référencée à partir d'un modèle de service virtuel et son exécution dans le tableau de bord du VSE.

Lorsque VSE reçoit une demande, la demande est examinée et mise en correspondance avec un état conversationnel existant (session) dans le VSE. Par exemple, un ID de cookie ou un autre identificateur de session peut lier des demandes dans des protocoles sans état, tels que le protocole HTTP. Le VSE utilise l'état conversationnel pour déterminer l'emplacement de la transaction actuelle dans l'arborescence des conversations et tout autre état, par exemple un jeton d'authentification préalablement soumis, qui n'est pas nécessairement inclus dans la demande actuelle, mais est utilisé dans la réponse ultérieure. Le nom d'utilisateur est fourni à titre d'exemple.

Si aucune session n'existe, le VSE tente d'établir une correspondance entre la demande et les transactions de démarrage de chaque conversation dans l'image. Si une correspondance est renvoyée, une session est créée, puis une réponse pertinente est renvoyée. La session est conservée pendant 2 minutes après le dernier affichage par le VSE. Vous pouvez modifier ce comportement à l'aide de la propriété **`lisa.vse.session.timeout.ms`**. Si aucun démarrage de conversations ne correspond, aucune session ne sera créée et la liste de transactions sans état sera consultée par ordre de définition. En cas de correspondance, la réponse appropriée sera renvoyée. Sinon, la réponse Unknown request (Demande inconnue) sera envoyée.

Lorsque le VSE détecte une session précédente au cours d'une conversation, la correspondance de transaction suivante dépend de plusieurs facteurs. Ces facteurs peuvent inclure des tolérances de navigation et de correspondance.

Si aucune transaction correspondante dans les transactions conversationnelles et sans état n'est trouvée, le VSE consulte l'image de service à nouveau afin de déterminer le type de réponse à envoyer pour une demande conversationnelle ou sans état sans correspondance.

Informations complémentaires :


[Procédure de gestion des demandes conversationnelles](#) (page 49)

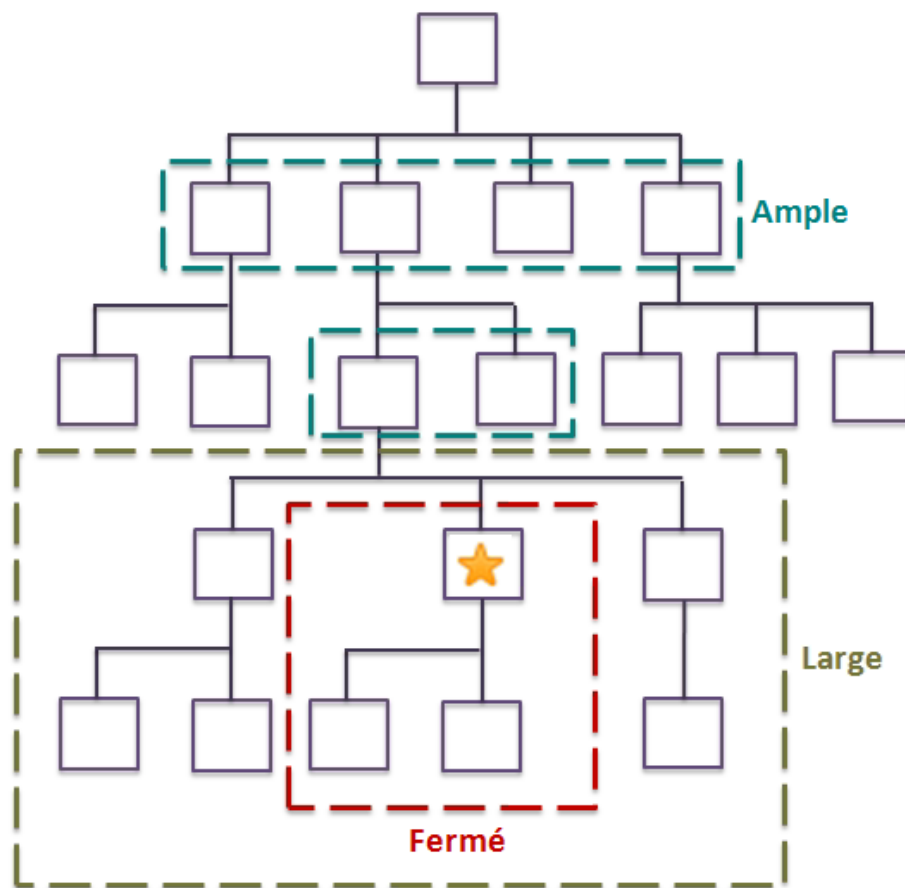
Procédure de gestion des demandes conversationnelles

La tolérance de navigation que vous pouvez spécifier pour tous les noeuds dans l'arborescence joue un rôle important dans la méthode de gestion d'une demande conversationnelle par le modèle de service virtuel. La tolérance de navigation est utilisée pour déterminer l'emplacement dans l'arborescence des conversations dans lequel un modèle de service virtuel recherche une transaction correspondant à la transaction spécifique.

Niveaux de tolérance de navigation

- **Close** : recherche des enfants de la transaction actuelle
- **Wide** : recherche Close comprenant la transaction actuelle, les frères et les nièces/neveux des transactions actuelles
- **Loose** : recherche Wide incluant le parent et les frères de la transaction actuelle, puis les enfants de la transaction de démarrage Si aucune des recherches ne trouve de correspondance, les transactions de démarrage de toutes les conversations seront vérifiées, ce qui revient à effectuer une recherche dans l'ensemble de la conversation.

Le graphique suivant illustre l'impact de la tolérance de navigation sur les transactions à rechercher dans une arborescence des conversations. Une étoile  indique la transaction actuelle.



Lors de l'enregistrement, l'enregistreur du VSE permet d'initialiser la tolérance de navigation sur des transactions à l'aide des paramètres suivants :

Default navigation (Navigation par défaut)

Définit la tolérance par défaut pour toutes les métatransactions disposant de métatransactions enfants.

Par défaut : Wide

Last (Dernier)

Définit la tolérance par défaut pour les métatransactions qui constituent des transactions terminales sans métatransaction enfant.

Par défaut : Loose

Vous pouvez changer ces paramètres ultérieurement pour chaque nœud via l'éditeur d'images de service dans la station de travail DevTest Workstation.

Les valeurs par défaut permettent d'obtenir de meilleurs résultats du comportement correct. VSE répond plus souvent de manière correcte dans des situations dans lesquelles les sessions d'exécution en cours redémarrent une conversation sans avoir à en démarrer une nouvelle.

Gestion des demandes inconnues

Une demande inconnue se produit dans les situations suivantes :

- En cas de conversation active
- En cas d'absence de conversation active

En cas d'absence de conversation active, une demande sera identifiée comme inconnue si aucune transaction sans état ne peut y répondre. Dans ce cas, la réponse de l'image de service pour les demandes sans état inconnues deviendra la réponse.

Lorsqu'aucune correspondance n'est établie entre une demande et une transaction de suivi :

- Si la tolérance de navigation n'est pas définie sur CLOSE, les transactions de démarrage de conversations auront la possibilité de répondre à la demande.
- Si une correspondance avec la demande n'est toujours pas établie alors qu'une transaction sans état peut produire une réponse, celle-ci sera envoyée. La session actuelle garde en mémoire son emplacement dans son arborescence des conversations.

En cas d'échec, la réponse de l'image de service pour les demandes conversationnelles inconnues deviendra la réponse.

Chaînes et dates magiques

Les chaînes et dates magiques sont essentielles à la nature dynamique d'un service virtualisé. Elles permettent au service virtualisé de renvoyer des résultats explicites pour des paramètres de demande jamais enregistrés.

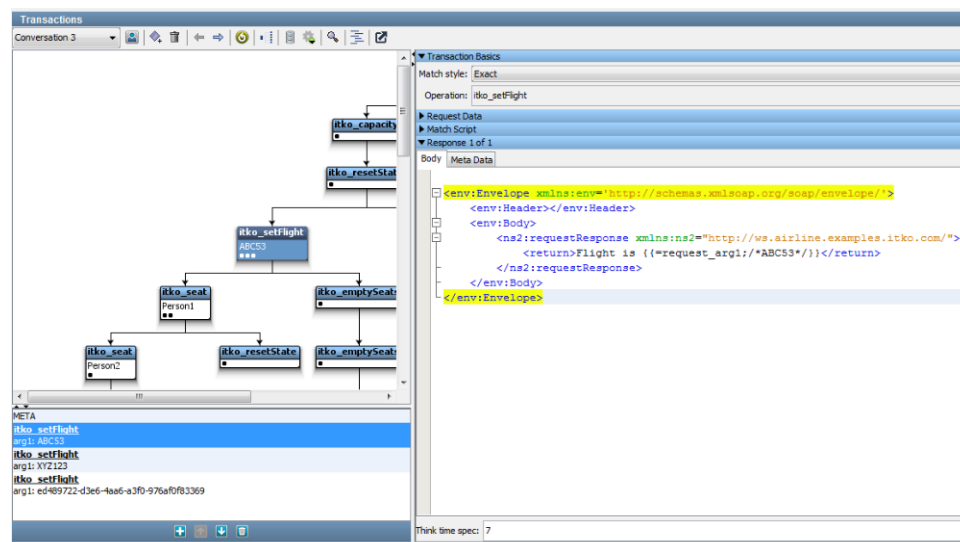
- [Chaînes magiques](#) (page 52)
- [Dates magiques](#) (page 57)

Chaînes magiques

Pendant la phase d'enregistrement, VSE recherche les arguments ou les paramètres pour chaque demande. Par exemple, un appel de service Web de prévisions météorologiques peut inclure un nom de ville, ou une demande de système de réservation d'une compagnie aérienne peut inclure un numéro de vol. Si le numéro de vol est inclus dans la réponse enregistrée, il sera classé comme *chaîne magique*.

Par exemple, si le VSE est en mode de lecture et une demande concernant un numéro de vol jamais enregistré est reçue, le numéro de vol correct est inclus dans la réponse.

Dans l'exemple suivant, une demande a été enregistrée pour définir un état dans la conversation ; dans le cas présent, il s'agit du numéro de vol (ABC53). L'enregistreur a reconnu que la réponse contenait la même chaîne ABC53 et l'a donc convertie en chaîne magique. La chaîne magique est enregistrée dans la base de données de l'image de service comme suit : '=request_arg1;/ABC53/.



La notation `{{ }}` représente une syntaxe de remplacement de propriété standard de VSE. Cette notation signifie remplacer le texte entre `{{ }}` par la valeur d'exécution du premier argument de la demande. Si aucun premier argument n'existe, utilisez ABC53 comme valeur par défaut.

L'effet pratique réside dans le fait que si un futur client demande le vol ZZ99, le service virtualisé répondra par le numéro de vol correct : ZZ99.

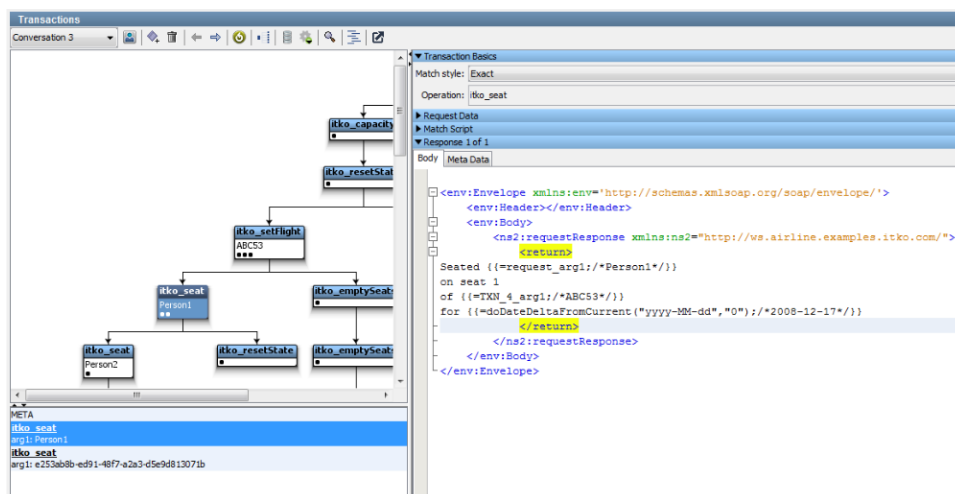
Le VSE ne détecte pas uniquement les chaînes magiques par une simple paire demande-réponse. Si un argument est détecté dans une réponse ultérieure dans une conversation, VSE détermine que la chaîne est magique et stocke la valeur de l'argument dans l'état de conversation.

Par défaut, le VSE ne prend en compte aucun élément dans une balise XML de chaînes magiques. Cette exclusion inclut les balises XML, les noms d'attribut et les valeurs d'attribut. Pour modifier ce comportement, définissez **lisa.magic.string.xml.tags=true** dans le fichier **local.properties**, auquel cas les règles de chaîne magique normales s'appliqueront.

En d'autres termes, si vous disposez de la structure **<foo bar="baz"/>**, aucun élément ne sera pris en compte pour des chaînes magiques. En revanche, si la structure est **<foo>bar</foo>**, bar (mais non foo) sera pris en compte pour des chaînes magiques.

L'exemple suivant se trouve ultérieurement dans la conversation précédente.

L'opération de demande est **itko_seat** et l'argument unique est un nom ; dans ce cas : **Person1**. La réponse contient le nom dans la demande, le numéro de vol préalablement défini et une date.



Il s'agit de l'exemple canonique de l'état conversationnel via un protocole sans état, dans ce cas SOAP via HTTP. Si le VSE en mode de lecture détecte une demande de définition du vol sur ZZ99, puis une demande pour le siège **PersonSomeoneElse**, la chaîne correcte **Seated PersonSomeoneElse on seat 1 of flight ZZ99** (**PersonSomeoneElse** assis sur le siège 1 du vol ZZ99) sera incluse dans la réponse, même si ces détails n'ont jamais été enregistrés.

Le VSE ne considère pas le numéro du siège, 1 dans ce cas, comme une chaîne magique. Le numéro de siège n'a jamais été détecté dans la série de demandes d'origine ayant entraîné cette réponse. Il n'est également pas assez élevé pour être considéré comme une chaîne magique : il doit comprendre un minimum de trois caractères et contenir un espace facultatif à gauche, à droite ou des deux côtés de la chaîne. Vous pouvez ajuster les paramètres de longueur et d'espace dans le fichier **local.properties**.

La notation de propriété `{{ }}` est puissante et flexible et n'est pas limitée à l'utilisation de chaînes magiques. Par exemple, `1` peut être remplacée dans la réponse de l'exemple par `DataSetValue` et un ensemble de données peut être défini dans le modèle de service virtuel. Cet ensemble de données est utilisé pour générer la valeur de réponse d'exécution. L'ensemble de données peut être un générateur de chaînes aléatoires, un compteur, un appel à une base de données, une référence à la ligne d'une feuille de calcul Excel, ou un autre élément. `}}` peut exécuter le code Java arbitraire et même générer des données quotidiennes réalistes, par exemple un numéro de carte de crédit valide `{{=:Credit Card:}}`.

Pour plus d'informations, consultez la section Utilisation de BeanShell dans DevTest de la rubrique *Utilisation*.

Exclusions de chaîne magique

Le VSE tente d'identifier des jetons dans des demandes et des réponses identiques (selon certaines règles) et de convertir les valeurs dans la réponse en chaîne magique. La valeur de la chaîne magique varie en fonction des valeurs dans la demande.

Toutefois, DevTest ne dispose d'aucun moyen pour savoir si ces valeurs correspondent en raison de leur conception, ou par hasard. Ce type de correspondance est très commun pour les valeurs suivantes :

- Booléens
- Jeton **__NULL** utilisé par l'agent DevTest et CAI pour le VSE Java. Par exemple :

Request (Demande)

```
<GetUserRequest>
<userId>lisaitko</userId>
<includeDetails>true</includeDetails>
</GetUserRequest>
```

Response (Réponse)

```
<GetUserResponse>
<userId>lisaitko</userId>
<isActive>true</isActive>
<isEmailVerified>true</isEmailVerified>
</GetUserResponse>
```

Les deux valeurs True dans la réponse n'ont aucun lien, ni avec la valeur incluse dans la demande. Dans des scénarios réels, le nombre de chaînes magiques inutiles générées est bien supérieur. Pour éviter ce problème, une méthode consiste à rechercher et remplacer ces chaînes magiques manuellement après l'enregistrement de l'image de service.

Il est toutefois plus simple de définir la propriété **lisa.magic.string.exclusion** dans **lisa.properties**.

Cette propriété permet de spécifier des valeurs non candidates à l'identification de chaîne magique. DevTest ne tentera pas de corréler ces valeurs dans la réponse avec des valeurs dans la demande pendant l'enregistrement. Si nécessaire, vous pouvez modifier manuellement l'image de service et ajouter des chaînes magiques.

Respect de la casse dans les chaînes magiques

Une *chaîne magique* est une chaîne qui se trouve dans un argument de demande, inclus ultérieurement dans une réponse pour cette conversation. Lorsque vous recherchez des chaînes magiques, les correspondances ne respectent pas la casse. Le traitement de la casse des chaînes magiques est résumée dans la logique suivante :

- Si la demande contient le terme dallas et la réponse contient DALLAS lors de l'enregistrement, la valeur austin est obtenue pendant la lecture et la réponse est AUSTIN.
- Si la demande contient le terme DALLAS et la réponse contient dallas lors de l'enregistrement, la valeur AUSTIN est obtenue pendant la lecture et la réponse est austin.
- Si la demande contient le terme dallas et la réponse contient Dallas lors de l'enregistrement, la valeur austin est obtenue pendant la lecture et la réponse est Austin.
- Si la demande contient le terme dallas et la réponse contient DaLLas lors de l'enregistrement, et la valeur austin est obtenue pendant la lecture, la réponse est austin, et non AuSTin. Si vous devez vous occuper explicitement de cette réponse en casse mixte dans votre image de service, modifiez la réponse pour que la méta-transaction inclue un rappel du code Java personnalisé, comme par exemple :

```
{%=MyHelperClass.mixedCase(request_city);%}
```

Dates magiques

Un analyseur de date puissant analyse les demandes, et les réponses sont également analysées au cours de l'enregistrement. Une valeur correspondant à une définition étendue de formats de date est identifiée et convertie en *date magique*. Dans l'exemple affiché dans la rubrique [Chaînes magiques](#) (page 52), la date magique est la suivante :

```
{{ =doDateDeltaFromCurrent("yyyy-MM-dd", "0D"); /2008-12-17 }}
```

L'utilisation des accolades {{ }} est importante dans ce cas.

Lors de l'exécution, cette chaîne est traduite de la manière suivante : générer une date au format aaaa-MM-jj de 0 jour à partir de la date actuelle. C'est-à-dire, générez la date actuelle.

Si l'enregistrement d'origine a été effectué le 1er février 2009 et que la réponse contenait la date 2009-02-10, la chaîne de date magique sera :

```
=doDateDeltaFromCurrent("yyyy-MM-dd", "10D"); /2009-02-10/
```

L'élément **10D** dans la chaîne magique signifie que le VSE génère une date dans la réponse 10 jours avant la date actuelle. Ainsi, si le VSE est en mode de lecture le 12 juin 2010, la réponse contiendra la chaîne **2010-06-22**.

Les paramètres valides pour les valeurs différentielles de date sont les suivants :

- **D** : jours
- **H** : heures
- **M** : minutes
- **S** : secondes
- **Ms** : millisecondes

Une autre variante des dates magiques est la suivante :

```
doDateDeltaFromRequest
```

Utilisez la variante **doDateDeltaFromRequest** lorsqu'une date sert de paramètre dans la demande et qu'une date est détectée dans la réponse. Par exemple, un système de réservation de compagnie aérienne peut accepter une demande de sièges pour un vol spécifique un jour particulier. Si cette date est détectée dans la réponse, le VSE remplace la date dans ses réponses ultérieures.

Prenons comme exemple plus significatif une demande de vol ayant généré une réponse détaillant un vol traversant la ligne de changement de date internationale. Un vol partant de Los Angeles vers Sydney arrive deux jours plus tard que la date de départ, selon la date du calendrier, même si la durée du vol est de 14 heures. Dans cet exemple, la réponse contient :

```
doDateDeltaFromRequest("yy-MM-dd", "2D")
```

Si le VSE traite une demande similaire pour un vol partant de Lax le 19 juin 2013, il inclura la date d'arrivée correcte du 21 juin 2013 dans la réponse.

Remarque : Vous pouvez ajouter un format de date/heure valide, y compris ceux contenant des fuseaux, dans **`lisa.properties`** pour effectuer des calculs de date magique.

Introduction aux transactions de VSE

Dans le contexte de CA Service Virtualization, une *transaction* est une unité de travail complète effectuée par un service. Une transaction inclut la demande que le client envoie au service et la réponse que le service envoie au client.

Pour la plupart des protocoles de service, y compris de services Web, HTTP et Java, les transactions sont réalisées de manière synchrone. La demande et la réponse sont directement liées l'une à l'autre.

Dans le cas des services Web et HTTP, la demande et la réponse sont contenues dans une connexion de socket unique et la réponse suit généralement la demande. Dans le cas de Java, la demande et la réponse sont contenues dans un thread unique. La réponse, c'est-à-dire la valeur renvoyée, suit toujours la demande, c'est-à-dire un appel de méthode.

La messagerie est différente, car elle est asynchrone. Les messages de demande et réponse ne doivent pas se produire dans la même connexion de socket, le même thread, ou même le même jour. Le client envoie un message de demande et effectue d'autres activités en attendant la réponse. Lorsque la réponse est envoyée à partir du service, le client la reçoit et la met en correspondance avec la réponse d'origine, pour terminer la transaction.

En général, une transaction comprend une seule réponse, comme c'est le cas d'un répondeur HTTP. Toutefois, le protocole de messagerie peut renvoyer plusieurs réponses pour une seule demande.

Transactions sans état et conversationnelles du VSE

Les transactions du VSE sont classées comme *stateless* (Sans état) ou *conversational* (Conversationnelle).

Stateless Transactions (Transactions sans état)

Les transactions sans état n'incluent aucune relation logique entre les transactions. Par exemple, HTTP et SOAP sont des protocoles sans état. Une transaction sans état contient toujours une réponse statique, indépendamment des appels précédents.

Dans une conversation sans état, chaque appel du service est indépendant des autres. Par exemple :

- Quel temps fait-il à Dallas, TX ? (Opération : temps ; arg1-ville)
- Quel temps fait-il à Atlanta, GA ? (Opération : temps ; arg1-ville)
- Quel temps fera-t-il à Dallas, TX demain ? (Opération : temps ; arg1-ville ; arg2-date)

Conversational/Stateful Transactions (Transactions conversationnelles/avec état)

Les conversations comprennent des transactions avec état. Les conversations comprennent la transaction logique qui, en cas de correspondance, démarre une session unique et les informations nécessaires à la création et l'identification de cette session. Une transaction dans une conversation dépend toujours du contexte créé par les conversations précédentes.

Voici un exemple d'une conversation avec état qui implique l'utilisation d'un guichet automatique.

1. Connectez-vous au guichet automatique. (Opération : connexion)
2. Quel compte voulez-vous utiliser ? (arg1-vérification)
3. Quel est mon solde ? (Opération : solde)
4. Réponse
5. Retirer une somme d'argent (Opération : retrait)
6. Combien ? (arg2-montant)
7. Quel est mon solde ? (Opération : solde)
8. Réponse (différente en fonction du montant retiré dans la demande précédente).
9. Terminer la session (Opération : déconnexion)

Démarreurs de conversations

La première transaction d'une conversation est appelée *démarreur de conversations*. Lorsque le VSE reçoit une demande entrante, il consulte les démarreurs de conversations pour déterminer si cette transaction signifie une nouvelle conversation.

Toutes les transactions d'une conversation étant liées les unes aux autres, le VSE doit pouvoir déterminer cette relation. En général, un type de jeton spécial est utilisé dans les transactions (par exemple, jsessionid), un cookie dans les transactions Web, ou un identificateur personnalisé dans le trafic de messages.

DevTest prend en charge les types de conversations suivants :

Conversations basées sur une instance

La couche de protocole est chargée d'identifier la chaîne unique, basée sur différentes instances du client. Cette chaîne est utilisée pour identifier des sessions côté serveur pour l'enregistrement et la lecture de l'image de service.

Conversations basées sur un jeton

Le VSE génère le jeton à l'aide d'un modèle de génération de chaînes, stocké avec la conversation dans l'image de service. Une fois le jeton généré, il fonctionne de la même manière qu'une conversation basée sur une instance. Vous ne pouvez pas déduire automatiquement des conversations basées sur le jeton pendant l'enregistrement. Pour spécifier l'emplacement des jetons, utilisez [l'enregistreur d'image de service virtuel](#) (page 302).

Tolérance de navigation

Dans une conversation, le VSE suit des règles spécifiques pour déterminer une méthode de détection de la transaction conversationnelle suivante. Les ensembles de règles, également appelées *tolérances de navigation*, se présentent comme suit :

Close

Les transactions doivent être situées au bas de l'arborescence. Les seuls candidats pour la transaction conversationnelle suivante sont les enfants de la transaction actuelle.

Wide

Tolérance par défaut. La tolérance Wide permet d'accéder à la transaction actuelle, aux enfants de la transaction actuelle, aux frères de la transaction actuelle et aux descendants immédiats des frères (neveux). L'ordre de priorité est le suivant :

1. Enfants d'une transaction actuelle
2. Enfants d'un frère
3. Frères d'une transaction actuelle

Loose

Il s'agit du niveau de tolérance de navigation le plus élevé. Le VSE applique d'abord les tolérances Close et Wide, puis ajoute la capacité de correspondance au parent de la transaction actuelle, aux frères du parent (oncles) et aux enfants des frères du parent (cousins). En cas d'échec de la mise en correspondance, la navigation est autorisée sur une transaction du deuxième ou troisième niveau de l'arborescence. L'ordre de priorité est le suivant :

1. Enfants d'une transaction actuelle (tolérance Close)
2. Enfants d'un frère (tolérance Wide)
3. Frère, ou transaction actuelle (tolérance Wide)
4. Frères de la transaction parente (oncles), mais non leurs enfants (cousins)
5. La transaction parente ou ses frères (parent ou oncles)
6. Enfants de la transaction de démarrage de la conversation actuelle (enfants immédiats de la racine de l'arborescence)
7. Transactions de démarrage pour toutes les conversations de l'image de service

Transactions logiques

Une transaction logique s'affiche sous forme de noeud unique dans une conversation. Une transaction logique comprend une métatransaction et une ou plusieurs transactions spécifiques.

Lorsqu'une transaction logique s'affiche dans la recherche de réponse à une demande particulière, la métatransaction est consultée. La métatransaction détermine si cette transaction veut répondre à la demande donnée. Si la métatransaction décide de répondre à la demande, toutes les transactions spécifiques devront indiquer si elles souhaitent répondre à la demande. Si aucune des transactions spécifiques ne veut répondre à la demande, la réponse sélectionnée sera celle de la métatransaction.

Cette logique peut être exprimée par le code fictif suivant :

```
pour chaque transaction logique \{  
  
    si (la demande de métatransaction correspond à la demande donnée) \{  
        // ce noeud gèrera la demande pour chaque transaction spécifique dans ce noeud  
        \{  
            si (la transaction spécifique correspond à la demande donnée) \{  
                la réponse est renvoyée à partir de la transaction spécifique  
            \}  
        \}  
        // Transaction spécifique introuvable pour la demande spécifique  
        la réponse est renvoyée à partir de la métatransaction  
    \}
```

Ultérieurement, une métatransaction physique contient une liste de transactions spécifiques et une liste de métatransactions enfants. Cette dernière permet de structurer les métatransactions en arborescence décisionnelle.

Tolérance de correspondance

La *Match Tolerance* (Tolérance de correspondance) définit la méthode utilisée par le VSE pour déterminer si une transaction donnée correspond à la transaction entrante.

Les niveaux de la tolérance de correspondance sont les suivants :

Operation (Opération)

Tolérance de correspondance la plus élevée. Le nom d'opération de la transaction entrante doit correspondre au nom de la transaction enregistrée.

Signature

Le nom d'opération doit correspondre et les noms des arguments doivent correspondre exactement, sans ajout ni suppression. L'ordre des arguments ne doit pas nécessairement être identique.

Exact

En plus de la correspondance de signature, les valeurs de chaque argument doivent correspondre aux valeurs enregistrées, telles que définies par les opérateurs de correspondance d'arguments.

Opérateurs de correspondance d'arguments

Tous les arguments d'une demande ont un *opérateur de correspondance*. Dans le cas d'une opération de correspondance Exact, il contrôle la mise en correspondance de la valeur de l'argument dans l'image de service avec la valeur de l'argument dans la demande entrante.

Les opérateurs de correspondance disponibles sont les suivants :

Anything (Tout type d'argument)

Renvoie toujours la valeur true. L'enregistreur de service virtuel définit par défaut la comparaison sur **Anything (Tout type d'argument)** lorsqu'il détermine qu'un argument est une date. La valeur entrante pour cet argument peut être une valeur quelconque. L'argument doit être présent pour que la correspondance entre la signature et les données META fonctionne, mais la valeur est ignorée et peut être vide ou nulle.

= Equal (Egal à)

Renvoie la valeur True si les valeurs sont identiques.

!= Not equal (Non égal à)

Renvoie la valeur True si les valeurs sont différentes.

<Less than (**Inférieur à**)

Renvoie la valeur True si la valeur entrante est inférieure ou antérieure à la valeur issue de l'image de service.

<= Less than or equal (**Inférieur ou égal à**)

Renvoie la valeur True si la valeur entrante est inférieure, antérieure ou égale à la valeur issue de l'image de service.

> Greater than (Supérieur à)

Renvoie la valeur True si la valeur entrante est supérieure ou ultérieure à la valeur issue de l'image de service.

>= Greater than or equal (Supérieur ou égal à)

Renvoie la valeur True si la valeur entrante est supérieure, ultérieure ou égale à la valeur issue de l'image de service.

Regular expression (Expression régulière)

Renvoie la valeur True si la valeur entrante (sous forme de chaîne) correspond à la valeur issue de l'image de service au format d'expression régulière.

Property Expression (Expression de propriété)

La valeur doit être au format d'une expression de script placée entre deux accolades {{ }}. Cette propriété ou expression est évaluée. Si le résultat commence par Y, y, T, t ou ON, l'argument renvoie une correspondance. La valeur d'argument dans la demande entrante est ignorée, sauf si elle est référencée dans le script.

Remarque : Si un argument est indiqué comme date, les valeurs issues des demandes à comparer seront converties en date avant la comparaison.

Métatransactions et réponses spécifiques

Lorsque le VSE recherche une correspondance de conversation, il recherche uniquement dans les métatransactions. La tolérance de correspondance d'une métatransaction ne peut pas être Exact (valeur par défaut : Signature). Chaque métatransaction contient une ou plusieurs réponses spécifiques, qui peuvent avoir la tolérance de correspondance de votre choix (valeur par défaut : Exact).

Si aucune des réponses spécifiques ne correspond à une métatransaction, la réponse spécifiée pour la métatransaction sera utilisée.

Mode de sélection de la réponse suivante

1. Si la demande entrante est contenue dans une conversation, recherchez une correspondance basée sur la tolérance de navigation et d'autres règles expliquées précédemment.
2. Si aucune correspondance n'est détectée dans la conversation actuelle, recherchez une autre conversation, sauf si la tolérance de navigation est définie sur Close.
3. Si une correspondance de conversation a été détectée et qu'une réponse spécifique a été obtenue (au lieu d'une simple correspondance de métatransaction), utilisez-la.
4. Dans le cas contraire, recherchez une correspondance spécifique dans les transactions sans état et utilisez-la (le cas échéant).
5. Si aucune correspondance spécifique n'a été détectée dans la liste de transactions sans état, mais qu'une correspondance de métatransaction a été détectée dans la liste des conversations, utilisez cette correspondance de métatransaction.
6. Si aucune correspondance conversationnelle, mais qu'une correspondance de métatransaction a été détectée dans la liste de transactions sans état, utilisez la correspondance de métatransaction.
7. Envoyez un échec avec le message No match found (correspondance introuvable) et renvoyez la réponse conversationnelle inconnue ou la réponse sans état inconnue spécifiée dans l'image de service. Vous pouvez remplacer la réponse en fonction du protocole et des gestionnaires utilisés.

Procédure de débogage des échecs de correspondance

Si les correspondances échouent, ouvrez le fichier **LISA_HOME\logging.properties** et définissez **log4j.logger.VSE** sur **DEBUG** (Déboguer) ou **TRACE** (Suivi). De cette façon, le VSE envoie des détails au fichier **vse_xxx.log**, où **xxx** correspond au nom de l'image de service. La propriété **log4j.logger.VSE** indique également les éléments correspondants et non correspondants de manière exacte.

Définissez la propriété **log4j.logger.VSE** sur **INFO** ou **WARN** pour un environnement de production. Ne conservez pas le niveau **DEBUG** ou **TRACE** plus longtemps que nécessaire.

Suivi des transactions

Dans le VSE, vous pouvez suivre les transactions réalisées à l'aide de la fonction ITR.

Lors de l'utilisation d'un service virtuel dans l'ITR, définissez le mode d'exécution dans le fichier de configuration utilisé par le projet.

Procédez comme suit:

1. Sélectionnez Actions, View tracked responses from the ITR (Afficher les réponses suivies à partir de l'exécution d'un test interactif).
2. Pour suivre les transactions d'un service virtuel spécifique, définissez le mode d'exécution dans le fichier de configuration actif en définissant la propriété **lisa.vse.execution.mode** sur **TRACK** (suivi).

Une nouvelle fenêtre dans l'éditeur répertorie les transactions suivies.

Chapitre 4: Utilisation du portail DevTest avec CA Service Virtualization

Ce chapitre traite des sujets suivants :

[Ouverture du portail DevTest](#) (page 67)

[Création d'un service virtuel](#) (page 69)

[Modification des services virtuels](#) (page 81)

[Déploiement d'un service virtuel](#) (page 103)

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 exécutés, 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).

Chapitre 5: Création d'un service virtuel

La fenêtre Create a Virtual Service (Créer un service virtuel) permet d'enregistrer un service virtuel.

Pour créer un service virtuel, entrez Recording Information (Informations de l'enregistrement) dans le volet supérieur de l'enregistreur CA Service Virtualization.


Procédez comme suit:

1. Saisissez le Recording Name (Nom de l'enregistrement). Pour afficher une liste de configurations d'enregistrement préalablement enregistrées, cliquez sur List. Pour supprimer des informations d'enregistrement sélectionnées à partir de l'enregistreur et effectuer un nouvel enregistrement, cliquez sur New Recording (Nouvel enregistrement).

Lorsque vous affichez une liste d'enregistrements, vous pouvez trier la liste par nom, ID, ou statut. Pour supprimer un enregistrement, sélectionnez-le et cliquez sur

Delete (Supprimer)  .

2. Sélectionnez VSE Server (Serveur VSE) dans la liste déroulante.
3. Saisissez une description du service virtuel.

4. Cliquez sur Txn (Transactions)  pour afficher les transactions pendant l'enregistrement du service virtuel.

5. Cliquez sur Status (Statut)  pour afficher les détails sur le statut de l'enregistrement.

Ce chapitre traite des sujets suivants :

[Enregistrement d'un site Web \(HTTP ou HTTP/S\)](#) (page 70)

[Configuration d'un service virtuel](#) (page 74)

[Enregistrement d'un service virtuel](#) (page 76)

Enregistrement d'un site Web (HTTP ou HTTP/S)

Pour enregistrer un service virtuel à partir d'un site Web HTTP/S, saisissez les informations sur le protocole de transport HTTP dans le volet inférieur de la fenêtre Create Virtual Service (Créer un service virtuel).

Procédez comme suit:

1. CA Service Virtualization a sélectionné un hôte par défaut pour l'enregistreur de service virtuel et affiche son adresse IP. Pour modifier le nom ou l'adresse IP de cet hôte sur lequel l'enregistreur est exécuté, sélectionnez une autre adresse IP à partir de la liste déroulante. S'il existe plusieurs adresses IP d'hôtes et que vous voulez effectuer l'enregistrement sur toutes ces adresses, l'option All (Toutes) est la première fournie dans la liste déroulante et est sélectionnée par défaut.
2. Pour entrer ou changer le port cible de l'enregistreur de service virtuel, cliquez sur l'icône en forme de verrou pour activer le champ de saisie. Entrez un numéro de port utilisé par l'enregistreur comme port d'écoute des demandes entrantes.
3. Pour spécifier le site Web à enregistrer, entrez l'URL dans le champ Enter Target URL (Saisir l'URL cible). Les formats suivants sont acceptés :

- <ip>:<port>
- <nom-hôte>:<port>
- <serveur>.<domaine>

Remarque : A mesure que vous saisissez l'URL cible dans ce champ, VSE valide le format. Une fois le champ complètement saisi et que vous quittez à l'aide de la tabulation, VSE effectue un appel au serveur cible pour vérifier son accessibilité. Un message d'erreur approprié est affiché.

4. Sélectionnez ou désélectionnez la case à cocher Do not modify host header parameter received from client (Ne pas modifier le paramètre d'en-tête de l'hôte reçu à partir du client).

Si elle est sélectionnée, cette option indique à l'appel dynamique d'envoyer l'en-tête d'hôte reçu à partir de l'application cliente au serveur cible. Si elle est désélectionnée, l'appel dynamique régénère le paramètre d'en-tête d'hôte de la manière suivante : **hôte: <hôte cible>:<port cible>**.

Remarque : A mesure que vous entrez des informations, observez la partie supérieure de la fenêtre et vérifiez si des messages d'erreur mettent en surbrillance des entrées non valides.

Remarque : Notez que le graphique de l'enregistrement change de couleur à mesure que les composants sont validés. Dans le graphique précédent, l'enregistreur de service virtuel et le serveur sont colorés, ce qui indique qu'ils ont été validés.

Le statut de l'enregistrement passe de Draft (Brouillon) à Ready (Prêt) lorsque toutes les informations obligatoires requises pour l'enregistrement sont saisies. Le statut apparaît sur le bouton bleu Status. Pour obtenir plus de détails sur l'enregistrement, cliquez sur Status (Statut).

5. Sélectionnez l'une des options suivantes :

- Pour enregistrer sans utiliser SSL, laissez les cases à cocher Use SSL (Utiliser une connexion SSL) désélectionnées. Cliquez sur Next (Suivant).
- Pour enregistrer à l'aide de SSL, sélectionnez une ou les deux cases à cocher Use SSL (Utiliser une connexion SSL). La case à cocher Use SSL (Utiliser une connexion SSL) à gauche permet de saisir des informations sur le certificat SSL à envoyer au client. La case à cocher Use SSL (Utiliser une connexion SSL) à droite permet de saisir des informations sur le certificat SSL à envoyer au serveur.

Remarque : Pour plus d'informations sur la connexion SSL, consultez la section [CA Service Virtualization et SSL](#) (page 73).

6. (Facultatif) Si vous sélectionnez une ou les deux cases à cocher Use SSL (Utiliser une connexion SSL), remplissez les champs suivants :

SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.

Pour rechercher un fichier de référentiel de clés sur le système de fichiers, cliquez sur Browse (Parcourir).

SSL Keystore password (Mot de passe du référentiel de clés SSL)

Indique le mot de passe associé au fichier de référentiel de clés spécifié.

SSL Key alias (Alias de la clé SSL)

Désigne un alias pour une clé publique.

SSL Key password (Mot de passe de la clé SSL)

Indique le mot de passe associé à l'alias dans le fichier de référentiel de clés spécifié.

7. Pour valider les paramètres SSL, cliquez sur Validate (Valider). A mesure que vous saisissez des données dans les champs SSL, le système effectue la validation de manière progressive.
8. Cliquez sur Start Recording (Lancer l'enregistrement).

Remarque : Si vous démarrez un enregistrement dans le panneau d'enregistrement, enregistrez quelques transactions, cliquez sur Next (Suivant), puis sur Prev (Précédent) pour revenir au panneau d'enregistrement, vous revenez aux étapes de configuration.

9. A l'issue de l'enregistrement, cliquez sur Next (Suivant).

CA Service Virtualization et SSL

Pendant l'enregistrement de <lvs>, l'application de serveur SSL est le système testé.

Utilisation de la connexion SSL (Entre l'enregistreur et le serveur cible)

Si la case à cocher Use SSL (Utiliser une connexion SSL) entre l'enregistreur et le serveur est sélectionnée, l'enregistreur communiquera avec le serveur cible via une connexion SSL. Par défaut, l'enregistreur approuve tous les certificats de serveur.

Si le serveur cible demande une authentification cliente (connexion SSL bidirectionnelle), l'enregistreur envoie le certificat spécifié dans les champs sous la case à cocher Use SSL (Utiliser une connexion SSL). Si aucune information de certificat n'est spécifiée, l'enregistreur utilisera les valeurs spécifiées dans le fichier local.properties (propriétés **ssl.client.cert.***) du serveur VSE sur lequel l'enregistrement est effectué.

Utilisation de la connexion SSL entre le client et l'enregistreur

Si la case à cocher **Use SSL** (Utiliser une connexion SSL) entre le client et l'enregistreur est sélectionnée, le client communiquera avec l'enregistreur via une connexion SSL. Dans cette partie de l'enregistrement, l'enregistreur est le délégué du serveur cible et doit renvoyer un certificat de serveur lorsque le client initialise une demande. Le certificat renvoyé au client est celui spécifié dans les champs sous la case à cocher Use SSL (Utiliser une connexion SSL).

Lecture du modèle de service virtuel

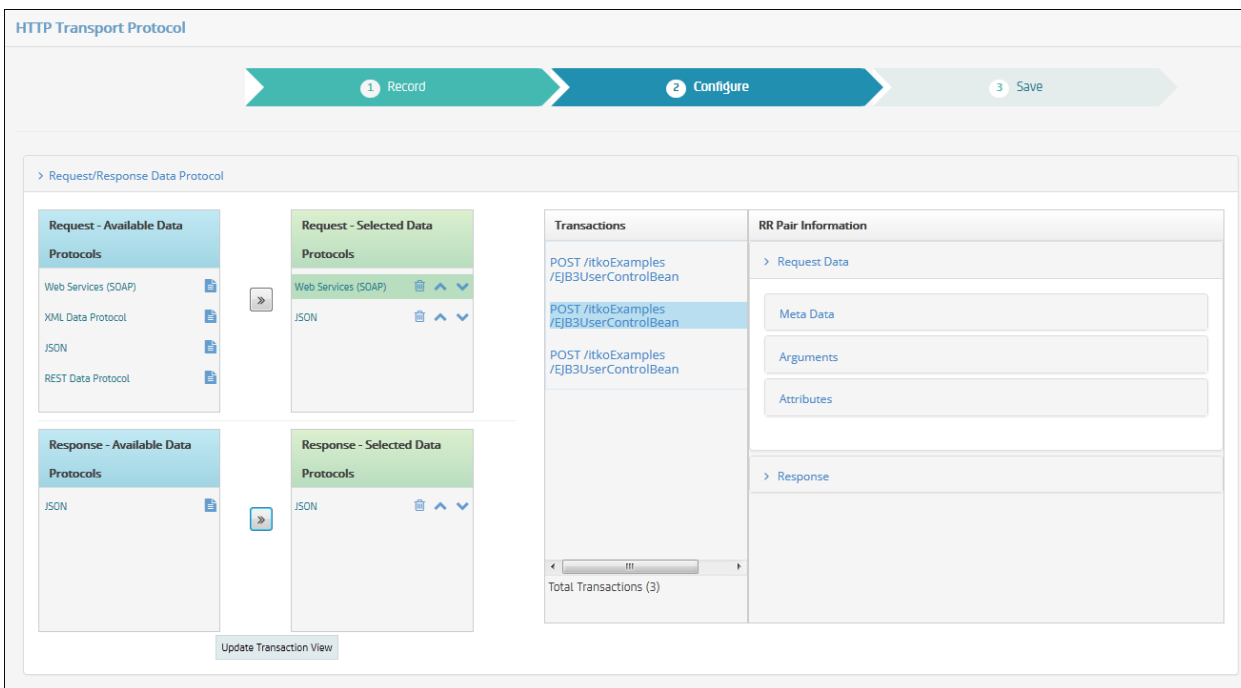
Si vous sélectionnez l'option Use SSL to Client (Utiliser une connexion SSL au client), un établissement de liaison SSL est effectué entre l'application cliente ou le client du scénario de test et le modèle de service virtuel. Le référentiel de clés fourni dans l'étape Listen (Ecouter) du modèle de service virtuel est utilisé comme certificat du serveur pour l'authentification unidirectionnelle. Il n'y a aucun établissement de liaison SSL entre le client et le modèle de service virtuel. L'établissement de liaison est une connexion HTTP simple.

Si l'étape Live Invocation (Appel dynamique) a été exécutée, un établissement de liaison SSL est effectué entre le client de modèle de service virtuel et le serveur réel. Si le serveur réel demande l'authentification du client, le référentiel de clés de l'étape HTTP/S Protocol Live Invocation (Appel dynamique via le protocole HTTP/S) est utilisé.

Si le référentiel de clés contient plusieurs certificats, CA Service Virtualization utilisera le premier.

Configuration d'un service virtuel

Le volet Configure (Configurer) permet d'accepter, d'ajouter, ou de supprimer des protocoles de données pour l'enregistrement réalisé. Vous pouvez également observer l'impact du protocole de données sur les transactions.



La colonne à gauche répertorie les protocoles de données disponibles côté demande et réponse. Les protocoles de données détectés automatiquement sont mis en surbrillance en vert et affichés dans la liste Selected (Sélectionnés). Tous les protocoles de données disponibles pour la demande et la réponse sont affichés.

Pour ajouter un protocole de données, mettez-le en surbrillance et cliquez sur Move



(Déplacer) pour déplacer le protocole de données dans la colonne Selected Data Protocols (Protocoles de données sélectionnés). Vous pouvez déplacer plusieurs protocoles de données dans la colonne de protocoles sélectionnés. Pour supprimer, déplacer vers le haut, ou vers le bas la sélection, utilisez les icônes situées au-dessus de la colonne.

Les transactions s'affichent dans le côté droit du volet, avec les données de demande et de réponse. Lorsque vous cliquez sur Update Transaction View (Mettre à jour la vue des transactions), le système applique ces protocoles de données aux transactions enregistrées. Les protocoles de données, une fois appliqués à une transaction, analysent les demandes et les réponses de manière différente. Pour plus d'informations, consultez la section Utilisation de protocoles de données dans la rubrique *Utilisation de CA Service Virtualization*.

Les transactions mises à jour, avec des demandes et des réponses analysées, sont récupérées à partir du serveur VSE et affichées dans l'aperçu du volet Transactions à droite du volet Configure (Configurer).

Remarque : Les protocoles de données sélectionnés s'appliquent et sont enregistrés pour le service virtuel uniquement lorsque vous cliquez sur Update Transaction View (Mettre à jour la vue des transactions). Les modifications apportées aux listes Available (Disponibles) et Selected (Sélectionnés) s'appliquent uniquement à l'interface utilisateur.

Enregistrement d'un service virtuel

Le volet Save (Enregistrer) de l'enregistreur CA Service Virtualization affiche les options suivantes :

- Enregistrer et fermer
- Save & Deploy Virtual Service (Enregistrer et déployer le service virtuel)
- Save & Edit Virtual Service (Enregistrer et modifier le service virtuel)

Pour chaque option, remplissez les champs suivants :

Select Project (Sélectionner un projet)

Détermine le projet dans lequel le service virtuel sera stocké. Sélectionnez un projet dans la liste déroulante. Les projets répertoriés dans la liste déroulante sont ceux disponibles dans le répertoire Projects.

Treat all transactions as stateless (Traiter toutes les transactions comme des transactions sans état)

Cette option est fournie pour des situations spécifiques dans lesquelles vous voulez traiter toutes les transactions enregistrées comme étant sans état. Dans la plupart des cas, laissez cette case à cocher désélectionnée.

Valeur par défaut : Sélectionné

Allow duplicate specific transactions (Permettre la duplication des transactions)

Spécifie si des transactions dupliquées spécifiques doivent être enregistrées.

Valeur par défaut : option désélectionnée

Pour afficher d'autres options, cliquez sur YES pour Advanced Mode (Mode avancé) dans le coin supérieur droit de la fenêtre. Vous pouvez également renseigner les champs suivants :

VSM style (Style de gestionnaire de pilotes)

Spécifie si un modèle de service virtuel, y compris les étapes de préparation, doit être généré.

Valeurs :

- **More Flexible** (Plus flexible) : inclut des étapes de préparation (donnant lieu à un modèle en cinq étapes).
- **More Efficient** (Plus efficace) : les étapes de préparation sont absentes (donnant lieu à un modèle en trois étapes).

Valeur par défaut : More Efficient (Plus efficace)

Default navigation (Navigation par défaut)

Spécifie la tolérance de navigation utilisée pour déterminer l'emplacement dans l'arborescence des conversations dans lequel un modèle de service virtuel recherche une transaction correspondant à la transaction spécifique. Sélectionnez la tolérance de navigation par défaut pour tous les éléments sauf les dernières transactions (terminales).

Valeurs :

- Close : recherche des enfants de la transaction actuelle
- Wide : recherche Close comprenant la transaction actuelle, les frères et les nièces/neveux des transactions actuelles
- Loose : recherche Wide incluant le parent et les frères de la transaction actuelle, puis les enfants de la transaction de démarrage Si aucune des recherches ne trouve de correspondance, les transactions de démarrage de toutes les conversations seront vérifiées, ce qui revient à effectuer une recherche dans l'ensemble de la conversation.

Par défaut : Wide

Leaf Navigation (Navigation de niveau feuille)

Spécifie la tolérance de navigation utilisée pour déterminer l'emplacement dans l'arborescence des conversations dans lequel un modèle de service virtuel recherche une transaction correspondant aux dernières transactions (feuille).

Valeurs :

- Close : recherche des enfants de la transaction actuelle
- Wide : recherche Close comprenant la transaction actuelle, les frères et les nièces/neveux des transactions actuelles
- Loose : recherche Wide incluant le parent et les frères de la transaction actuelle, puis les enfants de la transaction de démarrage Si aucune des recherches ne trouve de correspondance, les transactions de démarrage de toutes les conversations seront vérifiées, ce qui revient à effectuer une recherche dans l'ensemble de la conversation.

Par défaut : Loose

Save and Close (Enregistrer et fermer)

Save and Close (Enregistrer et fermer) permet d'enregistrer le service virtuel et de fermer l'onglet de l'enregistreur.

Save and Deploy Virtual Service (Enregistrer et déployer le service virtuel)

Save and Deploy Virtual Service (Enregistrer et déployer le service virtuel) permet d'enregistrer le service virtuel et de le déployer immédiatement.

Renseignez les champs suivants :

Balise group

Nom du [groupe de services virtuels](#) (page 433) pour le service virtuel. Si les services virtuels déployés ont des balises group, ces balises sont disponibles dans la liste déroulante. Une balise group doit commencer par un caractère alphanumérique et peut contenir des caractères alphanumériques et les caractères spéciaux suivants :

- Point (.)
- Tiret (-)
- Trait de soulignement (_)
- Symbole dollar (\$)

Concurrent capacity (Capacité simultanée)

Nombre qui indique la capacité de charge. La *capacité* est le nombre d'utilisateurs virtuels (instances) qui peuvent être exécutés simultanément avec le modèle de service virtuel. Dans cet exemple, la capacité indique le nombre de threads existants pour les demandes de service de ce modèle de service.

VSE alloue un nombre de threads équivalent à la capacité simultanée totale. Chaque thread consomme quelques ressources système, même lorsqu'ils sont inactifs. Par conséquent, pour optimiser les performances système globales, définissez ce paramètre aussi sur une valeur aussi faible que possible. Déterminez les paramètres corrects de manière empirique en les ajustant jusqu'à atteindre les performances souhaitées, ou jusqu'à leur augmentation n'entraîne plus d'amélioration des performances.

Les protocoles prédéfinis utilisent un service d'exécution de tâches de niveau structure pour réduire l'utilisation des threads. Dans le cas de ces protocoles, une capacité simultanée supérieure à 2-3 par noyau est rarement utile, sauf si le modèle de service virtuel a été considérablement personnalisé.

Pour des extensions et tout modèle de service virtuel qui n'utilise pas de protocole prédéfini, la définition d'un délai de réflexion long peut consommer un thread pour la durée de ce délai. Dans ces cas, vous pouvez augmenter la capacité simultanée.

La formule suivante donne un paramètre initial approximatif dans ces cas :

Capacité simultanée = (Transactions souhaitées par seconde / 1000) *
délai de réflexion moyen en ms * (pourcentage de délai de réflexion / 100)

Exemple :

Supposons que vous utilisez un protocole personnalisé qui n'utilise pas le service d'exécution de tâches de structure pour gérer des délais de réflexion. Vous voulez un débit global de 100 transactions par seconde. Le délai de réflexion moyen dans l'image de service est de 200 ms et le service virtuel est déployé avec un délai de réflexion de 100 %.

$$(100 \text{ transactions par seconde} / 1000) * 200 \text{ ms} * (100 / 100) = 20$$

Dans ce cas, les threads se bloquent pendant environ 200 ms en moyenne avant de répondre et durant ce temps, ils ne parviennent pas à gérer de nouvelles demandes. Nous avons par conséquent besoin d'une capacité de 20 pour contenir 100 transactions par seconde. Un thread deviendra disponible toutes les 10 ms en moyenne, ce qui devrait suffire à obtenir 100 transactions par seconde.

Valeur par défaut : 1

Think time scale (Pourcentage de délai de réflexion)

Indique le pourcentage de délai de réflexion enregistré.

Remarque : Une étape déduit sa propre durée de traitement du délai de réflexion pour obtenir une fréquence d'exécutions des tests cohérente.

Par défaut : 100

Exemples :

- Pour doubler le délai de réflexion, utilisez 200.
- Pour réduire de moitié le délai de réflexion, utilisez 50.

HTTP Transport Protocol

1 Record

2 Configure

3 Save

You have three options to choose to Save and Finish.

☐ Save & Close
 ☒ Save & Deploy Virtual Service
 ☐ Save & Edit Virtual Service

Select Project: Previous Project

☒ Treat all transactions as stateless
 ☐ Allow duplicate specific transactions

Options for Deploy Virtual Service

Group Tag:

Concurrent capacity:

Think Time Scale (%):

Save →

Enregistrement et modification

Save & Edit Virtual Service (Enregistrer et modifier le service virtuel) permet d'enregistrer le service virtuel et de l'ouvrir en mode d'édition dans un nouvel onglet d'éditeur.

Chapitre 6: Modification des services virtuels

Le portail DevTest permet d'ouvrir un service virtuel et d'apporter des modifications à divers composants qu'il comprend.

Pour des informations de base sur le portail DevTest, consultez la rubrique *Prise en main*.

Ce chapitre traite des sujets suivants :

[Ouverture d'un service virtuel](#) (page 81)

[Vue de transactions sans état](#) (page 82)

[Vue de conversation](#) (page 87)

[Réponses inconnues](#) (page 88)

[Recherche de texte dans un service virtuel](#) (page 90)

[Ajout d'une note à une signature](#) (page 91)

[Ajout d'une étiquette à une transaction spécifique](#) (page 91)

[Mise à jour manuelle d'un service virtuel](#) (page 92)

[Recherche de transactions correspondant à une demande](#) (page 101)

[URL de service virtuel](#) (page 102)

Ouverture d'un service virtuel

Ouvrez un service virtuel à partir du portail DevTest.

Procédez comme suit:

1. Accédez à la page d'accueil du portail DevTest.
2. Dans le volet gauche, cliquez sur Manage (Gérer).
3. (Facultatif) Sélectionnez un projet différent.
4. Développez la catégorie Virtual Services (Services virtuels).
5. Cliquez sur le nom du service virtuel.

Vue de transactions sans état

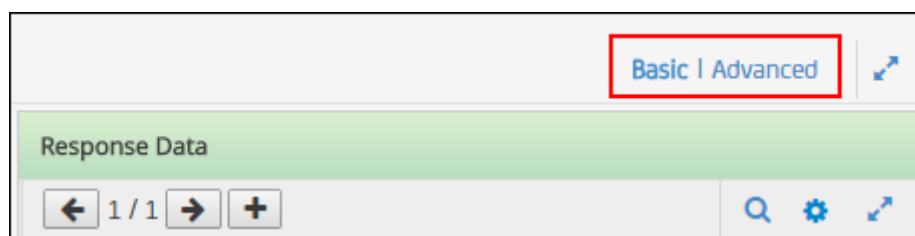
Pour afficher les transactions sans état, ouvrez le service virtuel et sélectionnez Stateless Transactions (Transactions sans état) dans la liste déroulante View (Afficher).

La vue Stateless Transactions (Transactions sans état) est composée des zones suivantes :

- Volet Stateless Signature (Signatures sans état)
- Onglet Specifics (Spécificités)
- Onglet Signature Definition (Définition de signature)

Par défaut, seules les fonctionnalités de base sont affichées. Les utilisateurs expérimentés peuvent accéder aux fonctionnalités avancées en cliquant sur Advanced (Avancé) dans la zone supérieure droite.

Le graphique suivant illustre les liens Basic (De base) et Advanced (Avancé).



Signatures sans état

Lorsque VSE tente de rechercher une correspondance pour une demande entrante dans les transactions sans état, VSE commence par rechercher une *signature* qui correspond à la demande entrante.

Une signature comprend les composants suivants :

- Nom de l'opération
- (Facultatif) Un ensemble de noms d'arguments

Dans un service virtuel, une opération décrit une action à réaliser. Par exemple : opération appelée **depositMoney** (Déposer de l'argent).

Dans un service virtuel, un argument est une paire de nom-valeur que vous pouvez inclure dans une opération. Par exemple, un argument peut être composé du nom **amount** (somme) et de la valeur **100.00**.

Remarque : VSE ne tient pas compte des valeurs d'argument à cette étape du processus de correspondance.

Chaque signature possède un identificateur unique. L'identificateur est unique dans le service virtuel.

Plusieurs signatures peuvent avoir le même nom d'opération. Par exemple :

- La signature A a l'opération **depositMoney** (Déposer de l'argent) et l'argument **amount** (somme).
- La signature B a l'opération de **depositMoney** (Déposer de l'argent) et deux arguments : **amount** (somme) et **date**.

Les identificateurs uniques peuvent vous aider à différencier des signatures dont le nom d'opération est identique.

Vous pouvez également distinguer des signatures dont le nom d'opération est identique en ajoutant des [notes](#) (page 91).

Le graphique suivant illustre le volet Stateless Signatures (Signatures sans état). Pour chaque signature, l'identificateur unique et le nom d'opération s'affichent. La signature sélectionnée a l'identificateur unique **159** et le nom d'opération **addUserObject** (Ajouter un objet d'utilisateur).

Stateless Signatures (7)	
<input type="checkbox"/>	^ v +
153	listUsers
159	addUserObject
165	addUserObject
173	deleteToken
179	GET /itkoExamples/TokenBean
185	GET /itkoExamples/EJB3UserControlBean
191	GET /itkoExamples/EJB3AccountControlBean

Pour afficher les noms d'argument, sélectionnez la signature, puis sélectionnez l'onglet Signature Definition (Définition de signature).

Arguments de données de demande et données de réponse

Si VSE détecte une signature correspondant, il recherchera alors une *transaction spécifique* qui correspond à la demande entrante.

Une transaction spécifique a les composants suivants :

- Nom de l'opération
- (Facultatif) Un ensemble de noms et de valeurs d'argument et d'autres critères de correspondance, tels qu'un opérateur

Chaque transaction spécifique a également un identificateur unique dans le service virtuel.

Le volet Request Data Arguments (Arguments de données de demande) affiche les transactions spécifiques pour la signature sélectionnée.

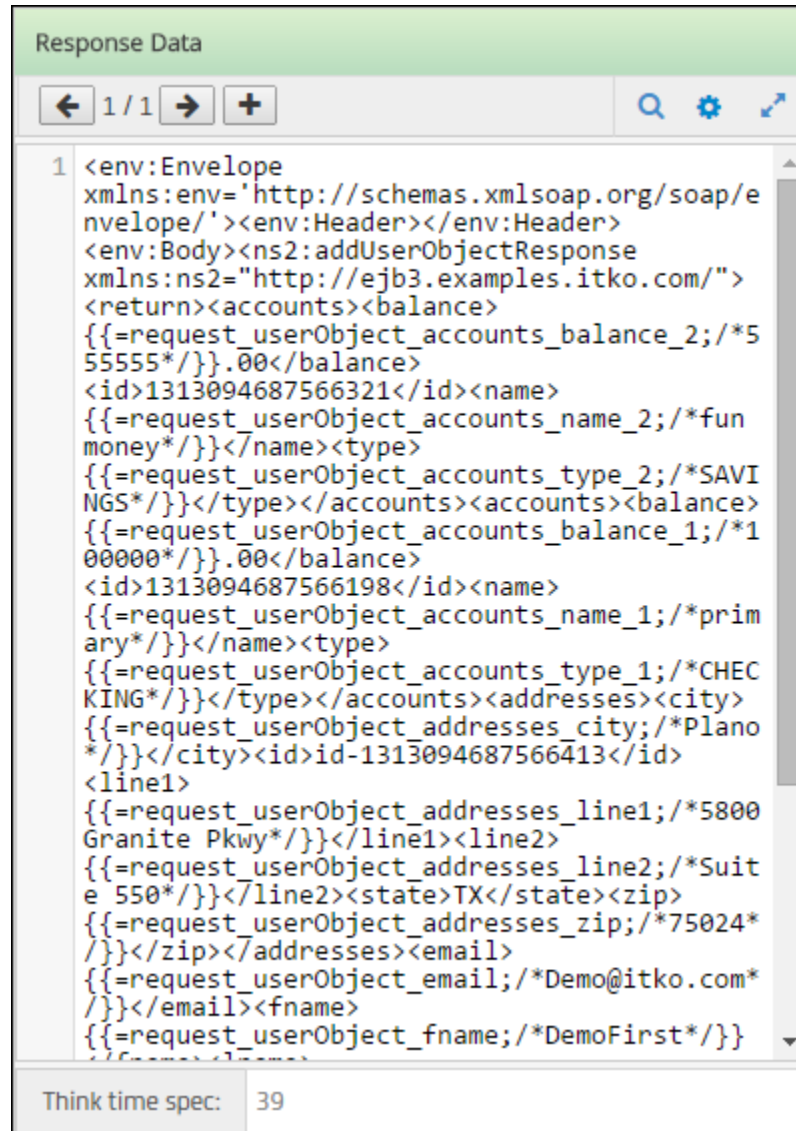
Le graphique suivant représente une transaction spécifique. L'identificateur unique est **162**. Le nom d'opération est **addUserObject** (Ajouter un objet d'utilisateur). Les arguments s'affichent dans une table.

Request Data Arguments			
162 addUserObject			
Name	Operator	Value	Magic String
userObject_accoun...	=	100000	<input type="checkbox"/>
userObject_accoun...	=	primary	<input type="checkbox"/>
userObject_accoun...	=	CHECKING	<input type="checkbox"/>
userObject_accoun...	=	555555	<input type="checkbox"/>
userObject_accoun...	=	fun money	<input type="checkbox"/>
userObject_accoun...	=	SAVINGS	<input type="checkbox"/>
userObject_addres...	=	Plano	<input type="checkbox"/>
userObject_addres...	=	5800 Granite Pkwy	<input type="checkbox"/>
userObject_addres...	=	Suite 550	<input type="checkbox"/>
userObject_addres...	=	TX	<input type="checkbox"/>
		10	20
« 1 2 »			

Si VSE détecte une transaction spécifique correspondante, il enverra la réponse associée à la transaction spécifique.

Le volet Response Data (Données de réponse) contient la réponse pour la transaction spécifique sélectionnée.

Le graphique suivant illustre le volet Response Data (Données de réponse).



Vous pouvez rechercher du texte dans une réponse. Vous pouvez également changer la mise en surbrillance de la syntaxe dans une réponse.

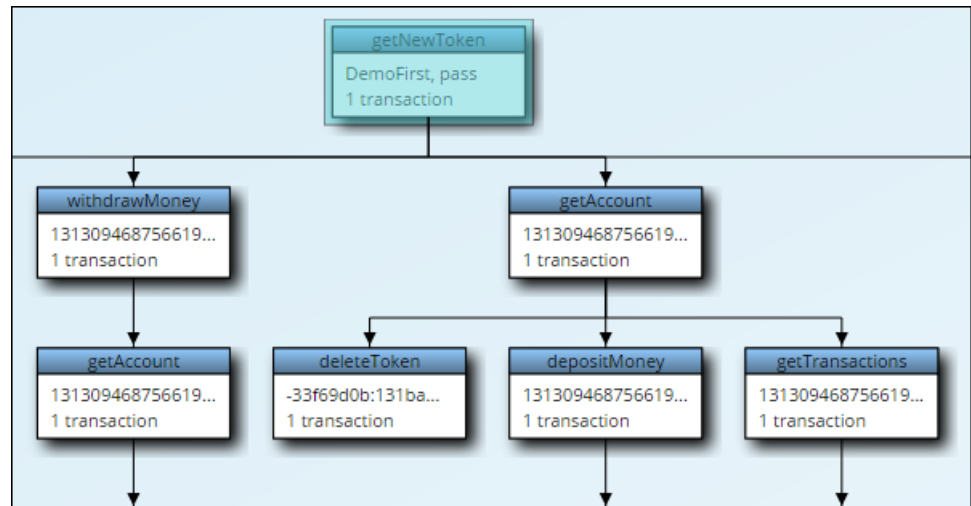
Si VSE ne détecte aucune transaction spécifique correspondante, il enverra la réponse associée à la *transaction par défaut*.

La transaction par défaut s'affiche au bas du volet Request Data Arguments (Arguments de données de demande). Le volet Response Data (Données de réponse) contient la réponse pour la transaction par défaut sélectionnée.

Vue de conversation

Pour afficher une conversation, ouvrez le service virtuel et sélectionnez le nom de la conversation dans la liste déroulante View (Afficher).

Le graphique suivant illustre une partie d'une conversation. La transaction de démarrage est sélectionnée.



Chaque nœud dans une conversation est une [transaction logique](#) (page 62).

La zone supérieure d'un nœud affiche le nom de l'opération.

La zone principale d'un nœud affiche les informations suivantes :

- Les valeurs d'argument de la première transaction spécifique
- Le nombre de transactions spécifiques

Pour afficher une vue miniature de la conversation, cliquez sur Show Outline (Afficher la structure). Cette fonctionnalité est particulièrement utile pour afficher des conversations volumineuses.

Les volets Request Data Arguments (Arguments de données de demande) et Response Data (Données de réponse) contiennent des informations détaillées sur le nœud sélectionné. Le comportement de ces volets est identique à celui pour [des transactions sans état](#) (page 84).

Réponses inconnues

Lorsque le VSE ne parvient pas à trouver de correspondance pour une demande entrante, il envoie l'une des réponses suivantes :

- Response for unknown conversational request (Réponse à la demande conversationnelle inconnue)
- Response for unknown stateless request (Réponse à la demande sans état inconnue)

Pour afficher les réponses inconnues, ouvrez le service virtuel et sélectionnez Service Image Properties (Propriétés de l'image de service) dans la liste déroulante View (Afficher).

Vous pouvez rechercher du texte dans une réponse. Vous pouvez également changer la mise en surbrillance de la syntaxe dans une réponse.

Le graphique suivant illustre un exemple de réponse inconnue.

Response Data

```
1 <html>
2   <head><title>404 Not Found</title></head>
3   <body>
4     <h1>Not Found</h1>
5     <p>The requested URL was not found on this server.
6   </p>
7     <hr/>
8     <p><i>The LISA VSE service could not match your
9 request to a recorded request.&nbsp; Consider expanding your
10 service image.</i></p>
11     <br/><font size="-2">Produced by a LISA virtualized
web server.</font>
    </body>
  </html>
```

Think time spec: 0

Recherche de texte dans un service virtuel

Vous pouvez rechercher du texte dans les composants suivants d'un service virtuel :

- Noms de signature
- Noms de transactions spécifiques
- Données de demande
- Données de réponse
- Identificateurs uniques
- Remarques

Le graphique suivant illustre la table de résultats de recherche.

Search Results for : savings found in Response Body			X
Type	Operation	Snippet	
conversational	getAccount	</balance><id>{{=request_accountId;/*1313094687566321*/}}</id><name>fun money</name><type> SAVINGS </type>	
stateless	listUsers	money</name> <type> SAVINGS </type> </accounts>	

La colonne Type indique si le résultat fait partie d'une transaction sans état ou d'une conversation.

Si le résultat est compris dans le corps d'une réponse, la colonne Snippet mettra en surbrillance l'emplacement du texte.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Dans le champ de recherche, saisissez le texte.
A mesure que vous saisissez, des propositions s'affichent sous le champ de recherche.
3. Cliquez sur une proposition ou appuyez sur Entrée.
La table de résultats de recherche affiche un ensemble de résultats.
4. Cliquez sur une entrée dans la table.
L'éditeur affiche le composant dans lequel le texte est situé.

Ajout d'une note à une signature

Pour annoter une signature, ajoutez une note.

Par exemple, supposons qu'une signature représente un scénario qui se produit rarement. Vous pouvez ajouter une note pour fournir une description du scénario.

Vous pouvez également ajouter une note pour indiquer que vous avez changé les paramètres par défaut d'une signature.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Cliquez avec le bouton droit de la souris sur une signature et cliquez sur Add Note (Ajouter une note).
3. Saisissez du texte. Vous pouvez copier et coller du texte dans la note.
4. Cliquez sur l'icône Save (Enregistrer).

Ajout d'une étiquette à une transaction spécifique

Pour annoter une transaction spécifique, ajoutez une étiquette.

Le graphique suivant illustre une transaction spécifique avec une étiquette. L'étiquette est mise en surbrillance.

176 [deleteToken] UseCase42			
Name	Operator	Value	Magic String
token	=	-33f69d0b:131ba6957...	<input type="checkbox"/>

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Cliquez sur le nom de l'opération de la transaction spécifique.
Un champ de texte s'ouvre.
3. Saisissez l'étiquette et cliquez sur la coche.
L'étiquette s'affiche à droite du nom de l'opération.

Mise à jour manuelle d'un service virtuel

Vous pouvez mettre à jour un service virtuel manuellement à partir du portail DevTest.

Les raisons qui vous amènent à mettre à jour un service virtuel peuvent être les suivantes :

- Pendant l'enregistrement du service virtuel, vous avez oublié d'appeler une opération.
- L'équipe de développement ajoute une nouvelle opération au service en cours de virtualisation.
- Vous voulez tester une proposition de modification pour un service avant l'implémentation de la modification.

Si vous devez effectuer des mises à jour étendues, envisagez une approche alternative :

- [Combiner](#) (page 110) le service virtuel avec un autre service virtuel
- Exécuter le service virtuel avec le mode d'exécution défini sur Image Validation (Validation d'image)
- Réenregistrer le service virtuel complet

Ajout, modification et suppression de signatures

Le portail DevTest vous permet d'effectuer l'une des actions suivantes :

- Ajouter une signature
- Déplacer une signature vers le haut ou le bas
- Modifier la définition d'une signature
- Supprimer une signature

Vous pouvez également ajouter une signature en [important des paires de demande-réponse](#) (page 95).

Lorsque VSE recherche une signature qui correspond à une demande entrante, il commence par le haut de la liste de signatures et continue vers le bas. Par conséquent, le déplacement d'une signature vers le haut ou le bas affecte l'ordre de la mise en correspondance.

Les modifications apportées à une définition de signature sont automatiquement appliquées à toutes les transactions spécifiques pour la signature. Par exemple, si vous ajoutez l'argument **NewArg**, les transactions spécifiques incluront ce nouvel argument.

Par défaut, la mise en correspondance respecte la casse. Pour que les [opérateurs de comparaison](#) (page 97) ignorent la casse d'un argument, accédez à l'onglet Signature Definition (Définition de signature) et désélectionnez la case à cocher dans la colonne Case Sensitive (Respecter la casse).

Le graphique suivant illustre l'onglet Signature Definition (Définition de signature).

Request Data Arguments			
<div> ^ v + 🗑️ ↕️ </div>			
Name	Date Pattern	Case Sensitive	Is Numeric
token	Not set	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Par défaut, les valeurs d'argument sont traitées comme des chaînes. Par exemple, 10000 est considéré comme une valeur inférieure à 9, car 1 est alphabétiquement antérieur à 9. Pour forcer le traitement d'une valeur d'argument comme nombre, accédez à l'onglet Signature Definition (Définition de signature) et sélectionnez la case à cocher dans la colonne Is Numeric (Nombre).

Les modèles de date sont utilisés pour analyser des informations de date. La colonne Date Pattern (Modèle de date) est en lecture seule.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Pour ajouter une signature :
 - a. Vérifiez que la vue avancée est affichée.
 - b. Dans le volet Stateless Signatures (Signatures sans état), cliquez sur le bouton Add (Ajouter) et sélectionnez Add New Signature (Ajouter une nouvelle signature).
 - c. Entrez le nom de l'opération.
 - d. Cliquez sur Continue (Continuer).
 - e. Sélectionnez la nouvelle signature.
 - f. (Facultatif) Accédez à l'onglet Signature Definition (Définition de signature) et ajouter un ou plusieurs arguments.
 - g. Accédez à l'onglet Specifics (Spécificités) et ajoutez une ou plusieurs transactions spécifiques.
3. Pour déplacer une signature vers le haut ou le bas, sélectionnez la signature et cliquer sur l'icône Up (Haut) ou Down (Bas). Vous pouvez également faire glisser une signature vers le nouvel emplacement.
4. Pour modifier la définition d'une signature :
 - a. Sélectionnez la signature.
 - b. Accédez à l'onglet Signature Definition (Définition de signature).
 - c. Apportez les modifications appropriées. Vous pouvez déplacer un argument vers le haut ou le bas. Vous pouvez ajouter ou supprimer un argument. Vous pouvez modifier les paramètres des colonnes Case Sensitive (Respecter la casse) et Is Numeric (Nombre). Vous pouvez changer le nom de l'opération.
5. Pour supprimer une signature, sélectionnez-la signature et cliquez sur l'icône Delete (Supprimer). Vous pouvez utiliser la touche Ctrl pour sélectionner plusieurs signatures.

Importation de paires demande-réponse

Une paire demande-réponse consiste en un fichier de demande et un ou plusieurs fichiers de réponse.

Lorsque vous importez une paire demande-réponse, la demande est traitée au niveau du service virtuel.

Si le service virtuel contient une signature avec les mêmes arguments, une transaction spécifique sera ajoutée à la signature.

Si le service virtuel ne contient pas de signature avec les mêmes arguments, une signature sera ajoutée. Les détails de la demande et de la réponse sont ajoutés comme transaction spécifique à la signature. Une transaction par défaut est également ajoutée.

La paire demande-réponse doit être au format texte ou XML.

La liste suivante contient un exemple des noms de fichier d'une paire demande-réponse :

- **depositMoney-req.xml**
- **depositMoney-rsp1.xml**
- **depositMoney-rsp2.xml**
- **depositMoney-rsp3.xml**

Le nom du fichier de demande doit inclure un préfixe suivi de la chaîne **-req**.

Le nom des fichiers de réponse doit inclure le même préfixe suivi de la chaîne **-rsp**. Comme illustré dans l'exemple précédent, vous pouvez fournir plusieurs fichiers de réponse en ajoutant un nombre après la chaîne **-rsp**. Plusieurs fichiers de réponse sont applicables aux scénarios de messagerie.

Vous pouvez spécifier des métadonnées de demande et de réponse en incluant un ou plusieurs fichiers sidecar. Pour plus d'informations, consultez la section [Fichiers sidecar avec des paires demande-réponse](#) (page 126).

Remarque : Si le service virtuel est basé sur un protocole de transport autre que le protocole HTTP, vous devez remplir la condition suivante pour cette fonctionnalité : ouvrez le modèle de service virtuel correspondant dans la DevTest Workstation (Station de travail) et saisissez le chemin d'accès au fichier de session d'enregistrement dans le champ Documentation. Un fichier de session d'enregistrement est l'un des fichiers suivants :

VRS, fichier

Ce fichier est généré par l'enregistreur d'image de service virtuel dans la DevTest Workstation (Station de travail).

VR2, fichier

Ce fichier est généré par l'enregistreur CA Service Virtualization dans le portail de DevTest.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Dans le volet Stateless Signatures (Signatures sans état), cliquez sur le bouton Add (Ajouter) et sélectionnez Import Request/Response Pairs (Importer des paires demande-réponse).

La boîte de dialogue Import Request/Response Pairs (Importer des paires demande-réponse) s'ouvre.

3. Spécifiez les fichiers qui contiennent les paires demande-réponse. Vous pouvez déplacer les fichiers dans la boîte de dialogue, ou cliquer sur un bouton pour sélectionner les fichiers à partir d'un système de fichiers.
4. Cliquez sur Done (Terminé).

Opérateurs de comparaison pour des arguments

Chaque argument d'une transaction spécifique contient un opérateur de comparaison. L'opérateur indique la méthode de correspondance de la valeur d'argument à une demande entrante.

Par défaut, l'opérateur est défini sur le signe égal (=). Par conséquent, une correspondance se produit si la valeur d'argument dans la demande entrante est identique à la valeur d'argument dans la transaction spécifique.

Les symboles mathématiques suivants sont également pris en charge :

- Différent de (!=)
- Inférieur à (<)
- Inférieure ou égale à (<=)
- Supérieur à (>)
- Supérieure ou égale à (>=)

Vous pouvez également spécifier une expression régulière ou une expression de propriété. Utilisez ces opérateurs avec la colonne Value (Valeur).

Par exemple, supposons qu'un code postal à cinq chiffres commençant par **750** est valide. Vous pourriez définir l'opérateur sur RegEx (Expression régulière) et définissez la valeur sur le texte suivant :

750\d\d

Si vous voulez autoriser tout type de valeur pour un argument , définissez l'opérateur sur Anything (Tout type d'argument).

Le graphique suivant illustre l'utilisation de divers opérateurs. Dans ce scénario, toutes les valeurs d'argument de la demande entrante correspondent aux critères dans la transaction spécifique. Par exemple, le solde de compte dans la demande entrante est de 5000. La transaction spécifique indique que le solde de compte doit être supérieur à 100.

Demande entrante		Transaction spécifique		
nom	Value (Valeur)	nom	Operator (Opérateur)	Value (Valeur)
Type de compte		Type de compte		
Soldede compte	Vérification		=	Vérification
Nom de compte	5000	Soldede compte	>	100
Nom 2		Nom de compte	Anything (Tout type d'argument)	Nom 1

Désactivation et activation de chaînes magiques

Vous pouvez classer chaque argument dans une transaction spécifique comme [chaîne magique](#) (page 52).

Lorsque vous affichez une transaction spécifique dans le portail DevTest, la colonne Magic String (Chaîne magique) indique si chaque argument a été classé comme chaîne magique.

La colonne Magic String (Chaîne magique) est en lecture seule. DevTest Workstation fournit une méthode pour changer le paramètre. Pour plus d'informations, reportez-vous à la section [Editeur de données de demandes](#) (page 304).

Modification des attributs de demande

Dans un service virtuel, un *attribut* est une paire nom-valeur qui contient des informations sur le corps de la demande.

Supposons que vous enregistrez un service virtuel avec le protocole de transport HTTP. Les attributs dans le service virtuel incluent l'espace de noms XML et la demande brute enregistrée.

Les attributs de demande ne sont pas utilisés pour la mise correspondance.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Vérifiez que la vue avancée est affichée.
3. Cliquez sur le lien Attributs à droite de l'étiquette Request Data (Données de demande).
4. Ajoutez, changez et supprimez des attributs en fonction de vos besoins.

Modification des métadonnées de demande

Dans un service virtuel, le terme *metadata* (métadonnées) fait référence aux paires nom-valeur qui contiennent des informations qui ne font pas partie du corps de la demande ou de la réponse.

Supposons que vous enregistrez un service virtuel avec le protocole de transport HTTP. Les métadonnées de demande dans le service virtuel incluent la méthode HTTP et le type de contenu.

Les métadonnées de demande ne sont pas utilisées pour la mise correspondance.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Vérifiez que la vue avancée est affichée.
3. Cliquez sur le lien Metadata (Métadonnées) à droite de l'étiquette Request Data (Données de demande).
4. Ajoutez, changez et supprimez des métadonnées en fonction de vos besoins.

Modification des métadonnées de réponse

Dans un service virtuel, le terme *metadata* (métadonnées) fait référence aux paires nom-valeur qui contiennent des informations qui ne font pas partie du corps de la demande ou de la réponse.

Supposons que vous enregistrez un service virtuel avec le protocole de transport HTTP. Les métadonnées de réponse dans le service virtuel incluent le code de réponse HTTP et le type de contenu.

Les métadonnées de réponse ne sont pas utilisées pour la mise correspondance.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Vérifiez que la vue avancée est affichée.
3. Cliquez sur le lien Metadata (Métadonnées) à droite de l'étiquette Response (Réponse).
4. Ajoutez, changez et supprimez des métadonnées en fonction de vos besoins.

Modification du délai de réflexion d'une réponse

Chaque réponse pour une transaction spécifique inclut un champ appelé Think time spec (Spécification du délai de réflexion). Ce champ spécifie la durée de l'attente avant l'envoi de la réponse. Si l'image de service a été créée à partir d'un enregistrement, la valeur initiale est basée sur le comportement de la réponse observé pendant l'enregistrement.

Pour simuler des performances plus rapides, diminuez la valeur.

Pour simuler des performances plus lentes, augmentez la valeur.

Par défaut, la valeur est définie en millisecondes. Pour spécifier l'unité de temps, ajoutez un suffixe au nombre. La casse est sans importance. Les suffixes suivants sont autorisés :

- **t** : millisecondes
- **s** : secondes
- **m** : minutes
- **h** : heures

Vous pouvez spécifier une plage de nombres. Le délai de réflexion est sélectionné de manière aléatoire dans la plage. Par exemple, la valeur **100-1000** spécifie un délai de réflexion aléatoire entre 100 et 1 000 millisecondes.

Si le délai de réflexion est de 10 millisecondes et que le traitement de la demande et la recherche d'une réponse prend 5 millisecondes, cela représente seulement 5 millisecondes supplémentaires de retard.

Si le délai de réflexion est de 0, ou inférieur à la durée de traitement de la demande, VSE envoie la réponse aussi rapidement que possible.

Les réponses inconnues dans un service virtuel contiennent le même champ.

Lorsque vous [déployez un service virtuel](#) (page 103), vous pouvez configurer un champ relatif appelé Think time scale (Pourcentage de délai de réflexion). Ce champ spécifie le pourcentage qui servira à multiplier tous les champs Think time spec (Spécification du délai de réflexion) dans une image de service.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Recherchez la réponse pour la transaction spécifique ou la réponse inconnue.
3. Changez la valeur du champ Think time spec (Spécification du délai de réflexion).

Recherche de transactions correspondant à une demande

Supposons que lors de l'envoi d'une demande à un service virtuel, la réponse n'est pas celle que vous attendiez.

Vous pouvez ouvrir le service virtuel et saisir la demande. Le portail DevTest indique les composants dans le service virtuel qui correspondent à la demande. Les résultats peuvent vous aider à déterminer comment corriger le problème.

La demande doit être au format texte ou XML.

Le portail DevTest vous permet de saisir le contenu réel de la demande, ou de spécifier un fichier de demande. Le nom du fichier de demande doit inclure un préfixe suivi de la chaîne **-req** (par exemple : **depositMoney-req.txt**).

Si les résultats incluent des signatures sans état ou des transactions spécifiques sans état, l'ordre d'affichage est identique à celui de leur correspondance lors de la lecture.

L'ordre des résultats conversationnels n'est pas significatif.

Remarque : Si le service virtuel est basé sur un protocole de transport autre que le protocole HTTP, vous devez remplir la condition suivante pour cette fonctionnalité : ouvrez le modèle de service virtuel correspondant dans la DevTest Workstation (Station de travail) et saisissez le chemin d'accès au fichier de session d'enregistrement dans le champ Documentation. Un fichier de session d'enregistrement est l'un des fichiers suivants :

VRS, fichier

Ce fichier est généré par l'enregistreur d'image de service virtuel dans la DevTest Workstation (Station de travail).

VR2, fichier

Ce fichier est généré par l'enregistreur CA Service Virtualization dans le portail de DevTest.

Procédez comme suit:

1. Ouvrez un service virtuel.
2. Cliquez sur Find a Match by Request (Rechercher une correspondance par demande).
3. Effectuez l'une des actions suivantes :
 - Collez la demande dans la zone de texte.
 - Faites glisser le fichier de demande dans la boîte de dialogue.
 - Cliquez sur Select Request File (Sélectionner un fichier de demande) et sélectionnez le fichier de demande à partir d'un système de fichiers.
4. Cliquez sur Find a Match (Rechercher une correspondance).

La table de résultats de la recherche affiche toutes les correspondances détectées.

5. Pour afficher la demande d'origine, cliquez sur View Request (Afficher la demande).
6. Cliquez sur une entrée dans la table Search Results (Résultats de la recherche).

Le composant correspondant est affiché.

URL de service virtuel

Lorsque vous ouvrez un service virtuel dans le portail DevTest, l'URL spécifie le chemin vers le composant de service virtuel affiché.

Le graphique suivant illustre le format de l'URL d'une transaction spécifique dans la vue Stateless Transactions (Transactions sans état).

```
http://localhost:1507/devtest/#/main/serviceimageeditor/0/bmS18CcJu-b74PB4tcz/bmS18CcJu/b74PB4tcz/kioskV6/0?signatureId=153&specificId=156&view=stateless
```

Vous pouvez mettre une URL donnée en signet. Vous pouvez ouvrir le service virtuel ultérieurement pour le même composant en entrant l'URL dans un navigateur.

Déploiement d'un service virtuel

Vous pouvez déployer un service virtuel à partir du portail DevTest.

Procédez comme suit:

1. Accédez à la page d'accueil du portail DevTest.
2. Dans le volet de navigation gauche, cliquez sur Manage (Gérer).
3. (Facultatif) Sélectionnez un projet différent.
4. Cliquez avec le bouton droit de la souris sur le service virtuel que vous voulez déployer et sélectionnez Deploy (Déployer).

La boîte de dialogue Deploy Virtual Service (Déployer un service virtuel) s'ouvre.

5. Sélectionnez le serveur de VSE sur lequel vous voulez effectuer le déploiement.
6. Modifiez les champs à votre convenance :

Balise group

Nom du [groupe de services virtuels](#) (page 433) pour le service virtuel. Si les services virtuels déployés ont des balises group, ces balises sont disponibles dans la liste déroulante. Une balise group doit commencer par un caractère alphanumérique et peut contenir des caractères alphanumériques et les caractères spéciaux suivants :

- Point (.)
- Tiret (-)
- Trait de soulignement (_)
- Symbole du dollar (\$)

Concurrent capacity (Capacité simultanée)

Nombre qui indique la capacité de charge. La *capacité* est le nombre d'utilisateurs virtuels (instances) qui peuvent être exécutés simultanément avec le modèle de service virtuel. Dans cet exemple, la capacité indique le nombre de threads existants pour les demandes de service de ce modèle de service.

VSE alloue un nombre de threads équivalent à la capacité simultanée totale. Chaque thread consomme quelques ressources système, même lorsqu'ils sont inactifs. Par conséquent, pour optimiser les performances système globales, définissez ce paramètre aussi sur une valeur aussi faible que possible. Déterminez les paramètres corrects de manière empirique en les ajustant jusqu'à atteindre les performances souhaitées, ou jusqu'à leur augmentation n'entraîne plus d'amélioration des performances.

Les protocoles prédéfinis utilisent un service d'exécution de tâches de niveau structure pour réduire l'utilisation des threads. Dans le cas de ces protocoles, une capacité simultanée supérieure à 2-3 par noyau est rarement utile, sauf si le modèle de service virtuel a été considérablement personnalisé.

Pour des extensions et tout modèle de service virtuel qui n'utilise pas de protocole prédéfini, la définition d'un délai de réflexion long peut consommer un thread pour la durée de ce délai. Dans ces cas, vous devrez peut-être augmenter la capacité simultanée.

La formule suivante donne un paramètre initial approximatif dans ces cas :

Capacité simultanée = (Transactions souhaitées par seconde / 1000) * délai de réflexion moyen en ms * (pourcentage de délai de réflexion / 100)

Exemple :

Supposons que vous utilisez un protocole personnalisé qui n'utilise pas le service d'exécution de tâches de structure pour gérer des délais de réflexion. Vous voulez un débit global de 100 transactions par seconde. Le délai de réflexion moyen dans l'image de service est de 200 ms et le service virtuel est déployé avec un délai de réflexion de 100 %.

$(100 \text{ transactions par seconde} / 1000) * 200 \text{ ms} * (100 / 100)$
= 20

Dans ce cas, les threads se bloquent pendant environ 200 ms en moyenne avant de répondre et durant ce temps, ils ne parviennent pas à gérer de nouvelles demandes. Nous avons par conséquent besoin d'une capacité de 20 pour contenir 100 transactions par seconde. Un thread deviendra disponible toutes les 10 ms en moyenne, ce qui devrait suffire à obtenir 100 transactions par seconde.

Valeur par défaut : 1

Think time Scale (Pourcentage de délai de réflexion)

Indique le pourcentage de délai de réflexion enregistré.

Remarque : Une étape déduit sa propre durée de traitement du délai de réflexion pour obtenir une fréquence d'exécutions des tests cohérente.

Par défaut : 100

Exemples :

- Pour doubler le délai de réflexion, utilisez 200.
- Pour réduire de moitié le délai de réflexion, utilisez 50.

7. Cliquez sur Deploy (Déployer).

Remarque : un service virtuel peut avoir les états suivants :

Deployed (Déployé)

Aucun service portant le nom saisi n'est déjà déployé. Le service est déployé.

Redeployed (Redéployé)

Un service portant le nom saisi est déployé avec le même fichier .vsm. Le service est redéployé.

Overridden (Remplacé)

Un service portant le nom saisi est déployé avec un fichier .vsm différent de celui associé au service indiqué. L'application vous invite à remplacer le service déployé.

Chapitre 7: Utilisation de la Workstation (Station de travail) et de la console avec CA Service Virtualization

Ce chapitre traite des sujets suivants :

[Création d'images de service](#) (page 109)

[Modification d'images de service](#) (page 295)

[Modification d'un modèle de service virtuel](#) (page 329)

[Désensibilisation des données](#) (page 375)

[Exécution de la virtualisation](#) (page 379)

Chapitre 8: Création d'images de service

Ce chapitre traite des sujets suivants :

[Ouverture d'une image de service](#) (page 109)
[Combine Service Images \(Combiner les images de service\)](#) (page 110)
[Suppression d'une image de service](#) (page 111)
[Création d'une image de service](#) (page 111)
[Création d'une image de service à partir de zéro](#) (page 113)
[Création d'une image de service à partir d'un fichier WSDL](#) (page 113)
[Création d'une image de service à partir d'un objet WADL](#) (page 117)
[Création d'une image de service à partir d'un objet RAML](#) (page 119)
[Création d'une image de service à partir de Layer 7](#) (page 121)
[Création d'une image de service à partir d'une paire réponse-demande](#) (page 122)
[Création d'une image de service à partir d'un objet PCAP](#) (page 127)
[Création d'une image de service à partir d'un enregistrement](#) (page 130)
[Création et déploiement d'un service virtuel avec VSEasy](#) (page 226)
[Utilisation des modèles de service virtuel](#) (page 227)
[Utilisation de protocoles de données](#) (page 229)

Ouverture d'une image de service

L'enregistreur d'image de service virtuel génère des images de service. Les images de service ont pour but de représenter vos enregistrements (une version manipulée ou modifiée du trafic brut enregistré).

Remarque : Si vous disposez d'images de service de versions de VSE antérieures à LISA 6.0, exportez-les. L'exportation permet de les déplacer de la base de données de versions antérieures vers le système de fichiers de la version actuelle dans lequel elles sont stockées. Pour plus d'informations, consultez la rubrique [Images de service héritées](#) (page 295).

L'ouverture d'une image de service permet de modifier l'image dans l'éditeur d'images de service.

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur l'image de service et sélectionnez Open (Ouvrir).
L'éditeur d'images de service s'ouvre.
2. Vérifiez l'image de service sélectionnée et apportez les modifications nécessaires.

Combine Service Images (Combiner les images de service)

La combinaison d'images de service permet d'ajouter d'autres fonctionnalités à une image de service existante.

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur l'image de service et sélectionnez Combine (Combiner).

La boîte de dialogue Combine Service Images (Combiner les images de service) s'ouvre.
2. Sélectionnez une ou plusieurs images de service à combiner dans l'image d'origine (la cible).
3. Pour remplacer les données cibles correspondantes, sélectionnez la case à cocher Favor source image(s) (Utiliser les images sources par défaut).

Lorsque les images de service sont combinées, une correspondance est établie entre chaque transaction sans état (au niveau méta) issue de chaque image de service source et l'image de service cible. Dans chaque correspondance, les transactions spécifiques issues de la source correspondent à celles de la cible. En cas de non-correspondance, les transactions sources seront ajoutées à l'image cible. En cas de correspondance, les transactions doivent être fusionnées.

Vous pouvez remplacer les données dans la transaction cible correspondante par les données sources (corps de réponse, par exemple). Vous pouvez également laisser les données cibles telles quelles.

Remarque : Le processus de combinaison des conversations est similaire. Pour chaque conversation dans chaque image de service source, une correspondance est établie entre leur transaction de démarrage et leur transaction de démarrage dans les conversations de la cible. Si aucun démarreur ne correspond, de nouvelles conversations seront créées dans l'image cible. En cas de correspondance, le processus de la liste de transactions sans état sera également appliqué à chaque noeud méta dans l'arborescence des conversations sources. Le processus ajoute de nouvelles transactions et fusionne les transactions correspondantes en conséquence.

Suppression d'une image de service

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur l'image de service et sélectionnez Delete (Supprimer).

Une boîte de dialogue de confirmation vous invite à confirmer la suppression de l'image de service sélectionnée.

2. Cliquez sur OK (Redéployer/déployer).

L'image de service sélectionnée est supprimée de façon permanente de votre projet.

Remarque : Il est recommandé de considérer les images de service comme transitoires et de supprimer les images de service inutilisées ou obsolètes.

Création d'une image de service

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur VirtualServices, Images.
2. Sélectionnez Create New VS Image (Créer une image de service virtuel).

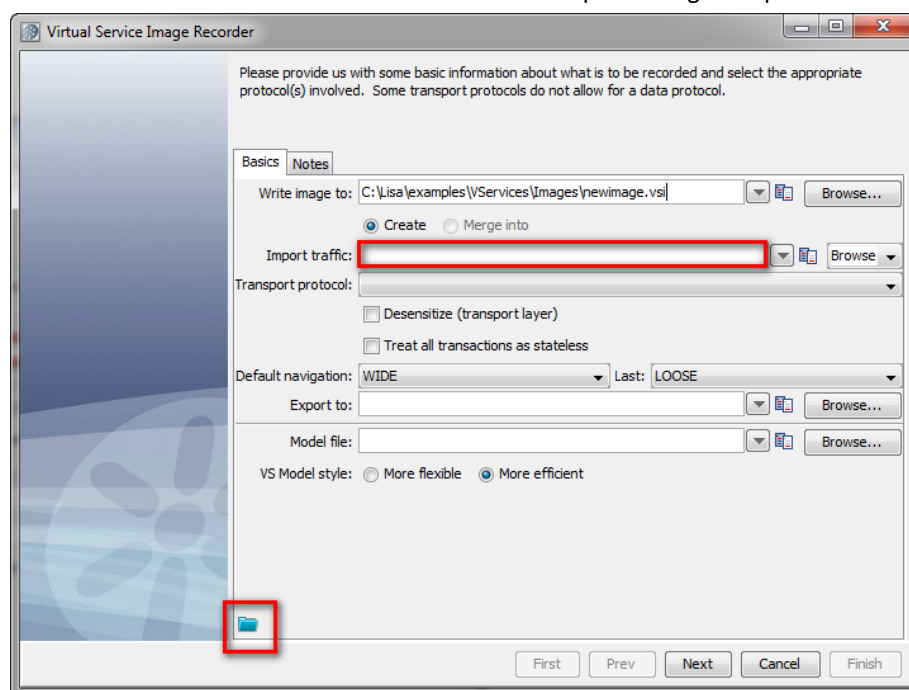
Importation du trafic brut

Vous pouvez uniquement importer le trafic brut via l'enregistreur d'image de service du VSE.

Procédez comme suit:

1. Créez une image de service et sélectionnez Virtual Service Image Recorder (Enregistreur d'image de service virtuel).
2. Entrez le nom du fichier de trafic brut ou sélectionnez-le à l'aide du navigateur de fichiers.
3. Pour charger les paramètres de l'image de service à partir d'une session d'enregistrement précédente, cliquez sur le dossier bleu situé au coin inférieur du panneau.

Vous êtes invité à rechercher un fichier .vrs à utiliser pour charger les paramètres.



4. Pour continuer le processus d'enregistrement normalement, sélectionnez un protocole de transport.

Lorsque vous lancez le processus d'enregistrement, le VSE importe le trafic brut vers une nouvelle image de service.

Création d'une image de service à partir de zéro

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur VirtualServices et sélectionnez Create New VS Image (Créer une image de service virtuel), From scratch (A partir de zéro).
2. Entrez les informations d'identification et du protocole et cliquez sur OK.
La fenêtre Service Image Editor (Editeur d'images de service) s'ouvre.
3. Entrez les paramètres et les informations spécifiques pour la nouvelle image de service.

Création d'une image de service à partir d'un fichier WSDL

Vous pouvez générer une image de service Web virtuel à partir d'un fichier WSDL de différentes manières :


- [A partir du menu Quick Start \(Démarrage rapide\)](#) (page 114)
- [A l'aide de l'option Create New VS Image \(Créer une image de service virtuel\)](#) (page 115)

A partir du menu Quick Start (Démarrage rapide)

Cette section décrit la procédure de création d'une image de service virtuel à partir d'un fichier WSDL à l'aide de l'interface utilisateur.


Procédez comme suit:

1. Sélectionnez **Create an SI from a WSDL** (Créer une image de service à partir d'un fichier WSDL) dans le menu Quick Start (Démarrage rapide).
2. Dans l'onglet **Connection** (Connexion), entrez le nom d'un fichier WSDL.

Pour rechercher des fichiers WSDL sur votre système, cliquez sur  **Utilities** (Utilitaires).

Une fois le nom du fichier WSDL saisi, les champs **Service** et **Port** sont remplis. Les opérations associées sont répertoriées dans l'onglet **Operations**. Vous pouvez sélectionner **All** (Tout), **None** (Aucun(e)), ou utiliser les cases à cocher pour sélectionner des opérations spécifiques à tester.

3. Saisissez le nom de l'image de service à créer dans le champ **Save to** (Enregistrer dans) au bas de la fenêtre.

Remarque : Si le nom de l'image de service par défaut est déjà utilisé, une icône d'avertissement s'affichera. 

4. Cliquez sur la flèche verte  au bas de la fenêtre.

L'éditeur d'images de service s'ouvre et affiche l'image de service.

Création d'une image de service virtuel


Cette section décrit la procédure de création d'une image de service virtuel à partir d'un fichier WSDL à l'aide de l'option Create New VS Image (Créer une image de service virtuel).

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur VirtualServices (Services virtuels), Images et sélectionnez Create New VS Image (Créer une image de service virtuel), From WSDL (A partir d'un fichier WSDL).

La fenêtre Virtual Service From WSDL (Service virtuel à partir d'un fichier WSDL) s'ouvre.

2. Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.
3. Acceptez les valeurs par défaut pour les autres champs de la fenêtre.

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

4. Cliquez sur Next (Suivant).

L'onglet Connection (Connexion) s'affiche.

5. Saisissez le numéro de port que le service virtuel écoute dans le champ Listen on port (Port d'écoute) au bas de la fenêtre.
6. Ajoutez le fichier WSDL à virtualiser dans le champ WSDL URL (URL du document WSDL).

Il peut s'agir d'un fichier local ou d'une URL.

7. Dans le champ Service, sélectionnez le service dans ce fichier WSDL qui doit être virtualisé.

En général, une seule sélection est disponible.

8. Dans ce service, sélectionnez les opérations à virtualiser.

Par défaut, toutes les opérations sont sélectionnées.

9. Cliquez sur Next (Suivant).

Les options de protocoles de données côté demande/réponse s'ouvrent.

10. Sélectionnez **Web Services (SOAP) (Services Web (SOAP))**.


La liste Request Side Data Protocols (Protocoles de données côté demande) est préremplie avec les services Web (SOAP) et les gestionnaires de protocoles de données XML. La liste Response Side Data Protocols (Protocoles de données côté réponse) est automatiquement remplie avec le protocole de données de texte délimité.

11. Cliquez sur Next (Suivant).

12. Pour des informations sur la configuration du protocole de données de texte délimité, consultez la section [Protocole de données de texte délimité](#) (page 254) dans la rubrique *Utilisation de CA Service Virtualization*. Une fois le protocole de données de texte délimité configuré, cliquez sur Next (Suivant).

Dans la fenêtre suivante, l'image de service est générée et l'assistant a terminé.

13. Cliquez sur Finish (Terminer).

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

Le modèle de service virtuel vide est rempli par des étapes.

14. Enregistrez le modèle de service virtuel.

L'image de service générée est un service stub. L'image de service renvoie des réponses correctement formatées, mais il s'agit de valeurs par défaut.

Le modèle de service virtuel enregistré est le modèle déployé dans l'environnement de services virtuels.

Création d'une image de service à partir d'un objet WADL


Cette section décrit la procédure de création d'une image de service virtuel à partir d'un fichier WADL.

Procédez comme suit:

1. Effectuez l'une des opérations suivantes :
 - Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur VirtualServices (Services virtuels) et sélectionnez Create New VS Image (Créer une image de service virtuel), From WADL (A partir d'un fichier WSDL).
 - A partir du menu File (Fichier), sélectionnez New (Créer), VS Image (Image de service), From WADL (A partir d'un fichier WADL).

La fenêtre Virtual Service From WADL (Service virtuel à partir d'un fichier WSDL) s'ouvre.


2. Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

Pour plus d'informations sur les descriptions de champ, consultez la section [Onglet Basics \(Paramètres de base\)](#) (page 131).

3. Cliquez sur Next (Suivant).
4. Dans le champ Listen on port (Port d'écoute) au bas de la fenêtre, entrer le numéro de port sur lequel le service virtuel écoute.
5. Dans le champ WADL URL (URL du fichier WADL), ajoutez le fichier WADL à virtualiser.

Il peut s'agir d'un fichier WADL sur le système de fichiers ou d'une URL.

6. Cliquez sur Refresh WADL Cache (Actualiser la mémoire cache du WADL) .
7. Dans le champ Endpoint (Terminal), sélectionnez le terminal dans le WADL qui doit être virtualisé.

Remarque : En général, une seule sélection est disponible.


8. Dans le volet Methods, sélectionnez les méthodes dans le terminal à virtualiser.

Par défaut, toutes les méthodes sont sélectionnées. Vous pouvez cliquer sur Select All (Tout sélectionner) ou Select None (Ne rien sélectionner), comme approprié.
9. Cliquez sur Next (Suivant).
10. Ajoutez ou placez sous forme de chaîne d'autres gestionnaires de protocole de données comme il convient. Par défaut, le gestionnaire de protocole de données Rest est sélectionné.

11. Cliquez sur Next (Suivant).

Dans la fenêtre suivante, l'image de service est générée et l'assistant a terminé.

12. Cliquez sur Finish (Terminer).

Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

Remarque : Le paramètre `lisa.vse.rest.max.optionalqueryparams` spécifie le nombre maximum de paramètres de requête facultatifs à traiter par méthode dans un fichier WADL. La valeur par défaut est cinq ; tout paramètre facultatif après le cinquième est ignoré. Il est recommandé de ne pas définir ce paramètre sur une valeur supérieure à cinq. Cela peut entraîner une augmentation exponentielle du nombre de réponses générées après le cinquième paramètre.

Création d'une image de service à partir d'un objet RAML

Cette section décrit la procédure de création d'une image de service virtuel à partir d'un objet RAML.

Un objet RAML peut définir des corps de message à l'aide d'une combinaison de propriétés Schema et Example. La propriété Example est utilisée pour le corps de la transaction dans DevTest. Veillez à spécifier la propriété Example dans un corps de message ; sinon, la propriété Schema sera utilisée à la place.


Important : La propriété Schema n'est interprétée d'aucune façon et s'affiche exactement telle que spécifiée dans le corps de la transaction dans DevTest.

Procédez comme suit:

1. Effectuez l'une des opérations suivantes :
 - Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur VirtualServices (Services virtuels) et sélectionnez Create New VS Image (Créer une image de service virtuel), From RAML (A partir d'un objet RAML).
 - A partir du menu File (Fichier), sélectionnez New (Créer), VS Image (Image de service), From RAML (A partir d'un objet RAML).

La fenêtre Virtual Service From RAML (Service virtuel à partir d'un objet RAML) s'ouvre.

2. Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

Pour plus d'informations sur les descriptions de champ, consultez la section [Onglet Basics \(Paramètres de base\)](#) (page 131).

3. Cliquez sur Next (Suivant).
4. Dans le champ Listen on port (Port d'écoute) au bas de la fenêtre, entrez le numéro de port sur lequel le service virtuel écoute.
5. Dans le champ RAML URL (URL de l'objet RAML), ajoutez le RAML du service Web à virtualiser.

Vous pouvez également sélectionner l'objet RAML dans la liste déroulante, ou cliquer sur le bouton Browse (Parcourir) pour rechercher l'objet RAML à partir du système de fichiers.

Il peut s'agir d'un fichier RAML sur le système de fichiers ou d'une URL.

6. Cliquez sur Refresh RAML Cache (Actualiser la mémoire cache du RAML) .

DevTest analyse le RAML, puis remplit le champ Endpoint (Terminal) et le volet Methods.

Remarque : Si DevTest ne parvient pas à analyser l'objet RAML, une icône d'avertissement s'affichera après le champ Endpoint (Terminal). Pour afficher le message d'erreur, cliquez sur l'icône d'avertissement.


7. Dans le volet Methods, sélectionnez les méthodes à virtualiser.

Par défaut, toutes les méthodes sont sélectionnées. Vous pouvez cliquer sur Select All (Tout sélectionner) ou Select None (Ne rien sélectionner), comme approprié. Au moins une méthode est requise.

8. Cliquez sur Next (Suivant).
9. Ajoutez ou placez sous forme de chaîne d'autres gestionnaires de protocole de données comme il convient. Par défaut, le gestionnaire de protocole de données Rest est sélectionné.
10. Cliquez sur Next (Suivant).

Dans la fenêtre suivante, l'image de service est générée et l'assistant a terminé.

11. Cliquez sur Finish (Terminer).

Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

Remarque : Le paramètre `lisa.vse.rest.max.optionalqueryparams` spécifie le nombre maximum de paramètres de requête facultatifs à traiter par méthode dans un fichier RAML. La valeur par défaut est cinq ; tout paramètre facultatif après le cinquième est ignoré. Il est recommandé de ne pas définir ce paramètre sur une valeur supérieure à cinq. Cela peut entraîner une augmentation exponentielle du nombre de réponses générées après le cinquième paramètre.

Création d'une image de service à partir de Layer 7

Le service virtuel issu de Layer7 génère une image de service virtuel à partir de Layer 7.

Conditions préalables

1. Téléchargez l'outil Layer 7 Command-line Migration Tool 2.2 à partir du site Web de support de Layer 7.
2. Extrayez le fichier .jar.

Remarque : Le fichier JAR contient le fichier cmt2.jar, requis pour l'étape 3. Vous pouvez placer ce fichier n'importe où.

Procédez comme suit:

1. Effectuez l'une des opérations suivantes :
 - Sélectionnez File (Fichier), New (Nouveau), VS Image (Image de service virtuel), From Layer 7 (De Layer 7).
 - Cliquez avec le bouton droit de la souris sur le noeud racine du projet et sélectionnez Create New VS Image (Créer une image de service virtuel), From Layer 7 (De Layer 7).
 - Cliquez avec le bouton droit de la souris sur le dossier Virtual Services Images (Images de services virtuels) et sélectionnez Create New VS Image (Créer une image de service virtuel), Layer 7.

La fenêtre Virtual Service From Layer7 (Service virtuel à partir d'un fichier WSDL) s'ouvre.

2. Remplissez les champs Layer7 Connection Info (Informations de connexion de Layer 7).
3. Cliquez sur l'icône de dossier pour rechercher le fichier cmt2.jar et cliquez sur Get Layer7 Services (Obtenir les services Layer 7).

L'onglet Layer7 Services affiche les services disponibles.

4. Sélectionnez un service, puis cliquez sur Next (Suivant).
5. Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.

Acceptez les valeurs par défaut pour les autres champs de la fenêtre.

Remarque : Pour charger des paramètres à partir d'une image de service enregistrée précédemment, cliquez sur l'icône Load from File (Charger à partir d'un fichier) au bas de la fenêtre.

6. Cliquez sur Next (Suivant).

L'onglet Connection (Connexion) s'affiche.
7. Dans ce service, sélectionnez les opérations à virtualiser.

Par défaut, toutes les opérations sont sélectionnées.

8. Cliquez sur Next (Suivant).

Les options de protocoles de données côté demande/réponse s'ouvrent.

9. Sélectionnez **Web Services (SOAP) (Services Web (SOAP))**.


La liste Request Side Data Protocols (Protocoles de données côté demande) est préremplie avec les services Web (SOAP) et les gestionnaires de protocoles de données XML. La liste Response Side Data Protocols (Protocoles de données côté réponse) est automatiquement remplie avec le protocole de données de texte délimité.

10. Cliquez sur Next (Suivant).

11. Pour des informations sur la configuration du protocole de données de texte délimité, consultez la section [Protocole de données de texte délimité](#) (page 254). A l'issue de la configuration, cliquez sur Next (Suivant).

Dans la fenêtre suivante, l'image de service est générée et l'assistant a terminé.

12. Cliquez sur Finish (Terminer).

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

Le modèle de service virtuel vide est rempli par des étapes.

13. Enregistrez le modèle de service virtuel.

L'image de service générée est un service stub. L'image de service renvoie des réponses correctement formatées, mais il s'agit de valeurs par défaut.

Le modèle de service virtuel enregistré est le modèle déployé dans l'environnement de services virtuels.

Création d'une image de service à partir d'une paire réponse-demande

Vous pouvez générer une image de service Web virtuel à partir d'une paire réponse-demande de différentes manières :

- [A partir de l'interface utilisateur](#) (page 123)
- [A partir d'une ligne de commande](#) (page 125)

Vous pouvez utiliser des [fichiers sidecar](#) (page 126) pour personnaliser les métadonnées à partir des paires demande-réponse.

Création d'une image de service à partir de paires demande-réponse dans l'interface utilisateur


Cette section décrit la procédure de création d'une image de service virtuel à partir d'une paire demande-réponse à l'aide de l'interface utilisateur.

Procédez comme suit:

1. Cliquez avec le bouton droit de la souris sur l'icône VirtualServices (Services virtuels), Images et sélectionnez Create New VS Image (Créer une image de service virtuel), From Request/Response Pairs (A partir de paires demande-réponse).

La page Virtual Service From Request/Response Pairs (Service virtuel à partir des paires demande-réponse) s'ouvre.

2. Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.
3. Acceptez les valeurs par défaut des autres champs de la fenêtre et cliquez sur Next (Suivant).

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

4. Accédez au système de fichiers et recherchez le répertoire qui contient vos paires demande/réponse.

Les paires demande/réponse doivent être nommées à l'aide d'un identificateur unique, suivi de -req côté demande et de -rsp côté réponse, avec l'extension **.xml** ou **.txt**. Par exemple : **addUserObject-req.xml** et **addUserObject-rsp.xml**.

Remarque : Une demande peut avoir plusieurs réponses. Dans l'exemple précédent, les noms de fichier suivants auraient créé une demande avec trois réponses.

- addUserObject-req.xml
- addUserObject-rsp1.xml
- addUserObject-rsp2.xml
- addUserObject-rsp3.xml

Le VSE génère une transaction pour chaque paire demande-réponse dans le répertoire spécifié. Dans le cas de HTTP/S, les fichiers doivent contenir l'enveloppe et les en-têtes SOAP complets.

5. Précisez les informations suivantes :
 - Protocole de transport
 - Codage approprié
 - Utilisation ou non de paires demande-réponse binaires
6. Cliquez sur Configure (Configurer).

La fenêtre de configuration pour le protocole de transport sélectionné s'ouvre.

7. Entrez les informations de configuration pour la demande et cliquez sur Finish (Terminer).

La fenêtre Virtual Service From Request/Response Pairs (Service virtuel à partir des paires demande-réponse) s'ouvre.

8. Cliquez à nouveau sur Configure (Configurer) pour conserver les valeurs saisies.

Si vous sélectionnez un protocole différent, il sera présenté. Vous pouvez indiquer ses informations de configuration.


9. Cliquez sur Next (Suivant).

Le panneau Data Protocols (Protocoles de données) s'ouvre et contient les paires demande-réponse analysées et les protocoles de données conformes à la demande et la réponse définis sur leur valeur par défaut.

10. Apportez les modifications nécessaires ou ajoutez d'autres protocoles de données et cliquez sur Next (Suivant).

Les paires demande-réponse de service virtuel pour l'identification du jeton les conversations s'affichent. Seules les transactions sans état sont prises en charge par les paires demande-réponse.

11. Cliquez sur Next (Suivant).


Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

A l'issue du traitement de l'image de service virtuel, vous pouvez ouvrir l'image de service et le modèle de service virtuel.

Création d'une image de service à partir de paires demande-réponse dans la ligne de commande

Hormis la création d'une image de service à partir de paires réponse-demande à l'aide de l'interface utilisateur, vous pouvez également créer l'image via la ligne de commande ServiceImageManager.

Procédez comme suit:

1. Créez une image de service dans l'interface utilisateur.
2. Une fois l'image créée, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer), puis cliquez sur Finish (Terminer).

Les paramètres sont enregistrés dans un fichier portant l'extension **.vrs**.

3. Accédez au répertoire LISA_HOME\bin et entrez la commande suivante :

```
ServiceImageManager -vrs recording-session-file  
vsi-file=vsi-file --vsm_file=vsm-file --record
```

recording-session-file

Définit le chemin d'accès au fichier **.vrs** que vous avez créé précédemment.

vsi-file

Définit le fichier d'image de service créé.

vsm_file

Définit les fichiers de modèle de service virtuel créés.

Fichiers sidecar avec des paires demande-réponse

Lorsque VSE crée une image de service à partir de paires demande-réponse, le système peut utiliser un autre fichier avec les paires demande-réponse, un fichier sidecar. Un fichier sidecar est un fichier de propriétés dans lequel vous pouvez ajouter des paires clé-valeur qui doivent être ajoutées aux métadonnées de certaines demandes, réponses ou les deux. Vous pouvez ajouter plusieurs fichiers génériques pour toutes les demandes et les réponses.

Exemple :

Un répertoire nommé **soap** contient des fichiers de paires demande-réponse nommés **abc-req.xml** et **abc-rsp.xml**. Vous ajoutez les fichiers sidecar selon la convention d'attribution de nom **abc-req-meta.properties**, **abc-rsp-meta.properties**, etc. Si vous voulez ajouter une propriété pour les métadonnées de toutes les demandes et les réponses de ce répertoire, utilisez les noms de fichier **meta-req.properties** et **meta-rsp.properties**. Les entrées dans les fichiers sidecar de transaction (**abc-req-meta.properties** et **abc-rsp-meta.properties**) remplacent tout le contenu des fichiers sidecar globaux (**meta-req.properties** et **meta-rsp.properties**). Par conséquent, vous pouvez avoir un ensemble constitué de toutes les valeurs par défaut dans les fichiers globaux, puis remplacez les valeurs par défaut pour des transactions spécifiques à l'aide de fichiers sidecar de transaction.

Par exemple, si vous avez trois demandes/réponses (**abc1-req.xml/abc1-rsp.xml**, **abc2-req.xml/abc2-rsp.xml** et **abc3-req.xml/abc3-rsp.xml**), vous pouvez avoir un fichier sidecar commun, **rsp-meta.properties**, pour lequel :

Content-type=text/plain

et un fichier sidecar abc3, **abc3-rsp-meta.properties**, pour lequel :

Content-type=text/html

Dans l'image de service, les métadonnées pour les réponses **abc1** et **abc2** ont la propriété meta Content-type définie sur text/plain, mais la réponse **abc3** a la valeur text/html.

Création d'une image de service à partir d'un objet PCAP

Si vous utilisez le logiciel de capture de paquets, par exemple Wireshark, pour créer des journaux de trafic, le VSE peut utiliser ces journaux pour créer une image de service virtuel à partir d'un fichier de capture de paquets (PCAP).

Conditions préalables

1. Téléchargez le package binaire approprié pour votre système d'exploitation à partir de l'adresse suivante : <http://jnetpcap.com/download>.
2. Ajoutez le fichier **jnetpcap.jar** du package binaire au répertoire **LISA_HOME\lib** et la bibliothèque native jnetpcap au répertoire **LISA_HOME\bin**.

La bibliothèque native varie en fonction du système d'exploitation. Dans le cas de Windows, il s'agit de jnetpcap.dll.

La fonctionnalité PCAP est désormais configurée.


3. S'il est en cours d'exécution, redémarrez DevTest Workstation pour récupérer les changements apportés à la configuration.

Procédez comme suit:

1. Cliquez avec le bouton droit de la souris sur le dossier VirtualServices (Services virtuels), Images et sélectionnez Create New VS Image (Créer une image de service virtuel), From PCAP (A partir d'un objet PCAP).

Entrez un nom d'image de service et le nom d'un fichier de modèle de service virtuel.

Acceptez les valeurs par défaut pour les autres champs de la fenêtre.

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

2. (Facultatif) Pour ajouter de la documentation sur ce service virtuel, cliquez sur l'onglet Notes.
3. Cliquez sur Next (Suivant).

La fenêtre de protocole de données s'ouvre.

4. Cliquez sur Next (Suivant).
5. Pour sélectionner le fichier de capture de paquets à utiliser pour les entrées, saisissez le nom du fichier ou accédez au système de fichiers.
6. Sélectionnez HTTP/S comme protocole de transport et cliquez sur Configure (Configurer).

La fenêtre Virtual Service from PCAP Transport Protocol Configuration (Service virtuel à partir de la configuration de protocole de transport de PCAP) s'ouvre.

7. Entrez les options de configuration suivantes :

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Permet de définir le numéro du port cible écouté par le serveur. Si vous prévoyez de sélectionner un style d'authentification directe proxy, laissez ce champ vide.

Valeur par défaut : 80 (HTTP) et 443 (HTTPS)

Remarque : Les ports hôte et cible sont importants. Ils déterminent la méthode de correspondance de paquets appliquée par DevTest. Sélectionnez Gateway (Passerelle), puis entrez l'adresse IP et le port du serveur hébergeant le service à virtualiser. Un trafic réseau peut être généré pour tous les ordinateurs du sous-réseau sur lequel la capture a été effectuée, selon le mode de capture de PCAP. En indiquant une adresse IP et un port, vous pouvez filtrer les données à partir du fichier pour tous les paquets se déplaçant vers ou à partir d'une adresse IP et d'un port spécifiques. Puis, ces paquets sont recréés et constituent des flux TCP valides, qui suppriment les doublons, réorganisent les paquets dans l'ordre correct, etc. Cette tâche est effectuée pour vous par la pile TCP du système d'exploitation lors d'un enregistrement réel. Ces flux sont relus sur le protocole réel (HTTP) et, dans le cadre de ce protocole, les données arrivent sur un flux TCP valide.

Recorder passthru style (Style d'authentification directe de l'enregistreur)

Indique le comportement de l'enregistreur d'image de service virtuel lors de l'enregistrement. Sélectionnez **Gateway (Passerelle)**.

Use SSL to server (Utiliser une connexion SSL au serveur)

Permet de spécifier si DevTest utilise une connexion HTTPS pour envoyer la demande au serveur.

- Si cette option est **sélectionnée**, DevTest envoie une demande HTTPS (secured layer) au serveur.

Si vous sélectionnez Use SSL to server (Utiliser une connexion SSL au serveur) sans sélectionner Use SSL to client (Utiliser une connexion SSL au client), DevTest utilisera une connexion HTTP pour l'enregistrement. DevTest enverra alors ces demandes au serveur à l'aide d'une connexion HTTPS.

- Si l'option est **désélectionnée**, DevTest enverra une demande HTTP au serveur.

Use SSL to client (Utiliser une connexion SSL au client)

Spécifie si un référentiel de clés personnalisé doit être utilisé pour lire une demande SSL à partir d'un client. Cette option est activée uniquement si l'option Use SSL to server (Utiliser une connexion SSL au serveur) a été sélectionnée.

Valeurs :

- **Option sélectionnée** : vous pouvez spécifier un référentiel de clés client personnalisé et une phrase secrète. Si ces paramètres sont saisis, ils seront utilisés, au lieu des valeurs par défaut codées de manière irréversible.
- **Option désélectionnée** : vous ne pouvez pas spécifier un référentiel de clés client personnalisé et une phrase secrète.

SSL keystore file (Fichier de référentiel de clés SSL)

Spécifie le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Spécifie le mot de passe associé au fichier de référentiel de clés sélectionné.

Remarque : Pour plus d'informations sur la configuration du VSE dans un environnement SSL bidirectionnel, reportez-vous à la section [Virtualisation des connexions SSL bidirectionnelles](#) (page 144).

Allow duplicate specific transactions (good for NTLM) (Permettre la duplication des transactions (approprié pour NTLM))

Spécifie si des transactions dupliquées spécifiques doivent être enregistrées.

8. Cliquez sur Finish (Terminer) pour revenir à la fenêtre précédente.
9. Pour démarrer l'enregistrement, cliquez sur Next (Suivant).

Création d'une image de service à partir d'un enregistrement

L'enregistreur d'image de service virtuel génère des images de service, qui doivent représenter vos enregistrements.

Procédez comme suit:

1. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des tâches suivantes :

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



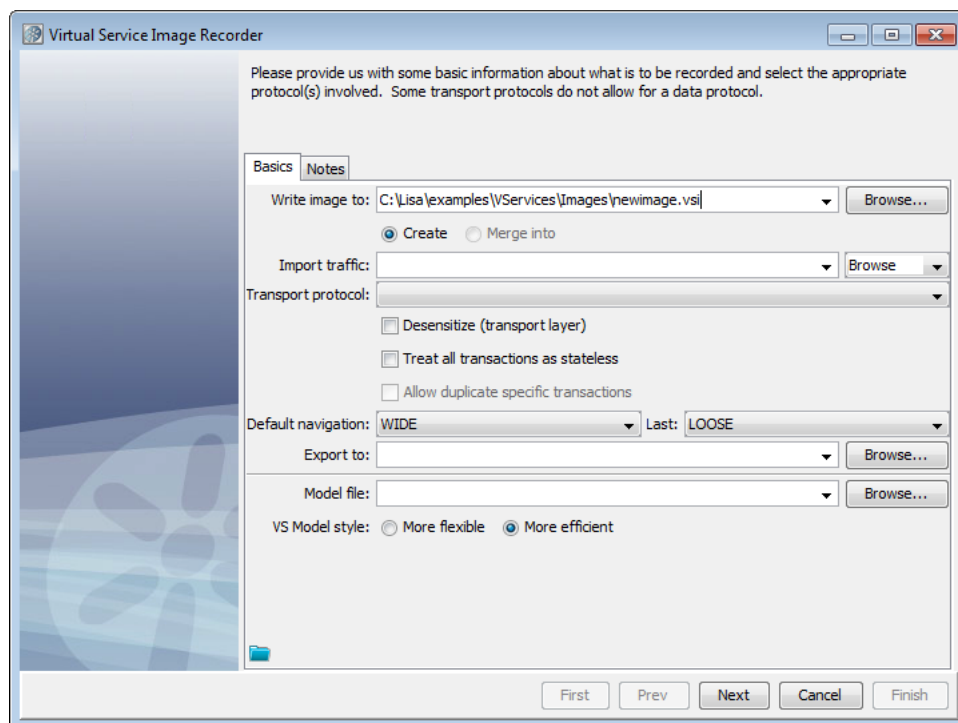
- Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image (Créer une image de service virtuel), by Recording (A l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

2. Remplissez les onglets de l'enregistreur d'image de service virtuel suivants, de manière appropriée :
 - [Notions de base :](#) (page 131)
 - [Protocoles de données](#) (page 135)
 - [Enregistrement par protocole de transport](#) (page 137) (fournit des instructions sur chaque méthode d'enregistrement d'images de service virtuel)

Onglet Basics (Paramètres de base)

L'onglet Basics (Paramètres de base) est la première fenêtre de l'assistant Virtual Service Image Recorder (Enregistreur d'image de service virtuel) et fournit le nom, le protocole et les options de navigation pour l'image. Entrez les informations requises dans les champs, selon le protocole utilisé. Toutes les fenêtres ultérieures sont spécifiques au protocole. Pour plus d'informations sur les protocoles, reportez-vous à la rubrique [Enregistreur de l'image de service virtuel - Protocoles de transport](#) (page 137).



Les options de l'onglet Basics (Paramètres de base) sont les suivantes :

Write image to (Emplacement d'écriture de l'image)

Indique un nom d'image de service unique.

Le chemin d'accès à l'image de service virtuel est remplacé par défaut par celui du projet ouvert lorsque DevTest Workstation a démarré. Si vous changez de projet, le chemin d'accès à l'image de service virtuel est toujours remplacé par défaut par celui du projet ouvert lorsque DevTest Workstation a démarré.

Import traffic (Importer un fichier de trafic)

Indique d'importer un fichier XML de trafic brut ou conversationnel. Si aucun fichier de ce type n'existe, ce champ peut être vide. Si un fichier est spécifié, les transactions dans le document XML référencé seront fusionnées avec celles issues de l'enregistrement qui s'ensuit.

Protocole de transport

Spécifie le protocole de transport à utiliser. Pour plus d'informations, reportez-vous à la rubrique [Enregistreur de l'image de service virtuel - Protocoles de transport](#) (page 137).

Desensitize (Désensibiliser)

Lors de l'enregistrement, permet de tenter de reconnaître des données sensibles et de les remplacer par des valeurs aléatoires. Pour plus d'informations, consultez la section [Désensibilisation des données](#) (page 375).

Treat all transactions as stateless (Traiter toutes les transactions comme des transactions sans état)

Cette option est fournie pour des situations spécifiques dans lesquelles vous voulez traiter toutes les transactions enregistrées comme étant sans état. Dans la plupart des cas, laissez cette case à cocher désélectionnée.

Allow duplicate specific transactions (Permettre la duplication des transactions)

Indique si DevTest peut répondre plusieurs fois au même appel tout en sélectionnant une réponse différente. La correspondance de tourniquet se produit uniquement si cette case à cocher est sélectionnée. Cette option est désactivée pour les protocoles de transport qui n'autorisent pas la duplication de transactions spécifiques.

Default navigation (Navigation par défaut)

Indique la tolérance de navigation qui détermine l'emplacement dans l'arborescence des conversations dans lequel un modèle de service virtuel recherche une transaction correspondant à la transaction spécifiée. Sélectionnez la tolérance de navigation par défaut pour tous les éléments sauf les dernières transactions (terminales).

Valeurs :

- CLOSE : recherche les enfants de la transaction actuelle
- WIDE : recherche CLOSE comprenant la transaction actuelle, les frères et les nièces/neveux des transactions actuelles
- LOOSE : recherche WIDE incluant le parent et les frères de la transaction actuelle, puis les enfants de la transaction de démarrage Si aucune des recherches ne trouve de correspondance, les transactions de démarrage de toutes les conversations seront vérifiées, ce qui revient à effectuer une recherche dans l'ensemble de la conversation.

Par défaut : WIDE

Last (Dernier)

Indique la tolérance de navigation qui détermine l'emplacement dans l'arborescence des conversations dans lequel un modèle de service virtuel recherche une transaction correspondant aux dernières transactions (feuilles).

Valeurs :

- CLOSE : recherche les enfants de la transaction actuelle

- **WIDE** : recherche CLOSE comprenant la transaction actuelle, les frères et les nièces/neveux des transactions actuelles
- **LOOSE** : recherche WIDE incluant le parent et les frères de la transaction actuelle, puis les enfants de la transaction de démarrage. Si aucune des recherches ne trouve de correspondance, les transactions de démarrage de toutes les conversations seront vérifiées, ce qui revient à effectuer une recherche dans l'ensemble de la conversation.

Par défaut : LOOSE

Export to (Emplacement d'exportation)

Spécifie le chemin complet d'un fichier dans lequel vous voulez journaliser le trafic brut. Lorsqu'un fichier est spécifié, chaque transaction envoyée à l'enregistreur (à partir du protocole de transport pendant l'enregistrement réel ou à partir du processus d'importation) sera écrite dans ce fichier. Vous pouvez capturer une session d'enregistrement pour l'importer ultérieurement, mais également la réutiliser pendant l'ajustement des détails du protocole de données.

Model file (Fichier de modèle)

Saisissez le chemin complet de votre fichier de modèle de service virtuel pour cette image de service. Si un nom de fichier est fourni dans ce champ, l'enregistreur générera un modèle de service virtuel automatiquement. Le style du modèle s'applique uniquement si un modèle de service virtuel est demandé.

VS Model style (Style du modèle de service virtuel)


Spécifie si un modèle de service virtuel, y compris les étapes de préparation, doit être généré.

Valeurs :

More flexible (Flexible) : permet d'inclure la procédure de préparation (donnant lieu à un modèle en cinq étapes dans le cas d'un protocole HTTP/S).

More efficient (Efficace) : la procédure de préparation est exclue (donnant lieu à un modèle en trois étapes dans le cas d'un protocole HTTP/S).

Valeur par défaut : More flexible

Remarque : Pour charger des paramètres à partir d'une image de service précédemment enregistrée, cliquez sur l'icône Load from File (Charger à partir d'un fichier)  au bas de la fenêtre.

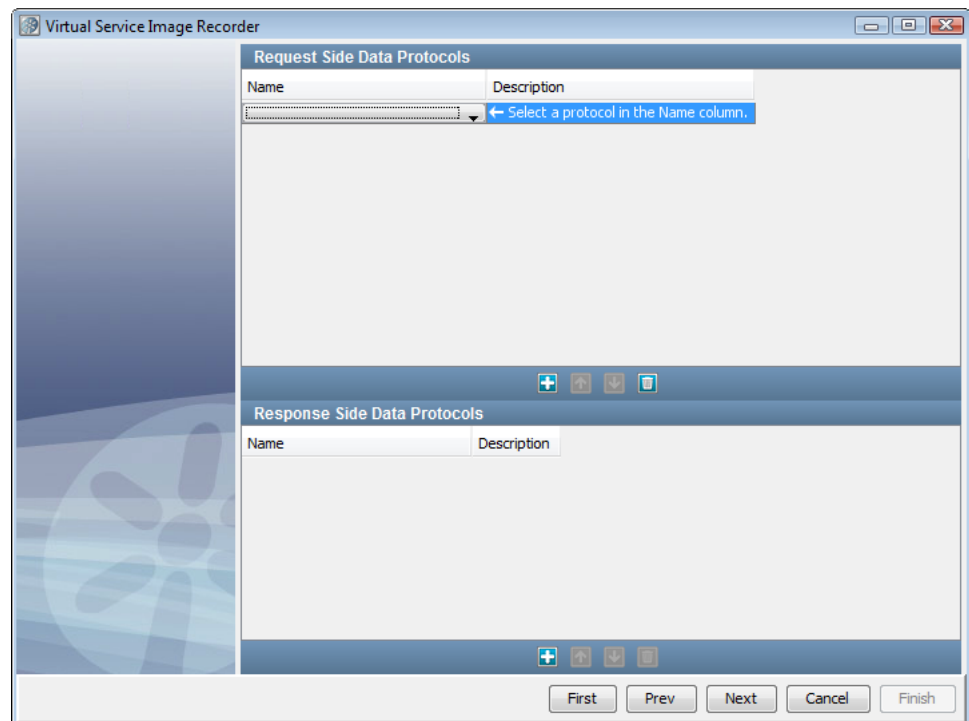
L'onglet Notes permet de créer de la documentation pour cette image de service.

Remarque : Si vous importez un fichier de trafic brut dans l'enregistreur de VSE, puis cliquez sur le bouton Back (Précédent) pour revenir au premier panneau, l'enregistreur de VSE importe de nouveau le fichier de trafic et traite deux fois le nombre de transactions.

De plus, si vous effectuez la procédure sans désigner au préalable un fichier de trafic et que vous accédez au panneau d'enregistrement, revenez au premier panneau et sélectionnez le fichier de trafic ; lorsque vous accédez de nouveau à l'enregistreur de VSE, il ne traite aucune transaction.

Protocoles de données

La deuxième fenêtre de l'assistant de l'enregistreur d'image de service virtuel permet de saisir des informations sur les protocoles de données pour le service virtuel.



L'enregistreur peut utiliser les gestionnaires de protocoles de données suivants. Sélectionner le protocole de données approprié lui permet d'analyser les informations qu'il enregistre pour distinguer les conversations et identifier les transactions qui leur appartiennent. Vous pouvez placer ces gestionnaires de protocole de données sous forme de chaîne pour les utiliser ensemble.

Détection des transactions de hachage automatique

Permet d'identifier un message par le code de hachage des données. Même une légère modification des données entraîne un changement du code de hachage, ce qui rend toutes les demandes uniques. Ce protocole est utile lorsque vous exécutez le même petit ensemble de demandes au niveau du service.

Protocole de données de copybook CICS

Répartit la demande enregistrée dans ses fragments de conteneur respectifs. Puis, il envoie chaque fragment vers le protocole de données de copybook et ajoute le fichier XML correspondant.

Protocole de données de copybook

Convertit le texte copybook en XML.

Protocole de données de copybook CTG

Répartit la demande enregistrée dans ses fragments de conteneur respectifs. Puis, il envoie chaque fragment vers le protocole de données de copybook et ajoute le fichier XML correspondant.

Désensibilisation des données

Tente de reconnaître les données sensibles et de substituer les valeurs aléatoires lors de l'enregistrement. Pour plus d'informations, consultez la section [Désensibilisation des données](#) (page 375).

Protocole de données de texte délimité

Convertit les chaînes délimitées en XML.

Protocole de données DRDA

Permet de convertir des charges utiles DRDA au format XML lors de l'enregistrement, pour améliorer l'alignement avec les fonctionnalités de DevTest natives, la lisibilité et la prise en charge de données dynamiques. Les réponses sont reconverties au format natif lors de la lecture.

Protocole de données EDI X12

Convertit les documents ANSI X12 EDI en une représentation XML dans le corps de la demande.

Analyseur de charge utile XML générique

Identifie si les demandes et les réponses sont des chaînes XML. Si vous utilisez ce protocole, vous pouvez identifier des variables dans les messages XML utilisés par l'enregistreur.

Protocole de données JSON

Convertit les données JSON en un format XML équivalent et les données XML en un format JSON.

Copieur de données de la demande

Copie les données de la demande entrante actuelle dans le contexte de test actuel.

Gestionnaire de données de demande

Manipule les demandes de VSE pendant leur enregistrement ou leur lecture.

Protocole de données REST

Analyse les demandes HTTP conformes au style d'architecture REST.

Protocole de données prenant en charge les scripts

Fournit des scripts côté demande, côté réponse ou les deux, pour traiter la demande ou la réponse.

Protocole de données SWIFT

Convertit les messages SWIFT en un format XML équivalent et les données XML en messages SWIFT.

Web Services Bridge (Pont des services Web)

S'applique uniquement à l'exemple DevTest Travel. Vous pouvez ignorer ce protocole de données, car il s'applique uniquement à l'exemple et n'est pas utile dans un cas général.

Web Services (SOAP) (Services Web (SOAP))

S'applique lors de l'utilisation d'un client de service Web.

Web Services (SOAP Headers) (Services Web (en-têtes SOAP))

Convertit des éléments d'en-tête SOAP en tant arguments de la demande.

WS-Security Request (Demande WS-Security)

Élimine la sécurité de la demande SOAP avant de l'envoyer vers la structure de virtualisation et applique la sécurité aux réponses SOAP sortantes.

Protocole de données XML

Convertit un document XML en un type d'opération/arguments approprié pour la demande.

Remarque : JDBC ne permet l'utilisation d'un protocole de données.

Pour plus d'informations sur ces protocoles de données, reportez-vous à la section Utilisation des protocoles de données.

Pour plus d'informations sur l'utilisation d'un protocole de données dynamiques, reportez-vous à la section [Analyseur de charge utile XML générique](#) (page 262).

Protocoles de transport

Chaque protocole de transport disponible est décrit dans les sections suivantes.

[HTTPS](#) (page 138)

[IBM WebSphere MQ](#) (page 148)

[JMS](#) (page 154)

[Standard JMS](#) (page 163)

[Java](#) (page 168)

[JDBC](#) (page 171)

[TCP](#) (page 180)

[Enregistrement de CICS \(LINK DTP MRO\)](#) (page 186)

[Enregistrement d'images de la passerelle de transaction CICS \(ECI\)](#) (page 206)

[Enregistrement d'images de service IMS Connect](#) (page 208)

[Enregistrement de RFC SAP via JCo](#) (page 212)


[Record JCo IDoc \(Enregistrer une image de service virtuel de service Web\)](#) (page 214)

[Opaque Data Processing \(Traitement de données opaques\)](#) (page 217)

HTTPS

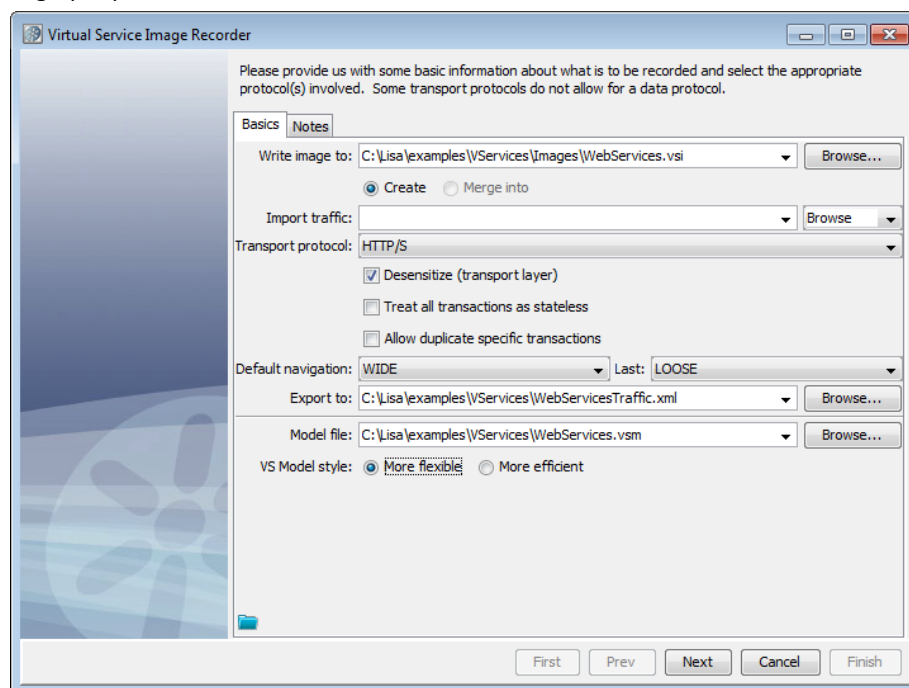
Procédez comme suit:

1. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :

- Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
- Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

2. Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



3. Cliquez sur Next (Suivant).

La fenêtre de l'assistant s'ouvre.

4. Entrez les informations du port et de l'hôte pour cette étape.

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest. En général, vous sélectionnez le port 8001, mais vous pouvez utiliser un autre numéro de port.

Target host (Hôte cible)

Indique le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté. Si vous prévoyez de sélectionner un style d'authentification directe proxy, laissez ce champ vide.

Target port (Port cible)

Permet de définir le numéro du port cible écouté par le serveur. Si vous prévoyez de sélectionner un style d'authentification directe proxy, laissez ce champ vide.

Valeur par défaut : 80 (HTTP) et 443 (HTTPS)

Recorder pass-through style (Style d'authentification directe de l'enregistreur)

Indique le comportement de l'enregistreur d'image de service virtuel lors de l'enregistrement. Les options disponibles sont Gateway (Passerelle) et Proxy. Si vous sélectionnez Proxy, le contenu des champs Target host (Hôte cible) et Target port (Port cible) sera effacé et les champs seront désactivés. Ce choix influe sur la connexion du client en mode d'enregistrement.

- Si l'écoute de l'enregistreur d'image de service virtuel est effectuée en mode Gateway (Passerelle), le client devra envoyer des requêtes HTTP directement à l'enregistreur et non au serveur. Si le client est un navigateur, l'URL contiendra l'hôte et le port de l'enregistreur, au lieu de l'hôte et du port du serveur.
- Si l'écoute de l'enregistreur d'image de service virtuel est effectuée en mode proxy, le client devra spécifier l'hôte de l'enregistreur et le port comme proxy. Si le client est un navigateur, l'URL contiendra l'hôte et le port du serveur. Les paramètres de proxy doivent être définis de sorte à router la demande dans l'enregistreur.

La plupart des clients HTTP contiennent un paramètre permettant de ne pas utiliser de proxy pour l'hôte local. Si l'enregistreur d'image de service virtuel est exécuté sur l'hôte local en mode proxy, désactivez ce paramètre de sorte que le trafic soit correctement transmis par l'enregistreur.

Ne modifiez pas de paramètre d'en-tête d'hôte reçu à partir d'un client.

Spécifie si l'authentification directe est activée pour la valeur du paramètre Host. Cette option est uniquement disponible lors de l'enregistrement en mode Gateway (Passerelle). L'option d'authentification directe indique à l'enregistreur de ne pas récrire le paramètre d'en-tête Host lors du renvoi du trafic au terminal cible.

Use SSL to server (Utiliser une connexion SSL au serveur)

Permet de spécifier si DevTest utilise une connexion HTTPS pour envoyer la demande au serveur.

- Si cette option est **sélectionnée**, DevTest envoie une demande HTTPS (secured layer) au serveur.

Si vous sélectionnez Use SSL to server (Utiliser une connexion SSL au serveur) sans sélectionner Use SSL to client (Utiliser une connexion SSL au client), DevTest utilisera une connexion HTTP pour l'enregistrement. DevTest enverra alors ces demandes au serveur à l'aide d'une connexion HTTPS.

Si l'option est **désélectionnée**, DevTest enverra une demande HTTP au serveur.

Use SSL to client (Utiliser une connexion SSL au client)

Spécifie si un référentiel de clés personnalisé doit être utilisé pour lire une demande SSL à partir d'un client. Cette option est activée uniquement si l'option Use SSL to server (Utiliser une connexion SSL au serveur) a été sélectionnée.

Valeurs :

- **Option sélectionnée** : vous pouvez spécifier un référentiel de clés client personnalisé et une phrase secrète. Si ces paramètres sont saisis, ils sont utilisés au lieu des valeurs par défaut codées de manière irréversible.

Option désélectionnée : vous ne pouvez pas spécifier un référentiel de clés client personnalisé et une phrase secrète.

SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

Remarque : Pour plus d'informations sur la configuration du VSE dans un environnement SSL bidirectionnel, reportez-vous à la section [Virtualisation des connexions SSL bidirectionnelles](#) (page 144).

5. Cliquez sur Next (Suivant).

L'enregistreur d'image de service virtuel lance l'enregistrement du trafic. Le port affecté et la cible du service s'affichent dans cette fenêtre.

6. Pour envoyer les demandes au serveur routé via l'enregistreur d'image de service virtuel pour le lancement de l'enregistrement du trafic, utilisez votre client HTTP.

A mesure que l'enregistreur d'image de service virtuel enregistre des transactions, elles sont reflétées dans les statistiques d'affichage dynamique dans la partie inférieure de la fenêtre. Les options et les statistiques d'affichage dynamique incluent les éléments suivants :

Conversation count (Nombre de conversations)

Indique le nombre de conversations enregistrées.

Total transactions (Nombre total de transactions)

Indique le nombre de transactions enregistrées.

Clear (Effacer)

Pour effacer des transactions actuellement enregistrées, cliquez sur ce bouton.

7. A l'issue de l'enregistrement, cliquez sur Next (Suivant) pour passer à l'étape suivante.

Si vous cliquez sur Next (Suivant) et qu'aucune transaction n'est enregistrée, un message d'erreur s'affiche. Cliquez sur OK pour continuer l'enregistrement.

Remarque : Si aucune transaction n'est enregistrée, un conflit de ports peut survenir. Le client envoie des transactions à l'application, non à l'enregistreur d'image de service virtuel. Si un autre service utilise ce port, arrêtez-le ou modifiez le paramètre de port afin d'éviter tout conflit.

L'onglet Transactions affiche une liste des dernières transactions enregistrées. Dans cette liste de transactions, vous pouvez double-cliquer sur l'une d'elles pour afficher une boîte de dialogue indiquant son contenu.

8. Vérifiez le chemin de base et effectuez une mise à jour le cas échéant.
9. Pour obtenir une étape d'association au port avant le traitement des demandes, sélectionnez la case à cocher A separate bind-to-port step is required (Une étape d'association au port distincte est requise.).
10. Cliquez sur Next (Suivant).
11. Dans la fenêtre suivante, ne sélectionnez aucune valeur pour le protocole de données et cliquez sur Next (Suivant).
12. Si aucune conversation n'a été détectée pendant le processus d'enregistrement, sélectionnez des transactions de démarrage de conversations. Pour les conversations basées sur un jeton, spécifiez l'emplacement de jetons. Utilisez la zone Token Identification (Identification du jeton) dans l'enregistreur d'image de service virtuel. Sélectionnez les transactions de démarrage de conversations et identifiez les jetons de session. Pour désigner la transaction de démarrage de conversations **getNewToken** répertoriée comme démarreur de conversations, sélectionnez-la et cliquez sur la flèche bleue.

Les composants de l'étape se présentent comme suit :

Conversation Starter Transactions (Transactions de démarrage de conversations)

Répertorie les transactions sélectionnées comme démarreurs de conversations. Pour déplacer une transaction vers la liste Remaining Transactions (Transactions restantes) (si vous ne voulez pas l'utiliser comme démarreur de conversations), sélectionnez-la et cliquez sur la flèche.

Remaining Transactions (Transactions restantes)

Répertorie les transactions enregistrées. Pour déplacer une transaction vers la liste Conversation Starter Transactions (Transactions de démarrage de conversations), sélectionnez-la et cliquez sur la flèche.

Icône Plus

Permet de sélectionner dans la liste toutes les transactions (démarreurs de conversations ou restantes) semblables à une transaction sélectionnée. Pour déplacer toutes les transactions sélectionnées, utilisez la flèche appropriée.

Conversation count (Nombre de conversations)

Affiche le nombre de conversations dans l'enregistrement. Lors de la création de conversations, ce nombre augmente.

Force stateless (Sans état obligatoire)

Dans la liste Remaining Transactions (Transactions restantes), sélectionnez les transactions qui doivent être sans état et cochez la case. Par exemple, vous pouvez décider qu'une transaction qui inclut une image doit rester sans état, même si elle contient un jeton de démarrage de conversations.

Stateless Transactions (Transactions sans état)

Cliquez pour afficher une liste de toutes les transactions qui demeurent sans état, selon les conversations identifiées dans ce panneau. Vous pouvez utiliser la liste pour vérifier que vous avez identifié toutes les transactions de démarrage de conversations.

Save (Enregistrer)

Cliquez sur cette option pour enregistrer les transactions brutes enregistrées. Cliquez sur Browse (Parcourir) pour accéder à l'emplacement d'enregistrement du fichier. Avant de lancer un nouvel enregistrement, vous pouvez importer l'enregistrement de trafic brut dans l'onglet Basics (Paramètres de base).

Response (Réponse)

Ce champ identifie les réponses à examiner pour la transaction actuellement sélectionnée. En général, 1 est la seule option.

Look in (Rechercher dans)

Ce champ identifie la partie de la réponse que vous voulez afficher lors de la recherche de jetons de conversation. La liste déroulante contient une entrée pour chaque entrée de métadonnées dans la réponse, plus une pour le corps de la réponse.

Token Identification area (Zone d'identification du jeton)


En fonction de la transaction et de la réponse sélectionnées, le contenu de la section Look in (Rechercher dans) de la réponse s'affiche ici.

- Pour marquer une partie du texte comme jeton de conversation, sélectionnez le texte et cliquez sur l'icône en forme de tampon à encre rouge. Le texte est alors mis en surbrillance en jaune.
- Pour marquer le texte comme n'étant plus un jeton de conversation, marquez une autre partie du texte ou cliquez sur Erase (Effacer).
- Après avoir marqué un jeton, vous pouvez utiliser l'icône Search (Rechercher) pour rechercher des transactions similaires et marquer leur jeton. Pour ouvrir une boîte de dialogue dans laquelle vous pouvez sélectionner du texte (tel que des balises XML) qui lie le jeton de conversation, cliquez sur l'icône Search (Rechercher). Pour spécifier le texte de début et le texte de fin à rechercher, utilisez cette méthode.

13. Cliquez sur Next (Suivant).

Lors du post-traitement, l'enregistreur d'image de service virtuel affiche le statut du traitement. Dans le cadre de la préparation de l'écriture du fichier .vsi, l'enregistreur vérifie que les corps des demandes et des réponses sont du texte, si elles sont marquées comme tel. Sinon, le type devient binaire.

L'enregistreur exécute le post-traitement de l'enregistrement.

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

14. Cliquez sur Finish (Terminer) pour stocker l'image.

15. Vérifiez et enregistrez le modèle de service virtuel dans DevTest Workstation.

Virtualisation de connexions SSL bidirectionnelles

Pour virtualiser une connexion SSL bidirectionnelle, DevTest doit contenir l'un des éléments suivants :

- Le référentiel de clés client et le référentiel de clés de serveur
- Le référentiel de clés client et le référentiel de clés DevTest (consultez **webreckeys.ks** dans le répertoire d'installation). Extrayez le certificat DevTest du référentiel de clés DevTest et ajoutez-le au référentiel d'approbations client. Le certificat DevTest est un certificat autosigné, non un certificat publié par une autorité de certification. Cette solution fonctionne uniquement lorsque le client accepte les certificats autosignés.

Les deux cas donnent lieu à deux référentiels de clés : un référentiel de clés client et un référentiel de clés de serveur (ou référentiel de clés DevTest).

Configurez les propriétés SSL dans le fichier local.properties (situé dans le répertoire d'installation) pour utiliser le référentiel de clés client comme suit :

ssl.client.cert.path

Définit le chemin vers votre référentiel de clés ; par exemple :

c:/mykeystore.jks.

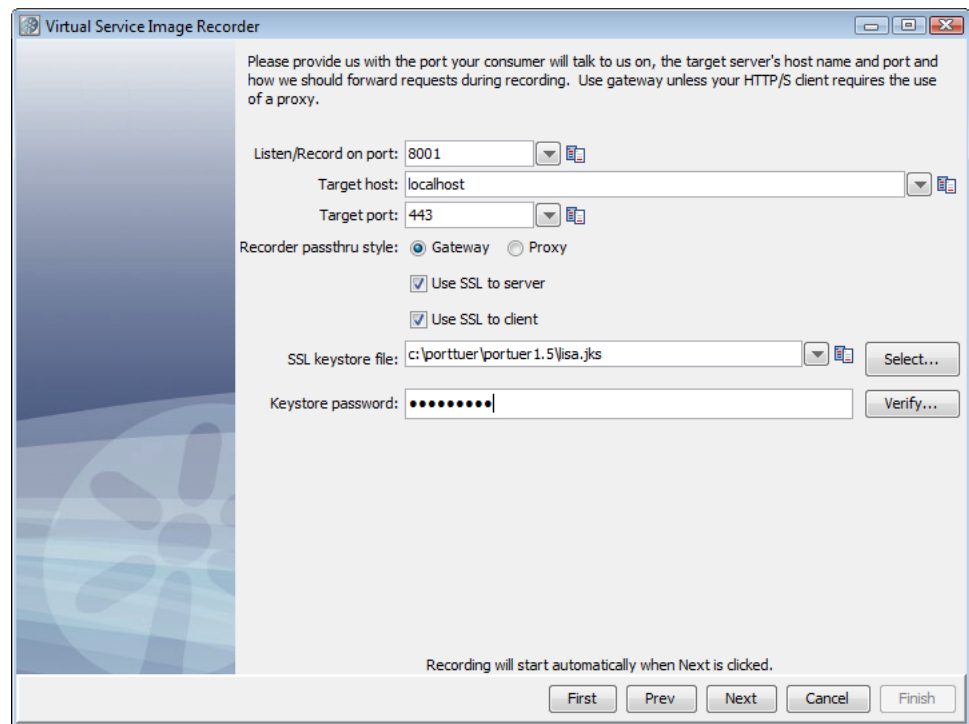
ssl.client.cert.pass

Définit le mot de passe de votre référentiel de clés.

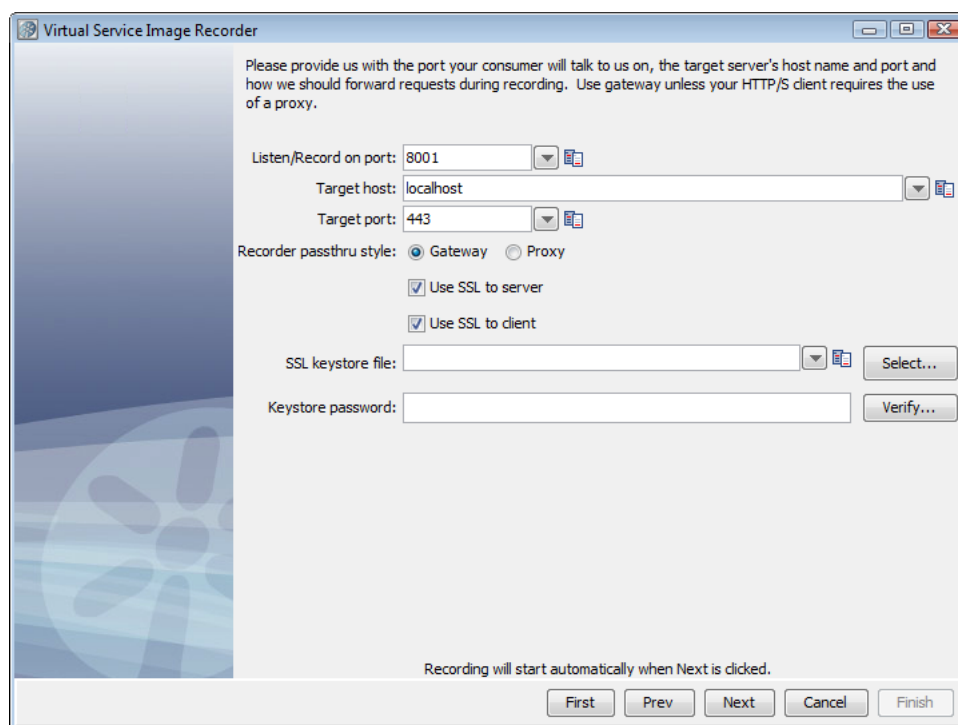
ssl.client.key.pass

Définit le mot de passe de votre certificat.

Démarrez l'enregistreur du VSE et configurez-le de sorte à utiliser la connexion SSL bidirectionnelle. Si vous utilisez un client et un référentiel de clés de serveur, votre enregistreur sera similaire au graphique suivant :

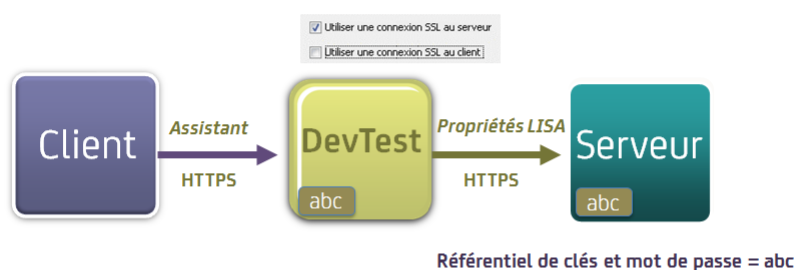


Si vous utilisez un référentiel de clés DevTest au lieu du référentiel de clés de serveur réel, il n'est pas nécessaire d'indiquer son chemin d'accès. Le référentiel de clés DevTest est utilisé par défaut et vous devez configurer votre enregistreur de manière similaire au graphique suivant :

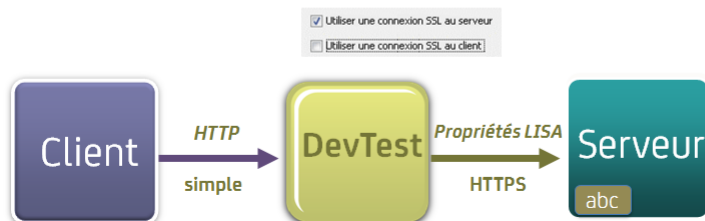


Les diagrammes suivants illustrent la virtualisation de connexions unidirectionnelles et bidirectionnelles SSL.

Connexion SSL bidirectionnelle DevTest




Connexion SSL unidirectionnelle LISA



IBM WebSphere MQ

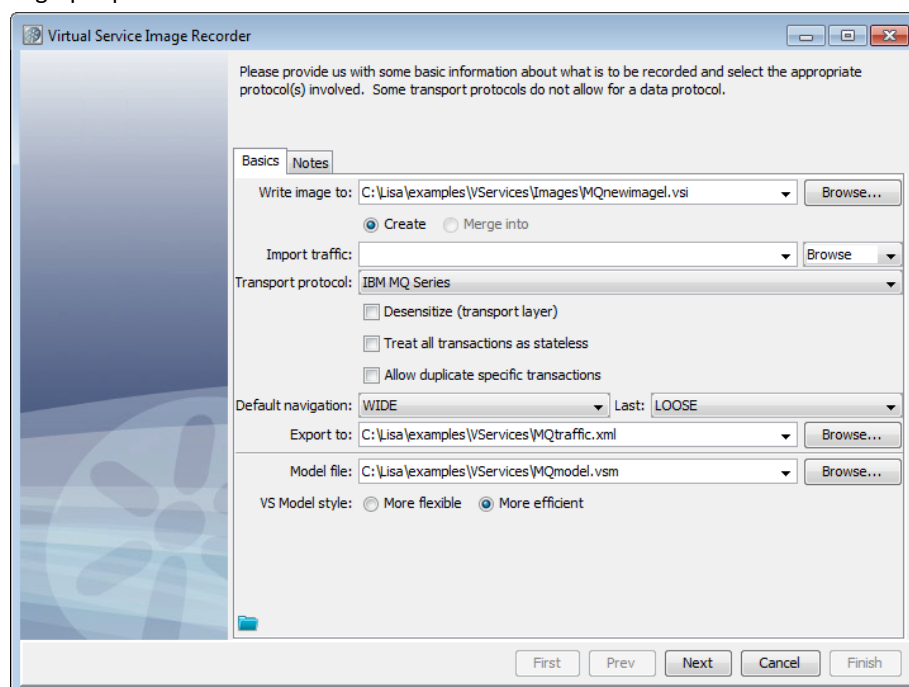
Conditions requises : L'utilisation de DevTest avec cette application requiert que la mise à disposition d'un ou plusieurs fichiers pour DevTest. Pour plus d'informations, consultez la section Conditions requises pour les fichiers des applications tierces de la rubrique *Administration*.

Procédez comme suit:

- Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :
 - Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
 - Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

- Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



- Cliquez sur Next (Suivant).

L'étape de sélection de mode d'enregistrement s'ouvre, avec une des options suivantes sélectionnées :

Proxy Mode (Mode Proxy)

Le mode Proxy est le seul mode d'enregistrement réellement pris en charge par VSE. Le mode Proxy fournit les options suivantes :

- Generate service using the Live queues (Générer le service à l'aide de files d'attente dynamiques)
- Generate service using the Proxy queues (Générer le service à l'aide de files d'attente de proxy)

Le mode Proxy permet de configurer et d'utiliser des files d'attente de proxy dans le bus de messagerie. L'application cliente est configurée de sorte à envoyer les messages vers ces destinations de proxy, puis DevTest les enverra vers la destination réelle une fois enregistrés. Le même comportement s'applique côté réponse. DevTest est configuré de sorte à écouter sur la destination de réponse réelle, récupérer le message, puis le transmettre à la destination de proxy de réponse. Ce mode peut se comporter de manière différente si vous avez activé les destinations temporaires et que les réponses sont automatisées.

Import Mode (Mode Importation)

Ce mode d'enregistrement des services virtuels est disponible si vous spécifiez un fichier de trafic brut dans le champ Import traffic (Importer un fichier de trafic) dans la fenêtre d'enregistrement initiale. Dans ce mode d'enregistrement, vous pouvez configurer des informations supplémentaires sur les files d'attente de demande et de réponse détectées dans le fichier de trafic brut. Vous pouvez également ignorer toutes les étapes de réponse et de demande.

4. Si vous enregistrez en mode Proxy, remplissez les champs suivants :

Generate service using the Live queues / Proxy queues (Générer le service à l'aide de files d'attente dynamiques/de proxy)

Indique si les files d'attente dynamiques ou de proxy doivent être utilisées lors de la génération du dernier modèle de service virtuel et de l'image.

Max Pending Transactions (Nombre maximum de transactions en attente)

Indique le nombre maximum d'écouteurs de réponse exécutés dans chaque file d'attente de réponses. Ces écouteurs de réponse sont exécutés lorsqu'une transaction est en attente. Lorsque le nombre de transactions en attente dépasse le nombre maximum, la plus ancienne transaction est automatiquement fermée. Si vous saisissez 0 dans ce champ, il n'existera pas de nombre maximum.

Disable Multiple Responses (Désactiver les réponses multiples)

Indique si plusieurs réponses doivent être prises en charge dans chaque transaction. Par défaut, une transaction reste en attente après la première réponse et attend que d'autres réponses s'y ajoutent. Si cette option est sélectionnée, la transaction est automatiquement fermée après la première réponse. Si de nombreuses transactions entrent rapidement, cette option peut renforcer les performances, notamment pour MQ.

Correlation

Contient les schémas de corrélation potentiels pour mettre en corrélation les demandes et les réponses de transactions. JMS est asynchrone, c'est-à-dire que les demandes et les réponses sont reçues séparément. Cette liste déroulante vous permet de définir dans l'enregistreur de VSE la demande à associer à une réponse. Le champ Correlation comprend les options suivantes :

- Sequential (Séquentiel) : toutes les réponses sont associées à la dernière demande reçue, dans l'ordre chronologique. Il n'y a aucun schéma de corrélation et l'enregistreur MQ de VSE place un verrou exclusif en lecture sur la file d'attente de réponses dynamiques pour qu'aucun autre écouteur MQ ne puisse utiliser ses messages de réponse.

Lorsque vous spécifiez un schéma de corrélation, via les champs Correlation ID (ID de corrélation) ou Message ID to Correlation ID (De l'ID de message à l'ID de corrélation), l'enregistreur MQ de VSE suppose que les autres écouteurs présents sur la file d'attente de réponses dynamiques utilisent également un schéma de corrélation. Si tous les écouteurs présents sur la file d'attente de réponses dynamiques utilisent des schémas de corrélation, l'enregistreur MQ de VSE peut maintenir ses réponses à part, sans devoir recourir à un verrou exclusif en lecture et il place l'indicateur d'entrée partagé en tête de la file d'attente.

- Correlation ID (ID de corrélation) : la demande et la réponse doivent avoir le même ID de corrélation.
- Message ID to Correlation ID (De l'ID de message à l'ID de corrélation) : l'ID de message de la demande doit être identique à l'ID de corrélation de la réponse.
- Message ID (ID du message) : la demande et la réponse doivent avoir le même ID de message.

5. Si vous enregistrez en mode import (Importation), remplissez les champs suivants :

Client Mode (Mode client)

Permet de spécifier la méthode d'interaction avec le serveur WebSphere MQ.

Valeurs :

- Native Client (Client natif) : implémentation purement Java utilisant des API IBM
- JMS : implémentation purement Java basée sur les spécifications JMS. Si vous voulez utiliser cette implémentation, il est recommandé d'utiliser le protocole de transport JMS au lieu de MQ.
- Bindings (Liaisons) : cette option requiert l'accès aux bibliothèques natives d'une installation de client WebSphere MQ. Vérifiez que ces bibliothèques sont accessibles par l'environnement d'exécution de l'application DevTest. Dans la plupart des cas, la présence de ces bibliothèques dans l'environnement PATH est suffisante.

Review the queues and transaction tracking mode (Vérifier les files d'attente et le mode de suivi des transactions)

Sélectionnez cette case à cocher pour ignorer les étapes de demande et réponse complètement. Si le fichier de trafic brut est issu de CAI, cette option sera désélectionnée. CAI détecte automatiquement les files d'attente et les transactions. Si le fichier de trafic brut provient d'un autre emplacement, cette option sera sélectionnée par défaut et permet de vérifier les paramètres de destination et de suivi.

Correlation

Contient les schémas de corrélation potentiels pour mettre en corrélation les demandes et les réponses de transactions. JMS est asynchrone, c'est-à-dire que les demandes et les réponses sont reçues séparément. Cette liste déroulante vous permet de définir dans l'enregistreur de VSE la demande à associer à une réponse. Le champ Correlation comprend les options suivantes :

- Sequential (Séquentiel) : toutes les réponses sont associées à la dernière demande reçue, dans l'ordre chronologique. Il n'y a aucun schéma de corrélation et l'enregistreur MQ de VSE place un verrou exclusif en lecture sur la file d'attente de réponses dynamiques pour qu'aucun autre écouteur MQ ne puisse utiliser ses messages de réponse.

Lorsque vous spécifiez un schéma de corrélation, via les champs Correlation ID (ID de corrélation) ou Message ID to Correlation ID (De l'ID de message à l'ID de corrélation), l'enregistreur MQ de VSE suppose que les autres écouteurs présents sur la file d'attente de réponses dynamiques utilisent également un schéma de corrélation. Si tous les écouteurs présents sur la file d'attente de réponses dynamiques utilisent des schémas de corrélation, l'enregistreur MQ de VSE peut maintenir ses réponses à part, sans devoir recourir à un verrou exclusif en lecture et il place l'indicateur d'entrée partagé en tête de la file d'attente.

- Correlation ID (ID de corrélation) : la demande et la réponse doivent avoir le même ID de corrélation.
- Message ID to Correlation ID (De l'ID de message à l'ID de corrélation) : l'ID de message de la demande doit être identique à l'ID de corrélation de la réponse.
- Message ID (ID du message) : la demande et la réponse doivent avoir le même ID de message.

6. Cliquez sur Next (Suivant).

L'onglet Destination Info (Informations sur la destination) s'ouvre.

7. Entrez les noms du proxy et de files d'attente dynamiques et sélectionnez le type de file d'attente.

La case à cocher Create Proxy Queue (Créer une file d'attente de proxy) permet de créer une file d'attente temporaire à la volée pour l'utiliser comme file d'attente de proxy. Sélectionnez cette option si vous n'avez pas créé manuellement la file d'attente de proxy sur WebSphere MQ.

8. Cliquez sur l'onglet Connection setUp (Configuration de la connexion).
9. Saisissez les paramètres de connexion utilisés pour la connexion au MOM.

Ces paramètres de connexion sont enregistrés en interne.


L'onglet Advanced (Avancé) au bas de l'onglet Connection setUp (Configuration de la connexion) contient les sous-onglets suivants :

Environnement

Ce paramètre permet d'ajouter, de supprimer et de spécifier des valeurs pour d'autres paramètres d'environnement MQ.


MQ Exits (Sorties MQ)

Ce paramètre permet de pointer vers des sorties MQ pour la sécurité, l'envoi et la réception.

10. Pour définir un ensemble distinct d'informations de connexion facultatif, à utiliser lorsque vos files d'attente de proxy résident dans un gestionnaire de files d'attente différent de celui de vos files d'attente dynamiques, cliquez sur l'onglet Connection setUp (Configuration de la connexion).
11. Définissez les détails de connexion pour les destinations de réponse à écouter pour récupérer des messages.
12. Définissez la file d'attente de proxy dans laquelle l'application cliente recevra les réponses.
 - Rappelez-vous qu'il est possible d'obtenir plusieurs réponses pour une demande. Pour enregistrer un ensemble de files d'attente de proxy de réponse, cliquez sur  Add (Ajouter).
 - Avant d'enregistrer la file d'attente de réponse que vous voulez utiliser pour les demandes inconnues, sélectionnez la case à cocher Use for Unknown Responses (Utiliser pour les réponses inconnues).
 - Pour définir une destination temporaire, sélectionnez la case à cocher Use temporary queue/topic (Utiliser une file d'attente ou une rubrique temporaire). La sélection de cette case à cocher désactive les champs de nom de destination et de nom de destination de proxy et marque l'écouteur de réponse que vous ajoutez comme réponse temporaire. Le nom de la destination est vide.

Une destination temporaire dans la messagerie est une destination créée à la demande pour un client de messagerie. Une destination temporaire est généralement utilisée dans des scénarios demande/réponse. DevTest prend en charge l'utilisation d'une file d'attente temporaire pour les réponses, mais prend uniquement en charge une transaction simultanée avec une file d'attente temporaire à la fois.
13. Cliquez sur Next (Suivant).

L'onglet Destination List (Liste de destinations) s'ouvre.
14. Cliquez sur l'onglet Current Connection Info (Informations sur la connexion en cours) et vérifiez que les informations de connexion sont correctes.

15. Les informations affichées dans l'onglet Current Connection Info (Informations sur la connexion en cours) sont copiées à partir des informations de connexion spécifiées précédemment. Vous pouvez les modifier dans le rare cas où les informations de connexion de réponse sont différentes.
16. Pour démarrer l'enregistrement, cliquez sur Next (Suivant).
 Les noms des files d'attente écoutées par le VSE s'affichent et le statut est défini sur Waiting (En attente).
 Si vous vous connectez au WebSphere MQ et vous avez décidé de créer les files d'attente de proxy à la volée, ces files d'attente de proxy seront créées.
17. Exécutez le client qui ajoute des messages à la file d'attente de proxy de demandes.
 VSE copie ces messages dans la file d'attente des demandes réelle. Le serveur récupère ces messages à partir de la file d'attente et envoie des réponses à la ou les files d'attente des réponses. Le VSE récupère ces messages et les copie dans la file d'attente de proxy des réponses, écoutée par le client.
 Lors de l'enregistrement des transactions, le nombre de messages augmente, tout comme le nombre total de sessions et de transactions dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel). A l'issue de l'enregistrement, toutes les demandes sont passées par la même file d'attente des demandes. Environ la moitié des réponses sont revenues via des files d'attente temporaires et l'autre moitié via la file d'attente des réponses non temporaire.
 A l'issue de l'exécution, le nombre de messages dans les files d'attente des réponses peut être inférieur de 1 à la valeur attendue. En effet, une demande unique peut avoir plusieurs réponses et VSE n'aura pas encore reconnu les dernières transactions terminées. Les messages correspondant à la dernière transaction ne sont par conséquent pas comptés.
18. Cliquez sur Next (Suivant) pour déclencher la clôture de la dernière transaction et effectuez le nettoyage nécessaire. Une fenêtre intermédiaire apparaîtra si vous avez utilisé un protocole de données dynamique. Dans le cadre de la préparation de l'enregistreur pour l'écriture du fichier .vsi, il vérifie que les corps des demandes et des réponses sont du texte, si elles sont marquées comme tel. Sinon, le type devient binaire.
 Un protocole de données de gestionnaire de données de demande a été ajouté côté demande. Pour plus d'informations sur la configuration de ce protocole, reportez-vous à la section [Gestionnaire de données de demande](#) (page 269). Vous pouvez modifier ces protocoles de données ou en ajouter d'autres.
19. Cliquez sur Next (Suivant).
Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).
20. Pour fermer la fenêtre et stocker l'image, cliquez sur Finish (Terminer).
21. Examinez et enregistrez le modèle de service virtuel généré dans la fenêtre principale.

JMS

Les rubriques suivantes contiennent des instructions détaillées sur l'enregistrement d'une image de service qui utilise le protocole de transport JMS :

- [Enregistrement de proxy de base](#) (page 155)
- [Enregistreur TIBCO Monitor](#) (page 160)

Conditions requises : L'utilisation de DevTest avec cette application requiert que la mise à disposition d'un ou plusieurs fichiers pour DevTest. Pour plus d'informations, consultez la section Conditions requises pour les fichiers des applications tierces de la rubrique *Administration*.

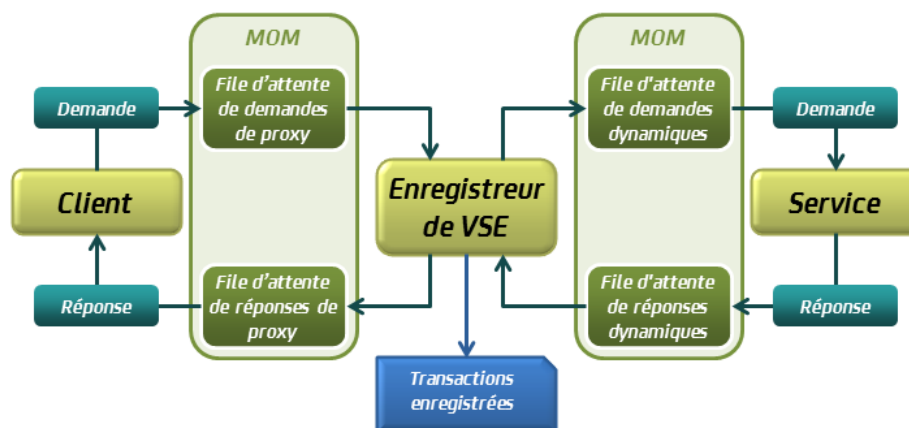
Enregistrement de proxy de base

L'enregistrement de proxy est une méthode commune pour l'enregistrement des applications de messagerie.

Le graphique suivant illustre les principaux composants de l'enregistrement de proxy :

- Client
- Service
- Files d'attente de proxy
- Files d'attente dynamiques
- Enregistreur de VSE

Les intergiciels asynchrones (Message-Oriented Middleware, MOM) sont une plate-forme sur laquelle les messages sont échangés.



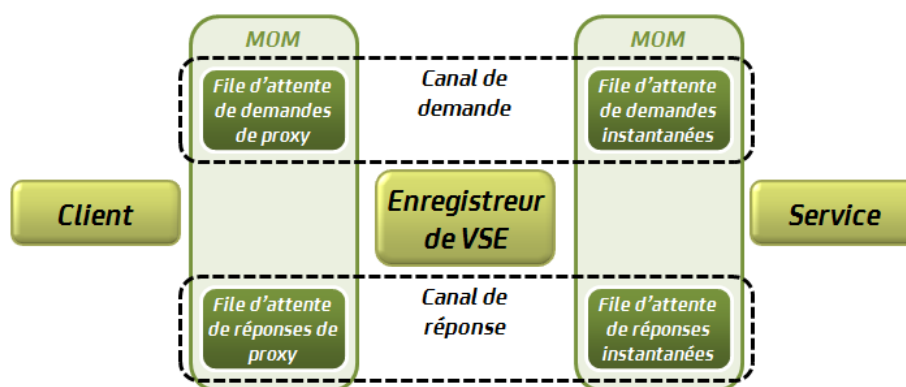
L'enregistreur de VSE est inséré dans le flux de message entre le client et le service. Chaque message de demande passe par l'enregistreur avant d'atteindre le service et chaque message de réponse passe par l'enregistreur avant d'atteindre le client.

Un *canal* permet d'associer une file d'attente de proxy et une file d'attente dynamique.

La file d'attente de demande de proxy et la file d'attente de demandes dynamiques forment un canal de demande.

La file d'attente de réponse de proxy et la file d'attente de réponses dynamiques forment un canal de réponse.

Le graphique suivant présente le canal de demande et le canal de réponse dans un enregistrement de proxy.



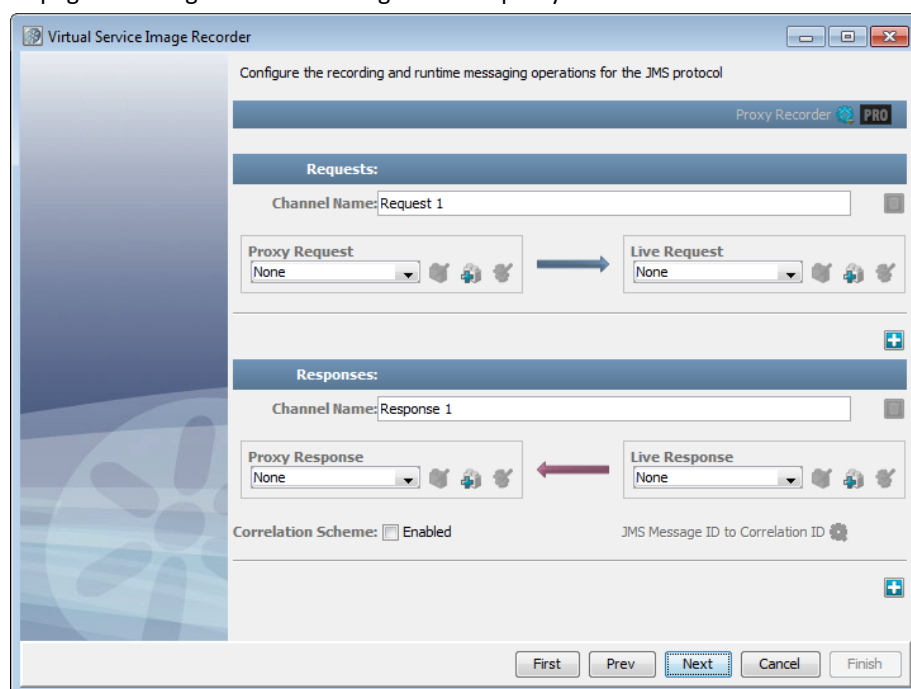
Un enregistrement de proxy peut avoir plusieurs canaux de demande et plusieurs canaux de réponse.

Dans cette procédure, vous sélectionnez un actif pour chaque file d'attente. Un *actif* est un ensemble de propriétés de configuration groupées dans une unité logique. Pour plus d'informations sur les actifs, reportez-vous à la rubrique *Utilisation de CA Application Test*.

Procédez comme suit:

1. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, définissez le protocole de transport sur JMS. Veillez à configurer l'image de service et le modèle de service virtuel.
2. Cliquez sur Next (Suivant).

La page de configuration de l'enregistreur de proxy s'ouvre.



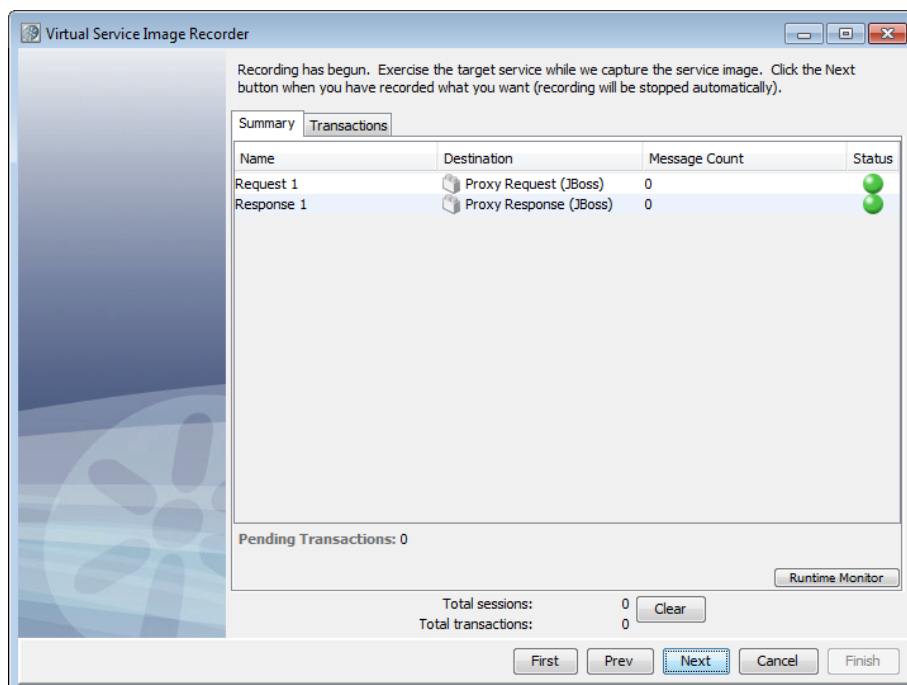
Cette page contient les paramètres de base et les paramètres avancés. Pour afficher les paramètres avancés, cliquez sur PRO en haut de l'éditeur.

Chaque canal doit avoir un nom unique. Le nom de canal permet au modèle de service virtuel et à l'image de service de communiquer sur les files d'attente utilisées dans la transaction.

3. Sélectionnez les actifs de file d'attente pour les listes Proxy request (Demande de proxy), Live request (Demande dynamique), Proxy Response (Réponse de proxy) et Live Response (Réponse dynamique). Si les actifs n'ont pas été créés, vous pouvez les créer à partir de cette page. Vous pouvez également modifier des actifs à partir de cette page.
4. (Facultatif) Cliquez sur les icônes plus (+) pour ajouter des canaux de demande et de réponse.

5. (Facultatif) Cochez la case Correlation Scheme (Schéma de corrélation) et spécifiez le schéma de votre choix. Chaque canal de réponse peut avoir un schéma de corrélation différent. Pour une description des schémas de corrélation, reportez-vous à la documentation de l'étape JMS Send Receive (Envoyer/recevoir des messages JMS) dans la rubrique *Utilisation de CA Application Test*.
6. Cliquez sur Next (Suivant) pour lancer la session d'enregistrement.

La page de commentaires de l'enregistrement s'ouvre.



Le tableau dans l'onglet Summary (Récapitulatif) répertorie chaque canal de demande et de réponse. Chaque ligne inclut le nom de canal, l'actif de destination sélectionné, le nombre de messages qui ont transité dans le canal et le statut.

Vous pouvez cliquer sur une cellule dans la colonne Destination pour afficher la destination de proxy associée au canal.

Au bas de la page, vous pouvez consulter le nombre de transactions en attente, le nombre de sessions et le nombre total de transactions.

Si une erreur empêche le démarrage de la session d'enregistrement, une boîte de dialogue s'ouvre et vous retournez à la page de configuration de l'enregistreur de proxy.

Des erreurs peuvent également se produire pendant la session d'enregistrement. Les erreurs associées à un canal de demande ou de réponse spécifique s'affichent dans la colonne Status (Statut). L'icône de statut devient rouge et une info-bulle affiche le message d'erreur. Les erreurs plus générales s'affichent en haut de la page de commentaires d'enregistrement.

Vous pouvez utiliser le moniteur d'exécution pour afficher les actifs utilisés en temps réel. Pour plus d'informations sur le fonctionnement du moniteur d'exécution, reportez-vous à la documentation de l'étape JMS Send Receive (Envoyer/recevoir des messages JMS) dans la rubrique *Utilisation de CA Application Test*.

7. Exécutez le service cible.
8. Lorsque le service est terminé, cliquez sur Next (Suivant) pour terminer l'enregistrement.
9. Spécifiez les protocoles de données et cliquez sur Next (Suivant).
10. (Facultatif) Enregistrez un fichier de session d'enregistrement.
11. Cliquez sur Finish (Terminer).

Vous avez créé l'[image de service](#) (page 328) et le [modèle de service virtuel](#) (page 365).

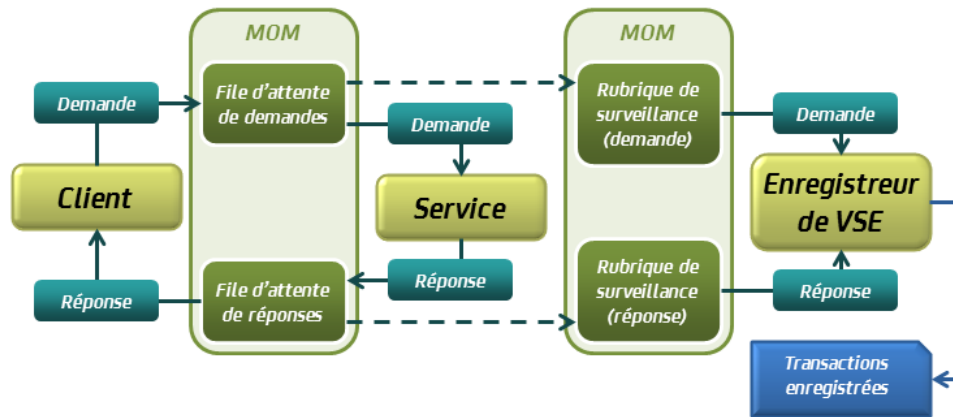
Enregistreur TIBCO Monitor

Pour chaque file d'attente existante sur un serveur TIBCO EMS, un utilisateur disposant de droits d'administrateur peut se connecter à une rubrique de moniteur spéciale. Si un administrateur est abonné à cette rubrique de moniteur, il peut recevoir une copie de tous les messages envoyés sur la file d'attente correspondante. Cette fonctionnalité permet à l'enregistreur de VSE de se situer complètement à l'extérieur du flux de message de l'application et de surveiller simplement le trafic de messages.

Le graphique suivant présente les composants principaux de l'enregistrement TIBCO Monitor :

- Client
- Service
- Files d'attente de demandes et de réponses
- Rubriques de surveillance de demande et de réponse
- Enregistreur de VSE

Les intergiciels asynchrones (Message-Oriented Middleware, MOM) sont une plate-forme sur laquelle les messages sont échangés.



Monitor Topic (Request) (Rubrique de surveillance (demande))

Rubrique de surveillance TIBCO associée à la file d'attente de demandes pour l'application. Le serveur TIBCO renvoie automatiquement chaque message envoyé dans la file d'attente de demandes vers cette rubrique.

Monitor Topic (Response) (Rubrique de surveillance (réponse))

Rubrique de surveillance TIBCO associée à la file d'attente de réponses pour l'application. Le serveur TIBCO renvoie automatiquement chaque message envoyé dans la file d'attente de réponses vers cette rubrique.

L'enregistreur TIBCO Monitor prend en charge TIBCO EMS 6.3.

Vous ne pouvez pas utiliser de rubriques de surveillance pendant la lecture pour virtualiser le service. Vous pouvez effectuer l'une des opérations suivantes :

- Créez un ensemble distinct de files d'attente et configurez le client et le service de VSE pour les utiliser.
- Arrêtez le service dynamique pour que le service de VSE puisse le remplacer.

Dans l'actif de connexion JMS, l'administrateur doit avoir disposer des droits d'administration.

Procédez comme suit:


1. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, définissez le protocole de transport sur JMS. Veillez à configurer l'image de service et le modèle de service virtuel.
2. Cliquez sur Next (Suivant).
La page de configuration de l'enregistreur de proxy s'ouvre.
3. Remplacez le type d'enregistreur Proxy Recorder (Enregistreur de proxy) par TIBCO Monitor Recorder (Enregistreur TIBCO Monitor).
La page de configuration de l'enregistreur de proxy contient des paramètres de base et des paramètres avancés. Pour afficher les paramètres avancés, cliquez sur PRO en haut de l'éditeur.
4. Sélectionnez les actifs de file d'attente pour la liste Receive Destination (Destination de réception) et la liste Send Destination (Destination d'envoi). Si les actifs n'ont pas été créés, vous pouvez les créer à partir de cette page. Vous pouvez également modifier des actifs à partir de cette page.
5. (Facultatif) Cliquez sur les icônes plus (+) pour ajouter des files d'attente de demandes et de réponses.
6. (Facultatif) Cochez la case Correlation Scheme (Schéma de corrélation) et spécifiez le schéma à utiliser.
7. Cliquez sur Next (Suivant) pour lancer la session d'enregistrement.
La page de commentaires de l'enregistrement s'ouvre.
Le tableau dans l'onglet Summary (Récapitulatif) contient une liste de chaque canal de demande et de réponse. Chaque ligne inclut le nom de canal, l'actif de destination sélectionné, le nombre de messages qui ont transité dans le canal et le statut.
8. Exécutez le service cible.
9. Lorsque le service est terminé, cliquez sur Next (Suivant) pour terminer l'enregistrement.
10. Spécifiez les protocoles de données et cliquez sur Next (Suivant).
11. (Facultatif) Enregistrez un fichier de session d'enregistrement.
12. Cliquez sur Finish (Terminer).
Vous avez créé l'image de service et le modèle de service virtuel.

Standard JMS

Cette rubrique contient des instructions détaillées sur l'enregistrement d'une image de service virtuel à l'aide du protocole de transport Standard JMS.

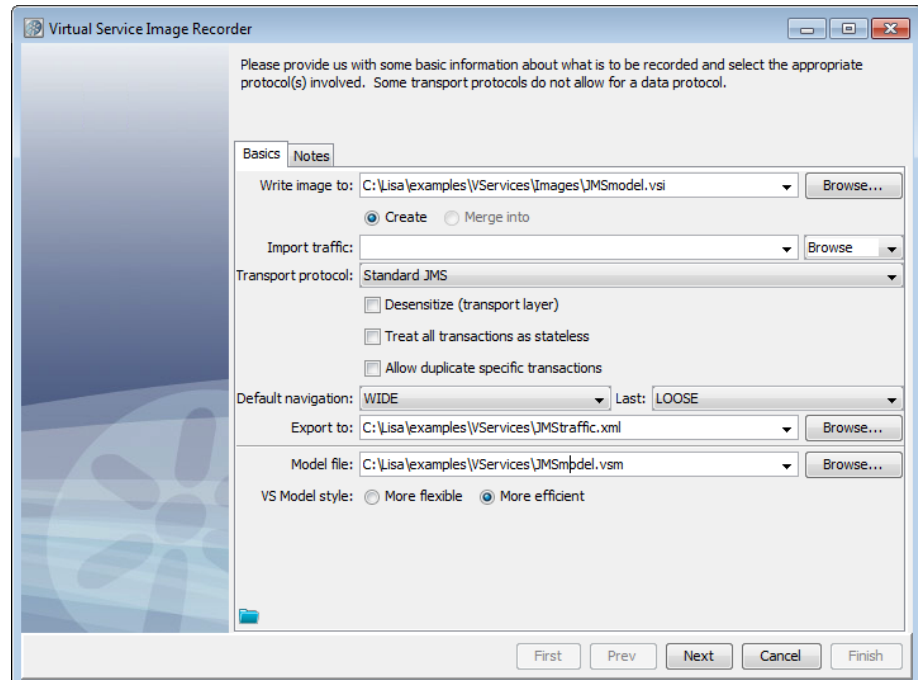
Conditions requises : L'utilisation de DevTest avec cette application requiert que la mise à disposition d'un ou plusieurs fichiers pour DevTest. Pour plus d'informations, consultez la section Conditions requises pour les fichiers des applications tierces de la rubrique *Administration*.

Procédez comme suit:

- Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :
 - Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
 - Cliquez avec le bouton droit de la souris sur le nœud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

- Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



- Cliquez sur Next (Suivant).

L'étape de sélection de mode d'enregistrement s'ouvre, avec une des options suivantes sélectionnées :

Proxy Mode (Mode Proxy)

Le mode Proxy est le seul mode d'enregistrement réellement pris en charge par VSE. Le mode Proxy fournit les options suivantes :

- Generate service using the Live queues (Générer le service à l'aide de files d'attente dynamiques)
- Generate service using the Proxy queues (Générer le service à l'aide de files d'attente de proxy)

Le mode Proxy permet de configurer et d'utiliser des files d'attente de proxy dans le bus de messagerie. L'application cliente est configurée de sorte à envoyer les messages vers ces destinations de proxy, puis DevTest les enverra vers la destination réelle une fois enregistrés. Le même comportement s'applique côté réponse. DevTest est configuré de sorte à écouter sur la destination de réponse réelle, récupérer le message, puis le transmettre à la destination de proxy de réponse. Ce mode peut se comporter de manière différente si vous avez activé les destinations temporaires et que les réponses sont automatisées.

Import Mode (Mode Importation)

Ce mode d'enregistrement des services virtuels est disponible si vous spécifiez un fichier de trafic brut dans le champ Import traffic (Importer un fichier de trafic) dans la fenêtre d'enregistrement initiale. Dans ce mode d'enregistrement, vous pouvez configurer des informations supplémentaires sur les files d'attente de demande et de réponse détectées dans le fichier de trafic brut. Vous pouvez également ignorer toutes les étapes de réponse et de demande.

4. Pour enregistrer en mode Proxy, remplissez les champs suivants :

Generate service using the Live queues / Proxy queues (Générer le service à l'aide de files d'attente dynamiques/de proxy)

Indique si les files d'attente dynamiques ou de proxy doivent être utilisées lors de la génération du dernier modèle de service virtuel et de l'image.

Max Pending Transactions (Nombre maximum de transactions en attente)

Indique le nombre maximum d'écouteurs de réponse exécutés dans chaque file d'attente de réponses. Ces écouteurs de réponse sont exécutés lorsqu'une transaction est en attente. Lorsque le nombre de transactions en attente dépasse le nombre maximum, la plus ancienne transaction est automatiquement fermée. Si vous saisissez 0 dans ce champ, il n'existera pas de nombre maximum.

Disable Multiple Responses (Désactiver les réponses multiples)

Indique si plusieurs réponses doivent être prises en charge dans chaque transaction. Par défaut, une transaction reste en attente après la première réponse et attend que d'autres réponses s'y ajoutent. Si cette option est sélectionnée, la transaction est automatiquement fermée après la première réponse. Si de nombreuses transactions entrent rapidement, cette option peut renforcer les performances, notamment pour MQ.

5. Pour enregistrer en mode Import (Importer), remplissez les champs suivants :

Review the queues and transaction tracking mode (Vérifier les files d'attente et le mode de suivi des transactions)

Sélectionnez cette case à cocher pour ignorer les étapes de demande et réponse complètement. Si le fichier de trafic brut est issu de CAI, cette option sera désélectionnée. CAI détecte automatiquement les files d'attente et les transactions. Si le fichier de trafic brut provient d'un autre emplacement, cette option sera sélectionnée par défaut et permet de vérifier les paramètres de destination et de suivi.

Correlation

Contient les schémas de corrélation potentiels pour mettre en corrélation les demandes et les réponses de transactions. JMS est asynchrone, c'est-à-dire que les demandes et les réponses sont reçues séparément. Cette liste déroulante vous permet de définir dans l'enregistreur de VSE la demande à associer à une réponse. Le champ Correlation comprend les options suivantes :

- Sequential (Séquentiel) : toutes les réponses sont associées à la dernière demande reçue, dans l'ordre chronologique. Il n'y a aucun schéma de corrélation et l'enregistreur MQ de VSE place un verrou exclusif en lecture sur la file d'attente de réponses dynamiques pour qu'aucun autre écouteur MQ ne puisse utiliser ses messages de réponse.

Lorsque vous spécifiez un schéma de corrélation, via les champs Correlation ID (ID de corrélation) ou Message ID to Correlation ID (De l'ID de message à l'ID de corrélation), l'enregistreur MQ de VSE suppose que les autres écouteurs présents sur la file d'attente de réponses dynamiques utilisent également un schéma de corrélation. Si tous les écouteurs présents sur la file d'attente de réponses dynamiques utilisent des schémas de corrélation, l'enregistreur MQ de VSE peut maintenir ses réponses à part, sans devoir recourir à un verrou exclusif en lecture et il place l'indicateur d'entrée partagé en tête de la file d'attente.

- Correlation ID (ID de corrélation) : la demande et la réponse doivent avoir le même ID de corrélation.
- Message ID to Correlation ID (De l'ID de message à l'ID de corrélation) : l'ID de message de la demande doit être identique à l'ID de corrélation de la réponse.
- Message ID (ID du message) : la demande et la réponse doivent avoir le même ID de message.

6. Cliquez sur Next (Suivant).

L'onglet Destination Info (Informations sur la destination) s'ouvre.

7. Entrez les noms de proxy et de destinations dynamiques, puis sélectionnez le type de destination.
8. Cliquez sur l'onglet Connection setUp (Configuration de la connexion).
9. Saisissez les paramètres de connexion utilisés pour la connexion au MOM.


Ces paramètres de connexion sont enregistrés en interne.

L'onglet Advanced (Avancé) au bas de l'onglet Connection setUp (Configuration de la connexion) permet de définir des propriétés de connexion personnalisées pour l'image de service.

10. Cliquez sur Next (Suivant).

L'onglet Destination List (Liste de destinations) s'ouvre.

11. Définissez les détails de connexion pour les destinations de réponse à écouter pour récupérer des messages.
12. Définissez la file d'attente de proxy dans laquelle l'application cliente recevra les réponses.

- Rappelez-vous qu'il est possible d'obtenir plusieurs réponses pour une demande. Pour enregistrer un ensemble de files d'attente de proxy de réponse, cliquez sur  Add (Ajouter).
- Avant d'enregistrer la file d'attente de réponse que vous voulez utiliser pour les demandes inconnues, sélectionnez la case à cocher Use for Unknown Responses (Utiliser pour les réponses inconnues).
- Pour définir une destination temporaire, sélectionnez la case à cocher Use temporary queue/topic (Utiliser une file d'attente ou une rubrique temporaire). La sélection de cette case à cocher désactive les champs de nom de destination et de nom de destination de proxy et marque l'écouteur de réponse que vous ajoutez comme réponse temporaire. Le nom de la destination est vide.

Une destination temporaire dans la messagerie est une destination créée à la demande pour un client de messagerie. Une destination temporaire est généralement utilisée dans des scénarios demande/réponse. DevTest prend en charge l'utilisation d'une file d'attente temporaire pour les réponses, mais prend uniquement en charge une transaction simultanée avec une file d'attente temporaire à la fois.

13. Cliquez sur l'onglet Current Connection Info (Informations sur la connexion en cours) et vérifiez que les informations de connexion sont correctes.

Les informations qui s'affichent dans cet onglet sont copiées à partir des informations de connexion fournies préalablement. Vous pouvez les modifier dans le rare cas où les informations de connexion de réponse sont différentes.

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

Les noms des destinations écoutées par le VSE s'affichent.

Le champ Pending Transactions (Transactions en attente) affiche le nombre de transactions stockées dans le tampon de transactions en attente, qui attend d'autres réponses. Les transactions se ferment lorsque le nombre maximum de transactions en attente est atteint ou à l'aide de l'option Disable Multiple Responses (Désactiver les réponses multiples). A mesure que les transactions se ferment, elles sont déplacées du tampon de transactions en attente vers le tampon de transactions totales.

15. Exécutez le client qui ajoute des messages à la file d'attente de proxy de demandes.

VSE copie ces messages dans la file d'attente des demandes réelle. Le serveur récupère ces messages à partir de la file d'attente et envoie des réponses à la ou les files d'attente de réponses. Le VSE récupère ces messages et les copie dans la file d'attente de proxy des réponses, écoutée par le client.


Lors de l'enregistrement des transactions, le nombre de messages augmente, tout comme le nombre total de sessions et de transactions dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel). A l'issue de l'enregistrement, toutes les demandes sont passées par la même file d'attente des demandes. Environ la moitié des réponses sont revenues via des files d'attente temporaires et l'autre moitié via la file d'attente des réponses non temporaire.

A l'issue de l'exécution, le nombre de messages dans les files d'attente des réponses peut être inférieur de 1 à la valeur attendue. En effet, une demande unique peut avoir plusieurs réponses et VSE n'aura pas encore reconnu les dernières transactions terminées. Les messages correspondant à la dernière transaction ne sont par conséquent pas comptés.

Les protocoles de données recommandés sont remplacés par les valeur par défaut dans ce panneau. Vous pouvez y ajouter ou modifier des protocoles de données côté demande et réponse. Si des panneaux de configuration sont disponibles pour les protocoles de données sélectionnés, ils s'ouvrent ensuite. Pour plus d'informations, reportez-vous à la section Utilisation des protocoles de données.

16. Cliquez sur Next (Suivant).

La dernière transaction est fermée et le nettoyage nécessaire est terminé.

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

17. Pour fermer la fenêtre et stocker l'image, cliquez sur Finish (Terminer).
18. Examinez et enregistrez le modèle de service virtuel généré dans la fenêtre principale.


Java

L'agent Java de DevTest est requis pour enregistrer des images de service Java. Dans l'application LISA Bank, l'agent est installé par défaut. Pour toute autre application, installez l'agent et lancez le registre.

A des fins de test, démarrez le serveur de démonstration sur lequel l'agent est installé, ou exécutez l'application LISA Bank. Pour l'exécuter avec une autre application Java, configurez l'agent. Pour plus d'informations sur l'agent, reportez-vous à la rubrique *Agents*.

Lorsque vous utilisez simultanément plusieurs enregistreurs dans la même station de travail pour enregistrer plusieurs appels distants EJB à partir de différentes classes, l'image de service et le modèle de service virtuel créés par chaque enregistreur affichent une liste cumulée de classes.

Procédez comme suit:

1. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :
 - Dans la barre d'outils principale, cliquez sur VSE Recorder (Enregistreur de VSE)
.
 - Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, By Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

2. Dans l'onglet [Basics \(Paramètres de base\)](#) (page 131), sélectionnez Java comme protocole de transport.
3. Cliquez sur Next (Suivant).

La fenêtre Select Java classes to virtualize (Sélectionner les classes Java à virtualiser) s'ouvre.
4. Pour déplacer des agents spécifiques entre les colonnes suivantes, utilisez les flèches disponibles :


Available Online Agents (Agents en ligne disponibles)

Répertorie les agents en ligne disponibles auquel vous pouvez vous connecter.


Connected Agents (Agents connectés)

Répertorie les agents en ligne ou hors ligne connectés pour le modèle de service virtuel. Les agents hors ligne apparaissent en police italique grise. Vous pouvez sélectionner les agents dans la liste Available Online Agents (Agents en lignes disponibles).

Lorsque vous sélectionnez un agent dans l'une des listes, le nom d'hôte et la classe principale s'affichent.

5. Pour ajouter un agent qui ne figure pas dans la liste, saisissez son nom dans le champ au-dessus de la liste des agents connectés et cliquez sur Add Agent (Ajouter un agent) .

Ce mécanisme est fourni pour ajouter des agents hors ligne qui n'étaient pas présents préalablement dans la liste d'agents connectés. Le nom d'agent ne peut pas être vide ni déjà présent dans la liste d'agents connectés. Si le nom d'agent entré est présent dans la liste d'agents en ligne, il sera déplacé de la liste des agents en ligne disponibles vers la liste des agents connectés.

6. Pour sélectionner une classe en effectuant une recherche, procédez comme suit :
 - a. Double-cliquez sur Search for Classes (Rechercher des classes).
 - b. Saisissez une chaîne de recherche comme nom complet (y compris le package), à l'aide d'expressions régulières.
 - c. Cliquez sur l'icône Search (Rechercher).
 - d. Sélectionnez une classe. Certaines classes peuvent apparaître plusieurs fois. Sélectionnez une classe une seule fois.
 - e. Cliquez sur la flèche droite.
7. Pour sélectionner une classe en saisissant manuellement son nom, procédez comme suit :
 - a. Double-cliquez sur Manually Enter a Class Name (Entrer manuellement un nom de classe).
 - b. Saisissez le nom complet de la classe.
 - c. Cliquez sur la flèche droite.
8. Pour sélectionner une classe dans une liste de classes proposées par l'agent, procédez comme suit :
 - a. Double-cliquez sur Agent Suggestions (Suggestions de l'agent).
 - b. Cliquez sur Retrieve (Récupérer) .
 - c. Sélectionnez une classe.
 - d. Cliquez sur la flèche droite.
9. Pour ajouter un protocole à l'enregistrement, double-cliquez sur Protocols et sélectionnez le protocole de votre choix.
10. Arrêtez le système testé.
11. Cliquez sur Next (Suivant).
La page Recording Has Begun (L'enregistrement a commencé.) s'ouvre.
12. Redémarrez le système testé.

Remarque : L'arrêt et le redémarrage du système testé sont facultatifs. Le redémarrage permet à VSE de capturer le trafic initial qui se produit lors de l'initialisation du système testé.


Une fois que le système testé est démarré, une liste des transactions les plus récentes s'affiche. Pour afficher le contenu de la transaction sélectionnée, double-cliquez dessus.

13. A l'issue de l'enregistrement, cliquez sur Next (Suivant).

Le panneau Data Protocols (Protocoles de données) s'ouvre et contient les protocoles de données pertinents préremplis. Vous pouvez modifier ces protocoles de données ou en ajouter d'autres. Il peut y avoir des panneaux supplémentaires pour configurer les protocoles de données. Pour plus d'informations sur la configuration de protocoles de données, reportez-vous à la section Utilisation des protocoles de données.

14. Une fois les protocoles de données de texte délimité configurés, cliquez sur Next (Suivant).

Lors de la préparation de l'enregistrement du fichier .vsi, l'enregistreur vérifie que les corps des demandes et des réponses sont du texte, s'ils sont marqués comme tel. Sinon, le type devient binaire.

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

15. Cliquez sur Finish (Terminer).

JDBC


Cette rubrique contient des instructions détaillées sur l'enregistrement d'une image de service virtuel à l'aide du protocole de transport JDBC (basé sur le pilote).

Remarque : Le protocole de transport JDBC (basé sur un pilote) ne sera plus pris en charge dans une future version. Cette fonctionnalité est remplacée par la virtualisation JDBC basée sur l'agent.

Pour plus d'informations sur les conditions préalables et la procédure de préparation, reportez-vous à la section [Virtualisation du protocole JDBC](#) (page 175).

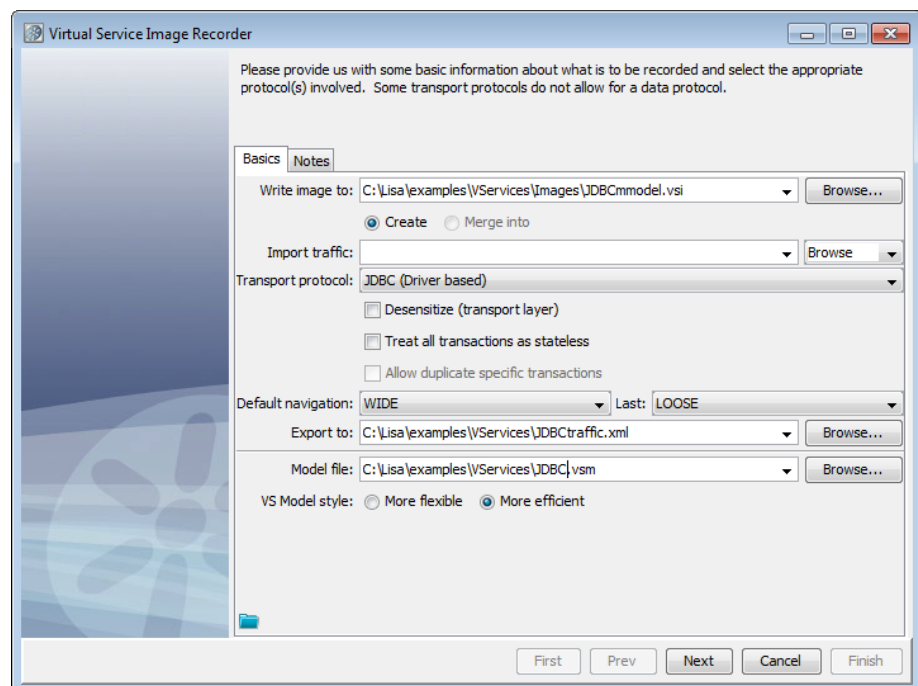
Procédez comme suit:

1. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :

- Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
- Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

2. Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



3. Cliquez sur Next (Suivant).

La fenêtre de configuration de terminal s'ouvre.

4. Permet de configurer l'hôte de simulation et la plage de ports (valeur par défaut : 2999) de manière adéquate.

Le VSE pour JDBC prend en charge plusieurs terminaux pendant l'enregistrement et la lecture. L'enregistreur et l'éditeur d'étapes d'écoute via JDBC contiennent une table dans laquelle figurent le Driver Host (Hôte de pilote), le Base Port (Port de base) et le Max Port (Port maximum). Si le port de base et le port maximum sont différents, un terminal unique sera créé le port de base, le port maximum et chaque port entre les deux.

5. Pour vous connecter au serveur de simulation JDBC, cliquez sur Next (Suivant).

Pour arrêter les tentatives de connexion, cliquez sur Cancel (Annuler) dans la fenêtre de statut.

6. Sélectionnez les URL de connexion et les utilisateurs à enregistrer dans l'onglet SQL Activity (Activité SQL).

L'onglet SQL Activity indique les dix plus récentes instructions exécutées (et le nombre d'exécution) pour chaque combinaison unique URL de connexion/utilisateur de la base de données. Cet affichage s'actualise environ toutes les cinq secondes.

- Pour placer la chaîne de connexion sélectionnée et l'utilisateur dans les champs URL et User (Utilisateur) situés au bas du panneau, cliquez sur To URL (URL de destination).
- Pour placer l'URL dans la liste URL to Record (URL à enregistrer), cliquez sur Add (Ajouter).

7. Cliquez sur l'onglet Loaded Drivers (Pilotes chargés).

L'onglet Loaded Drivers répertorie tous les pilotes JDBC installés et enregistrés dans le système testé.

- Sélectionnez un pilote et cliquez sur To URL (URL de destination) pour placer une expression régulière correspondant à une connexion via ce pilote dans le champ URL situé au bas du panneau.
- Cliquez sur Add (Ajouter) pour ajouter le modèle à la liste URLs to Record (URL à enregistrer).

URLs to Record (URL à enregistrer)

Les URL ajoutées à partir de la liste SQL Activity (Activité SQL) ou des champs URL/User entry (Saisie d'URL/d'utilisateur) sont affichées. Pour supprimer une entrée, sélectionnez cette option et cliquez sur Remove (Supprimer).

User (Utilisateur)

Zone sans nom sous le champ URLs to Record (URL à enregistrer), dans laquelle vous pouvez inclure un utilisateur de la base de données avec l'URL. Si vous n'incluez aucun utilisateur de la base de données et laissez cette zone vide, l'enregistrement sera effectué pour tous les utilisateurs connectés à l'URL.

Remarque : Pour activer le bouton Add (Ajouter), indiquez une URL de connexion dans le premier champ et un nom d'utilisateur dans le champ suivant.

La section inférieure contient la liste des combinaisons URL de connexion/utilisateur de base de données enregistrées. L'URL peut être une expression régulière. En général, cette liste est au départ vide, sauf si l'attribut d'état est défini sur RECORD (Enregistrement) dans une URL de connexion de simulation pour les systèmes testés de style DataSource.

Si un pilote est sélectionné dans la liste de pilotes, le bouton To URL (URL de destination) peut être utilisé pour copier une expression régulière générique qui correspond aux URL de connexion pour ce pilote dans le champ de saisie URL. Si le noeud de l'URL de connexion (niveau supérieur) dans l'arborescence d'activité est sélectionné, le bouton To URL (URL de destination) peut être utilisé pour copier l'URL et l'utilisateur dans les champs de saisie URL et User (Utilisateur). Aussi, pour ajouter l'URL de connexion et l'utilisateur directement à la liste d'enregistrements, utilisez le bouton Add (Ajouter) à droite de l'arborescence d'activité.

Les URL de connexion peuvent être exactes ou une expression régulière qui correspond à des demandes de connexion et peuvent être ajoutées à la liste d'enregistrements avec ou sans utilisateur de base de données. L'absence d'utilisateur de base de données correspond à la tentative de connexion d'un utilisateur quelconque. Si le bouton Add (Ajouter) à droite des champs de saisie URL/User (Utilisateur) est désactivé, il est probable que la liste d'enregistrements couvre déjà la combinaison URL/utilisateur.

8. Lorsque la liste d'enregistrements contient toutes les URL et les utilisateurs à enregistrer, cliquez sur Next (Suivant).

La fenêtre Recording Has Begun (L'enregistrement a commencé.) s'ouvre.

Les options et les statistiques d'affichage dynamique incluent les éléments suivants :

Conversation count (Nombre de conversations)

Indique le nombre de conversations enregistrées.

Total transactions (Nombre total de transactions)

Indique le nombre de transactions enregistrées.

Clear (Effacer)

Pour effacer des transactions actuellement enregistrées, cliquez sur ce bouton.

Le nombre de conversations totales et de transactions totales augmente en fonction du nombre de transactions enregistrées.

Remarque : Si aucune transaction n'a été enregistrée, un conflit de ports peut survenir. Le client envoie des transactions à l'application, non à l'enregistreur d'image de service virtuel. Si un autre service utilise ce port, arrêtez-le ou modifiez le paramètre de port afin d'éviter tout conflit.

Si vous rencontrez des problèmes de performance, il est possible que la base de données cible réponde lentement. Dans le fichier **user.home\lisatmp\tmanager.log**, recherchez des messages de débogage indiquant le temps d'exécution de la requête.

Par exemple :

```
2009-07-01 15:35:39,248 [AWT-EventQueue-0] DEBUG
com.itko.lisa.vse.stateful.model.SqlTimer - Exec time 72ms :
SELECT TRAFFIC_ID, LAST_MODIFIED, SERVICE_INFO, CREATED_ON,
NOTES, UNK_CONV_RESPONSE, UNK_STLS_RESPONSE FROM SVSE_TRAFFIC
```


Si la durée de la requête dépasse un seuil, des messages d'avertissement semblables au message suivant seront générés :

```
2009-07-01 15:17:11,161 [AWT-EventQueue-0] WARN
com.itko.lisa.vse.stateful.model.SqlTimer - SQL query took a long
time (112 ms) : SELECT TRAFFIC_ID, LAST_MODIFIED, SERVICE_INFO,
CREATED_ON, NOTES, UNK_CONV_RESPONSE, UNK_STLS_RESPONSE FROM
SVSE_TRAFFIC
```

9. A l'issue de l'enregistrement, cliquez sur Next (Suivant).

L'enregistreur prépare l'écriture du fichier .vsi en vérifiant les corps des demandes et des réponses. L'enregistreur vérifie qu'il s'agit de texte, si elles sont marquées comme tel. Sinon, le type devient binaire.

L'enregistreur exécute le post-traitement de l'enregistrement.

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

10. Cliquez sur Finish (Terminer) pour stocker l'image.
11. Vérifiez et enregistrez le modèle de service virtuel dans DevTest Workstation.

Virtualisation du protocole JDBC

Cette section fournit plus d'informations sur la préparation avant l'utilisation de VSE pour virtualiser le trafic de base de données JDBC. Des informations sur la configuration JDBC sont également disponibles dans la section [Installation du simulateur de base de données](#) (page 23). Cette section décrit diverses méthodes permettant de combiner la virtualisation JDBC basée sur le pilote et les options de configuration prises en charge. Cette section n'inclut aucune information sur la procédure d'implémentation de la configuration dans un conteneur d'application spécifique.

Remarque : La virtualisation de JDBC fonctionne uniquement avec Java 1.6 (non avec Java 1.5).

Propriétés de System-Level

Vous pouvez spécifier les propriétés suivantes de l'une des manières suivantes :

- A l'aide de variables système (en ajoutant l'indicateur **-Dvar=val** au démarrage du système testé, ou via un autre mécanisme)
- A l'aide d'un fichier de propriétés nommé **rules.properties**, qui doit être enregistré dans la variable classpath (probablement dans le même répertoire que **lisajdbcsim.jar**).

Si une propriété existe dans les deux emplacements, la propriété système remplace le fichier de configuration.

lisa.jdbc.sim.require.remote

Détermine si le pilote doit attendre une connexion au VSE avant de traiter des commandes.

Valeur par défaut : false

lisa.jdbc.hijack.drivermanager

Détermine si le pilote doit tenter de prendre en charge toutes les connexions à la base de données à partir de cette application.

Valeur par défaut : false

lisa.jdbc.sim.port

Définit le port d'écoute par défaut.

Valeur par défaut : 2999

lisa.log.level

Spécifie le niveau de journalisation par défaut.

Valeurs :

- OFF

- FATAL
- ERREUR
- AV.
- INFO
- DEBUG
- DEV

Par défaut : WARN

`lisa.log.target`

Spécifie l'emplacement d'écriture des journaux.

Valeurs :

- stderr
- stdout
- Spécifie un emplacement de fichier.

Valeur par défaut : stderr

Pilote du VSE (autonome)

Le système testé peut utiliser le pilote de VSE directement ou vous pouvez encapsuler le pilote de VSE dans une source de données, telle que Apache BasicDataSource. Dans ces cas, la configuration (autre que le nom d'utilisateur et le mot de passe) est transmise via l'URL. Les propriétés de pilote sont les suivantes :

URL (Propriété/URL)

Spécifie l'URL transférée au pilote sous-jacent. Elle doit être le dernier élément transmis.

State (Etat)

Spécifie l'état initial du pilote.

Valeurs :

- watch
- enregistrement
- playback (lecture)

JdbcSimPort

Désigne le port d'écoute des connexions provenant de VSE. Cette valeur remplace la propriété de niveau système.

Pilote

Spécifie le nom de la classe du pilote réel à utiliser.

User (Utilisateur)**Password (Mot de passe)**

Spécifiez **com.itko.lisa.vse.jdbc.driver.Driver** comme nom de la classe du pilote.

L'exemple suivant présente le format de l'URL. Tout est facultatif, sauf le pilote et l'URL.

```
jdbc:lisasim:driver=<real driver class>;state=<initial  
state>;jdbcSimPort=<starting port>;url=<real url>
```

Remarque : L'élément URL doit être placé à la fin. Tous les éléments qui suivent url= seront transmis tels quels au pilote sous-jacent. DevTest prend en charge le transfert d'informations supplémentaires au pilote sous-jacent en les intégrant à l'URL réelle.

Par exemple, l'URL suivante convient à une connexion Derby :

```
jdbc:lisasim:driver=org.apache.derby.jdbc.ClientDriver;state=watch  
;jdbcSimPort=4000;url=jdbc:derby://localhost:1527/sample;create=true
```

Cette URL indique au pilote de VSE de :

- Utiliser le pilote client Derby
- Démarrer en mode watch
- Utiliser le port 4000 pour communiquer avec le VSE
- Transférer l'URL jdbc:derby://localhost:1527/sample;create=true au vrai pilote Derby

Encapsulage d'un pilote à l'aide d'une source de données de VSE

L'encapsulage d'un pilote à l'aide d'une source de données de VSE peut être une configuration plus habituelle pour des conteneurs d'application, comparée à l'utilisation directe d'un pilote de VSE. Le VSE fournit une source de données permettant d'encapsuler un pilote réel. La plupart des conteneurs d'application fournissent un mécanisme permettant de spécifier directement des propriétés pour la source de données.

Spécifiez **com.itko.lisa.vse.jdbc.driver.VSEDataSource** comme nom de la classe du pilote.

Propriétés de la source de données**Pilote**

Nom du pilote réel à utiliser (configuration normale)

URL (Propriété/URL)

Spécifie l'URL réelle de connexion (ne doit pas commencer par **jdbc:lisasim**).

User (Utilisateur)

Password (Mot de passe)

JdbcSimPort

Spécifie le port d'écoute des connexions provenant de VSE. Cette valeur remplace la propriété de niveau système.

CustomProperties

Liste de paires nom=valeurs séparées par des points-virgules. Ces propriétés sont analysées et déléguées à la source de données encapsulée, le cas échéant. Si le premier caractère n'est pas alphanumérique, il sera utilisé comme délimiteur au lieu du point-virgule.

Encapsulage d'une source de données à l'aide d'une source de données de VSE

L'encapsulage d'une source de données est une configuration inhabituelle, lorsqu'un conteneur d'applications ne permet pas d'instancier un pilote directement avec `Class.forName()`. Pour l'utiliser, au lieu de spécifier **driver=<nom de classe du pilote réel>** (par exemple, **oracle.jdbc.OracleDriver**), spécifiez **datasource=<nom de classe de la source de données réelle>** (par exemple, **oracle.jdbc.pool.OracleDataSource**).

Indiquez **com.itko.lisa.vse.jdbc.driver.VSEDataSource** comme nom de la classe du pilote.

Prise en charge de terminaux multiples

Si une application utilise plusieurs connexions JDBC que vous voulez virtualiser (ou si un serveur d'applications contient plusieurs applications avec des connexions JDBC différentes que vous voulez virtualiser), spécifiez un port **JdbcSimPort** différent pour chaque connexion. Vous pouvez spécifier ce terminal comme propriété de la source de données du VSE, ou comme paramètre dans l'URL du pilote.

Exemple :

Supposez qu'une application utilise deux connexions JDBC et que vous voulez virtualiser les deux. Supposez que l'application utilise directement des pilotes.

Spécifiez **com.itko.lisa.vse.jdbc.driver.Driver** pour chaque connexion, avec les paramètres sous-jacents appropriés tels que le pilote et l'URL pour chacune d'elles. Pour la première connexion, vous pouvez spécifier **jdbcSimPort=3000** et pour la seconde vous pouvez utiliser **jdbcSimPort=3001**. Votre configuration peut se présenter comme suit :

```
connection1.url=jdbc:lisasim:driver=org.apache.derby.jdbc.ClientDriver;jdbcSimPort=3000;url=jdbc:derby://localhost:1527/sample;create=true
```

```
connection2.url=jdbc:lisasim:driver=org.apache.derby.jdbc.ClientDriver;jdbcSimPort=3001;url=jdbc:derby://localhost:1527/sample2;create=true
```


Bien que vous deviez encore effectuer des enregistrements distincts, vous pouvez déployer les deux services simultanément et les démarrer et les arrêter indépendamment.

TCP

Cette rubrique contient des instructions détaillées sur l'enregistrement d'une image de service virtuel à l'aide du protocole de transport TCP.

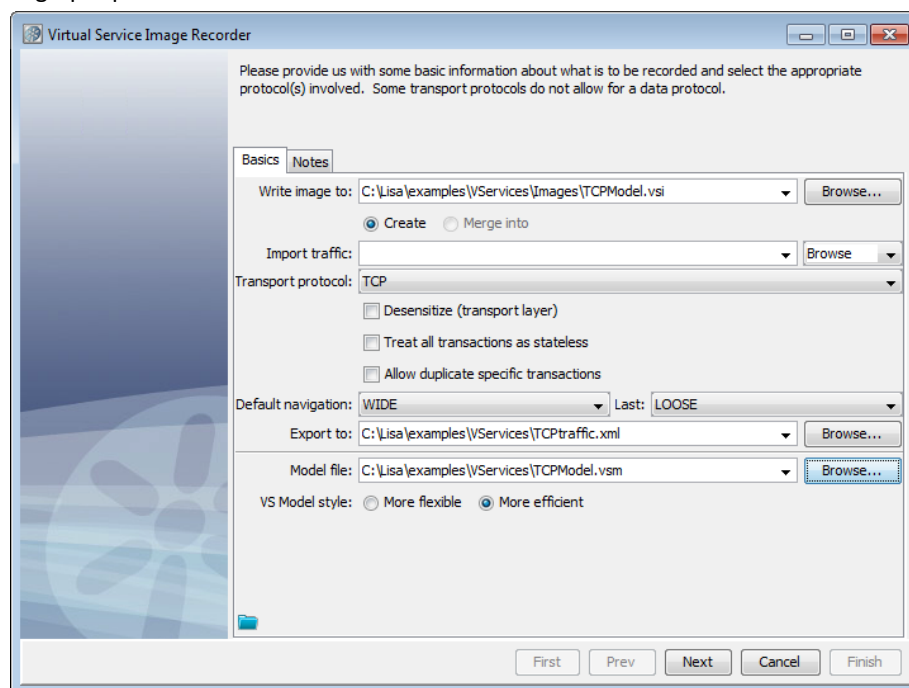
Pour plus d'informations sur les conditions préalables et la procédure de préparation, reportez-vous à la section [Virtualisation du protocole TCP](#) (page 183).

Procédez comme suit:

1. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :
 - Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
 - Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

2. Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



3. Cliquez sur Next (Suivant).
4. Saisissez les informations sur le client et le serveur, sélectionnez vos délimiteurs et ajoutez des paramètres de connexion SSL.

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Définit le numéro de port de l'hôte cible sur lequel le serveur est exécuté.

Treat request as text (Traiter la demande comme du texte)

Spécifie si la demande doit être traitée comme du texte. Pour plus d'informations, consultez la section [Préparation de la virtualisation du protocole TCP](#) (page 183).

Request Encoding (Codage de demande)

Répertorie les codages de demande disponibles sur l'ordinateur sur lequel la station de travail DevTest Workstation est exécutée. Il s'agit par défaut de la UTF8.

Treat response as text (Traiter la réponse comme du texte)

Spécifie si la réponse doit être traitée comme du texte. Pour plus d'informations, consultez la section [Préparation de la virtualisation du protocole TCP](#) (page 183).

Response Encoding (Codage de réponse)

Répertorie les codages de réponse disponibles sur l'ordinateur sur lequel la station de travail DevTest Workstation est exécutée. Il s'agit par défaut de la UTF8.

Use SSL to server (Utiliser une connexion SSL au serveur)

Permet de spécifier si DevTest utilise une connexion HTTPS pour envoyer la demande au serveur.

- Si cette option est **sélectionnée**, DevTest envoie une demande HTTPS (secured layer) au serveur.

Si vous sélectionnez Use SSL to server (Utiliser une connexion SSL au serveur) sans sélectionner Use SSL to client (Utiliser une connexion SSL au client), DevTest utilisera une connexion HTTP pour l'enregistrement. DevTest enverra alors ces demandes au serveur à l'aide d'une connexion HTTPS.

Si l'option est **désélectionnée**, DevTest enverra une demande HTTP au serveur.

Use SSL to client (Utiliser une connexion SSL au client)

Spécifie si un référentiel de clés personnalisé doit être utilisé pour lire une demande SSL à partir d'un client. Cette option est activée uniquement si l'option Use SSL to server (Utiliser une connexion SSL au serveur) a été sélectionnée.

Valeurs :

- **Option sélectionnée** : vous pouvez spécifier un référentiel de clés client personnalisé et une phrase secrète. Si ces paramètres sont saisis, ils sont utilisés au lieu des valeurs par défaut codées de manière irréversible.

Option désélectionnée : vous ne pouvez pas spécifier un référentiel de clés client personnalisé et une phrase secrète.

SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.


Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

5. Pour démarrer l'enregistrement, cliquez sur Next (Suivant).
6. A l'issue de l'enregistrement, cliquez sur Next (Suivant).
7. Si la réponse est chiffrée, compressée, ou codée, sélectionnez un protocole de données de réponse.
8. L'enregistreur tente de détecter des délimiteurs de message qui indiquent à DevTest qu'une demande ou une réponse complète a été lue. Confirmez ou corrigez ces délimiteurs dans cette fenêtre.

Remarque : Un délimiteur de demande est requis. Un délimiteur de réponse doit être sélectionné pour rendre l'appel dynamique disponible.

9. L'enregistreur vérifie que les corps des demandes et des réponses sont du texte, s'ils sont marqués comme tel. Sinon, le type devient binaire.

Remarque : Pour enregistrer les paramètres dans cet enregistrement à charger dans un autre enregistrement d'image de service, cliquez sur Save (Enregistrer)  au-dessus du bouton Finish (Terminer).

Le protocole TCP/IP prend en charge une étape d'appel dynamique. Pour activer l'étape d'appel dynamique, sélectionnez un protocole de réponse. Si vous ne sélectionnez aucun protocole de réponse, aucune étape d'appel dynamique ne sera incluse dans le modèle de service virtuel.

Virtualisation du protocole TCP

Le protocole de transport TCP permet de virtualiser le trafic TCP/IP brut entre un ou plusieurs clients et un serveur. Ce transport est similaire au protocole HTTP et utilise de nombreuses fonctionnalités identiques.

Lors de l'enregistrement, configurez le protocole TCP de la même manière que le protocole HTTP en mode Endpoint (Terminal). Attribuez-lui un socket d'écoute et ajoutez le serveur cible et la destination. Le protocole envoie des données entre le client et le serveur, puis écoute ces données pour créer des transactions de VSE.

Conversion d'octets en transactions du VSE

Il peut être difficile de connaître les éléments constituant une demande ou une réponse complète, ainsi que le mode de corrélation de ces demandes et réponses. Les données TCP n'ont aucune structure inhérente. Un délimiteur de demande est requis pour identifier les demandes dans le flux de données entrantes. Vous pouvez également sélectionner un délimiteur de réponse, mais cela n'est pas strictement requis.

L'enregistreur utilise les règles suivantes pour déterminer des demandes et des réponses et les corréler ensemble dans des transactions :

1. Une demande est suivie de 0 ou 1 réponse.
2. Chaque réponse est associée à la dernière demande complète.
3. Une demande est considérée complète lorsque des données de la réponse arrivent, même si le délimiteur de demande indique qu'elle n'est pas complète.
4. Une réponse est considérée complète lorsque des données de la demande arrivent, même si le délimiteur de réponse (le cas échéant) indique qu'elle ne l'est pas.
5. Si le délimiteur de réponse détecte une réponse complète lorsque des données de réponse sont reçues, toute donnée de réponse supplémentaire sera rejetée.

Par exemple :

Request 1 - Request 2 - Response A - Partial Request 3 - Response B
- Request 4 - Partial Response C - Request 5 - Response D With Trailing Data

Ces données sont analysées dans des transactions comme suit :

- Demande 1 sans réponse
- Demande 2 avec réponse A (règle 2)
- Demande partielle 3 avec réponse B (règle 3)
- Demande 4 avec réponse partielle C (règle 4)
- Demande 5 avec réponse D, mais avec les données de fin sont ignorées (règle 5)

Ce système fonctionne si le client et le serveur suivent un paradigme demande - réponse synchrone et qu'il gère les demandes sans réponse. Le système ne gère pas la communication asynchrone, ni une demande avec des réponses multiples.

Création de conversations

Par défaut, lors de l'enregistrement d'images de service TCP, le VSE crée des conversations basées sur le nom de votre ordinateur (ou l'adresse IP) et le socket de sortie. Lors de chaque changement de socket de sortie, une nouvelle conversation est lancée. Pour modifier ce comportement, cochez la case Treat all conversations as stateless (Traiter toutes les conversations comme conversations sans état).

Séparateurs prêts à l'emploi

DevTest contient les délimiteurs suivants :

Les enregistrements se terminent par des fins de ligne.

Les demandes et les réponses se terminent par un ou plusieurs caractères de nouvelle ligne. Le délimiteur même n'est pas inclus dans la demande/réponse.

Valeurs :

- \n
- \r
- Caractères Unicode 0085, 2028 et 2029

Records are of fixed length (La longueur des enregistrements est fixe.)

Les demandes et les réponses correspondent à une longueur en octets spécifique.

Records are delimited by specific characters (Les enregistrements sont délimités par des caractères spécifiques.)

Un caractère ou une séquence de caractères spécifique (par exemple, une virgule) termine les demandes et réponses. La séquence complète doit correspondre. Le délimiteur même n'est pas inclus dans la demande/réponse. Cette option est semblable à l'importation d'un fichier CSV.

Delimiters match a regular expression (Les délimiteurs correspondent à une expression régulière.)

Chaque demande/réponse se termine par un ensemble d'octets correspondant à l'expression régulière. Le délimiteur même n'est pas inclus dans la demande/réponse.

Si les caractères délimiteurs ne sont pas inclus dans la demande/réponse (comme dans la plupart des cas), les ensembles dos à dos de délimiteurs seront ignorés.

Remarque : Une fois l'enregistrement terminé, vous ne pouvez plus modifier le délimiteur sélectionné. Vous ne pouvez pas changer le délimiteur sélectionné dans l'éditeur de modèles.

Options Treat Request as Text (Traiter la demande comme du texte) et Treat Response as Text (Traiter la réponse comme du texte)

Les cases à cocher Treat Request as Text (Traiter la demande comme du texte) et Treat Response as Text (Traiter la réponse comme du texte) contrôlent le mode d'analyse des données de demande et de réponse.

Treat Request as Text (Traiter la demande comme du texte)

Spécifie si la demande doit être utilisée comme corps de la demande de VSE.

Valeurs :

- Si elle est **sélectionnée**, elle définit l'ensemble de la demande comme corps du texte de la demande du VSE. Le premier terme (avant le premier espace) est utilisé comme nom d'opération.
- **Option désélectionnée** : conserve le format binaire du corps de la demande et définit le nom d'opération sur OPERATION.

Treat Response as Text (Traiter la réponse comme du texte)

Spécifie si la réponse doit être utilisée comme corps de la réponse de VSE.

Valeurs :

- **Option sélectionnée** : le corps de la réponse est considéré comme du texte et tous les caractères nuls intégrés (\0) sont convertis en espaces.
- Si elle est **désélectionnée**, le corps de la réponse sera binaire.

Pourquoi le nombre de transactions est erroné ?

Le nombre de transactions est toujours décalé d'au moins 1 transaction. Lorsque le protocole TCP détecte une transaction complète, il la met en cache jusqu'au démarrage de la transaction suivante, car le serveur peut intentionnellement fermer le socket après l'envoi de la réponse. Si cela se produit, le protocole TCP le détecte et définit **lisa.vse.close_socket_after_response=TRUE** dans la réponse. Cette propriété indique au VSE de fermer le socket côté serveur après avoir envoyé cette réponse pendant la lecture.

Si vous n'utilisez aucun délimiteur de réponse, le nombre de transactions sera décalé d'une autre transaction. Le protocole TCP ignore que la réponse est terminée jusqu'au démarrage d'une nouvelle demande. L'erreur sur le nombre total peut donc être de 2 transactions. Le nombre sera correct à l'issue de l'enregistrement.

Enregistrement de CICS (LINK DTP MRO)

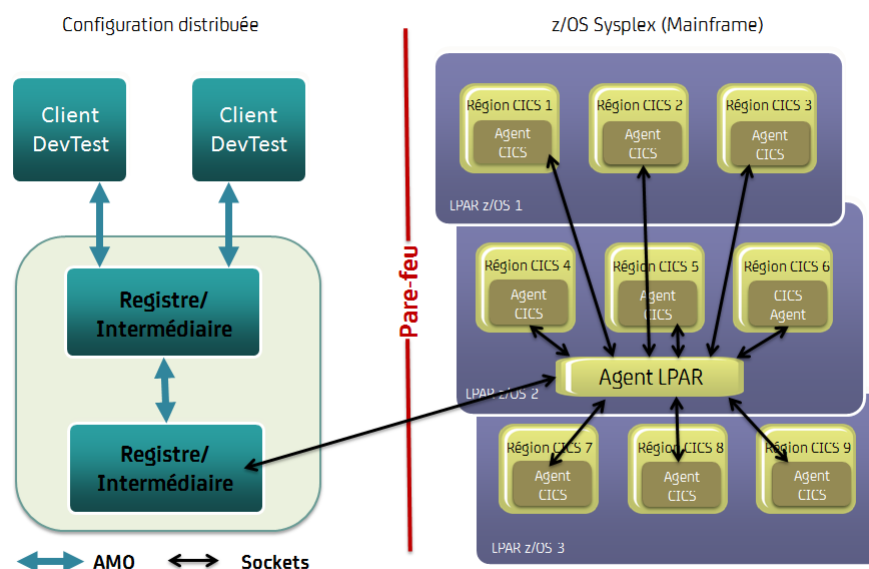
Cette rubrique présente des instructions détaillées sur l'enregistrement d'une image de service virtuel à l'aide du protocole de transport CICS. La section contient les rubriques suivantes :

- [Présentation de l'enregistrement de CICS d'IBM](#) (page 187)
- [Exemple d'enregistrement de CICS d'IBM](#) (page 189)
- [Utilisation du panneau CICS Programs to Virtualize \(Programmes CICS à virtualiser\)](#) (page 198)

Présentation de CICS d'IBM

Pour permettre l'enregistrement d'images de service CICS d'IBM, DevTest contient une passerelle Mainframe intégrée au registre. Cette passerelle Mainframe fournit un point de contact unique au mainframe pour tous les stations de travail DevTest Workstation. La passerelle Mainframe dispose d'un agent de partition logique exécuté sur une partition logique z/OS. L'agent de partition logique fournit un point de contact unique à tous les agents CICS.

Chaque région CICS a un agent CICS pour DevTest qui fournit la fonctionnalité de virtualisation. Les agents CICS communiquent avec l'agent de partition logique via des sockets et l'agent de partition logique communique avec la passerelle Mainframe via des sockets. La passerelle Mainframe fournit une interface ActiveMQ à tous les clients DevTest via le registre.



La passerelle Mainframe peut être exécutée dans les modes suivants :

Client mode (Mode Client)

La passerelle Mainframe initialise la connexion à l'agent de partition logique.

Server mode (Mode Serveur)

La passerelle Mainframe attend l'initialisation de la connexion par l'agent de partition logique.

Le fichier **lisa.properties** contient des **propriétés pour activer la passerelle mainframe**.
Pour plus d'informations, reportez-vous à la section Propriétés de la passerelle Mainframe.

Exemple de CICS d'IBM

Dans cet exemple, l'enregistreur du VSE enregistre une image de service CICS LINK.

Lorsque le registre est lancé, il indique que la passerelle Mainframe a démarré en mode client et celle-ci est connectée à l'agent de partition logique.

Cet exemple illustre la suppression d'une dépendance de ressource. Dans l'exemple, une application CICS requiert un fichier VSAM pour insérer des entrées dans des champs d'informations de solde de compte, mais ce fichier VSAM est indisponible. Cette démonstration illustre la virtualisation du programme qui gère le fichier VSAM manquant, en éliminant la dépendance de la ressource.

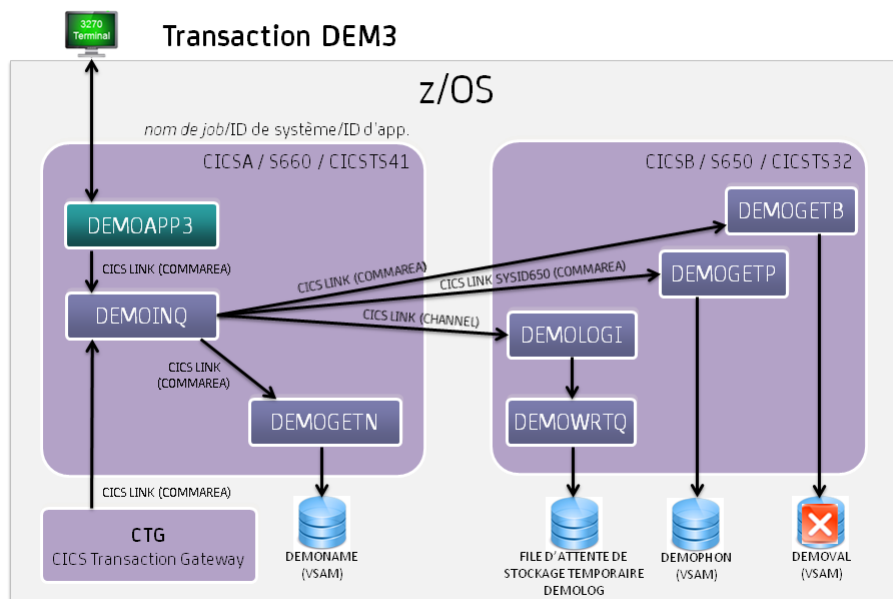
Le graphique suivant illustre la transaction CICS (DEM3) exécutée. La transaction est exécutée à partir d'un terminal 3270, mais elle peut également être exécutée à partir de la passerelle de transaction CICS.

L'exécution de la transaction initialise DEMOAPP3, qui établit un lien CICS LINK à DEMOINQ. DEMOINQ exécute alors les liens CICS LINK suivants :

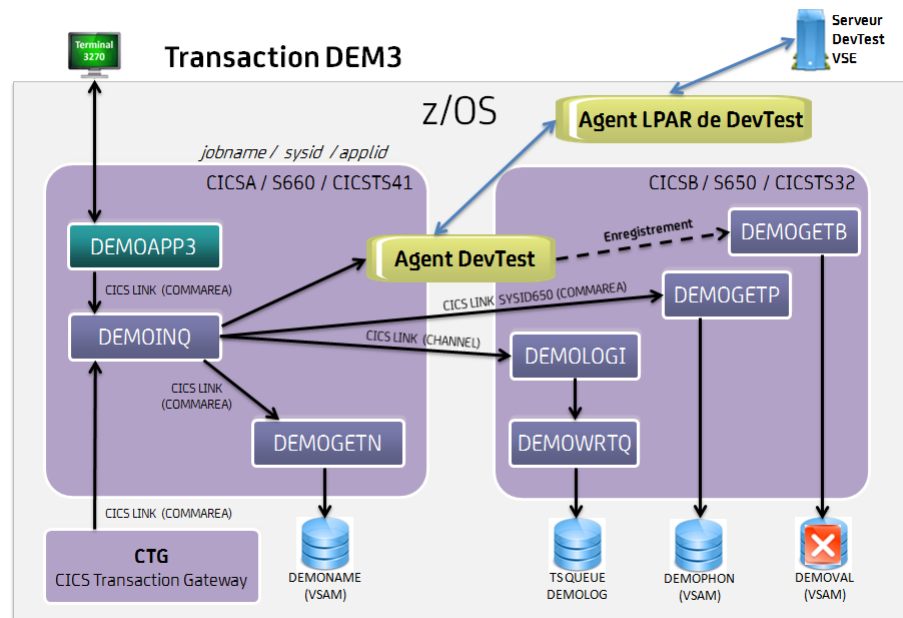
1. Un LINK local passe par une COMMAREA vers DEMOGETN pour obtenir le nom et l'adresse du client à partir d'un fichier VSAM.
2. Un lien CICS LINK vers CICSB passe par une COMMAREA vers DEMOGETB. DEMOGETB lit le DEMOGETBAL du fichier VSAM, qui n'est pas disponible dans l'exemple.
3. Un lien CICS LINK passe par une COMMAREA vers DEMOGETP pour obtenir les informations de numéro de téléphone.
4. Un lien CICS LINK passe par un CHANNEL avec des CONTAINERS vers DEMOLOGI.

Les régions CICS suivantes sont impliquées :

- Jobname CICS A, avec l'ID de système S660 et l'applid CICSTS41
- Jobname CICS B, avec l'ID de système S650 et l'applid CICSTS32



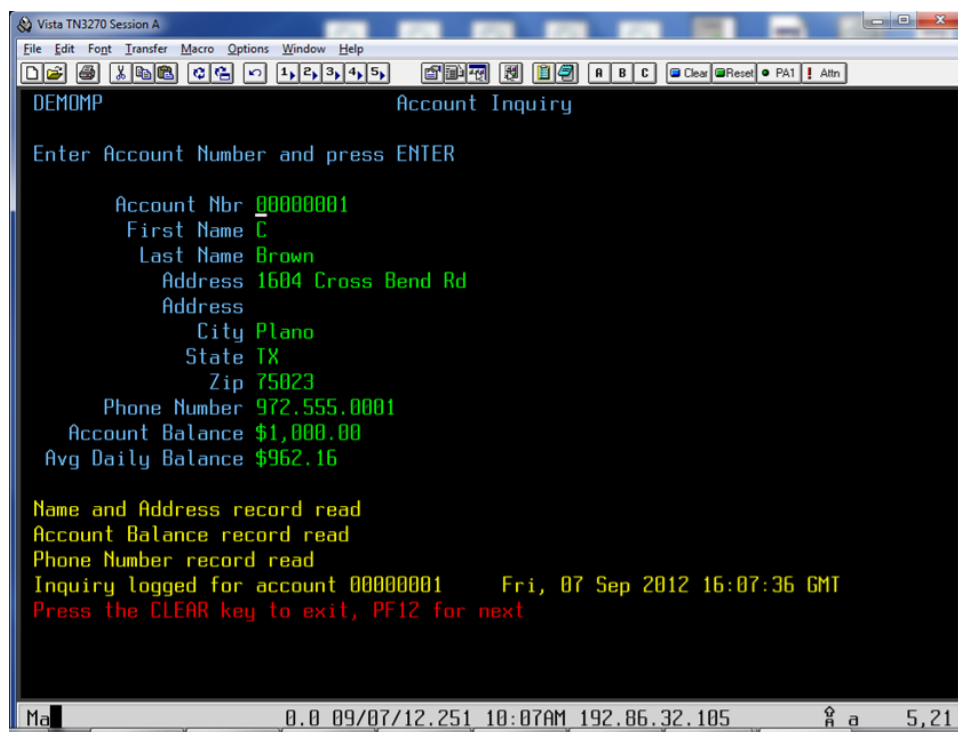
Cet exemple illustre la virtualisation du programme DEMOGETB sur CICSTS41. L'exemple utilise l'agent DevTest, exécuté dans CICSA, communiquant avec l'agent de partition logique, qui communique avec le serveur du VSE. La transaction CICS est d'abord exécutée pour enregistrer CICS LINK dans le programme DEMOGETB. Cet enregistrement sera alors utilisé pour virtualiser CICS LINK.



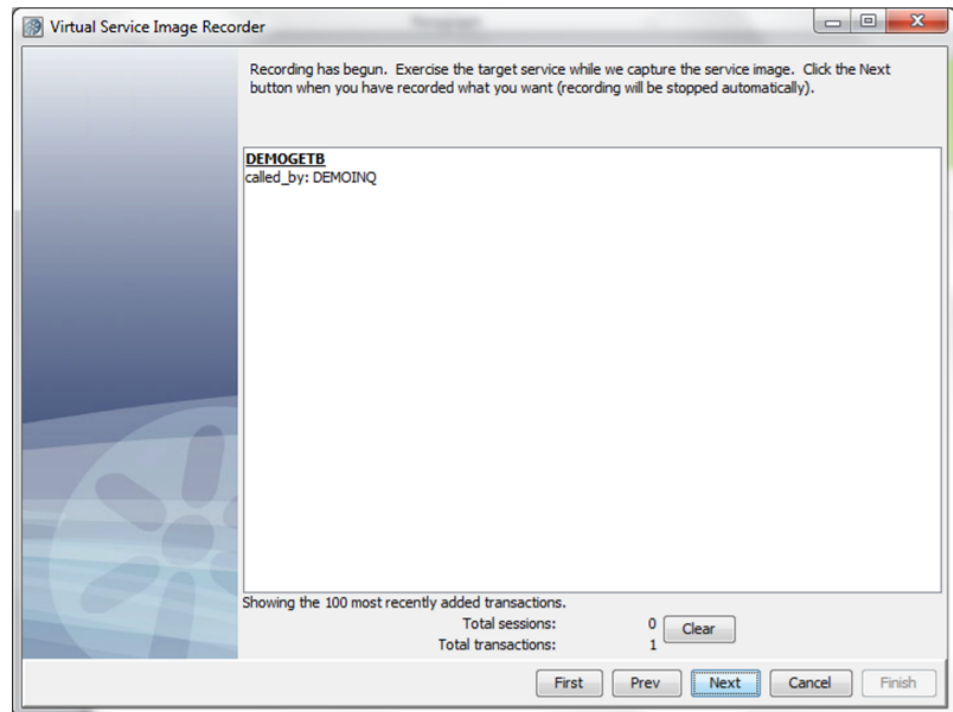
L'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel contient le nom de l'image de service (DEMOGETB.vsi), le modèle du service virtuel (DEMOGETB.vsm) et le protocole de transport de CICS LINK.

Le panneau CICS Programs to Virtualize (Programmes CICS à virtualiser) apparaît ensuite. Pour des informations détaillées sur ce panneau, reportez-vous à la section [Utilisation du panneau CICS Programs to Virtualize \(Programmes CICS à virtualiser\)](#) (page 198).

Pour lancer l'enregistrement, cliquez sur bouton Next (Suivant). Lors de l'enregistrement, la transaction DEM3 est exécutée pour appeler le programme DEMOGETB avec un lien CICS LINK.



Comme l'illustre le graphique suivant, après la première exécution, le VSE enregistre une transaction.



Dans le panneau suivant, VSE ajoute le protocole de données Copybook CICS côtés demande et réponse.

Le graphique suivant illustre le fichier XML de mappage de copybook à utiliser dans cet exemple. Ce fichier définit le mappage pour DEMOGETB et indique d'utiliser le fichier DEMOGETB.txt comme copybook si une demande de programme DEMOGETB existe.

payload-copybook-mapping-CICS.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<payloads>
  <!-- DEMOGETB -->
  <payload name="DEMOGETB" type="request" matchType="metaData"
    key="DEMOGETB" value="DEMOGETB" >
    <section name="Body">
      <copybook key="">DEMOGETB.txt</copybook>
    </section>
  </payload>
  <payload name="DEMOGETB" type="response" key="DEMOGETB" value="DEMOGETB">
    <section name="Body">
      <copybook key="">DEMOGETB.txt</copybook>
    </section>
  </payload>
</payloads>
```

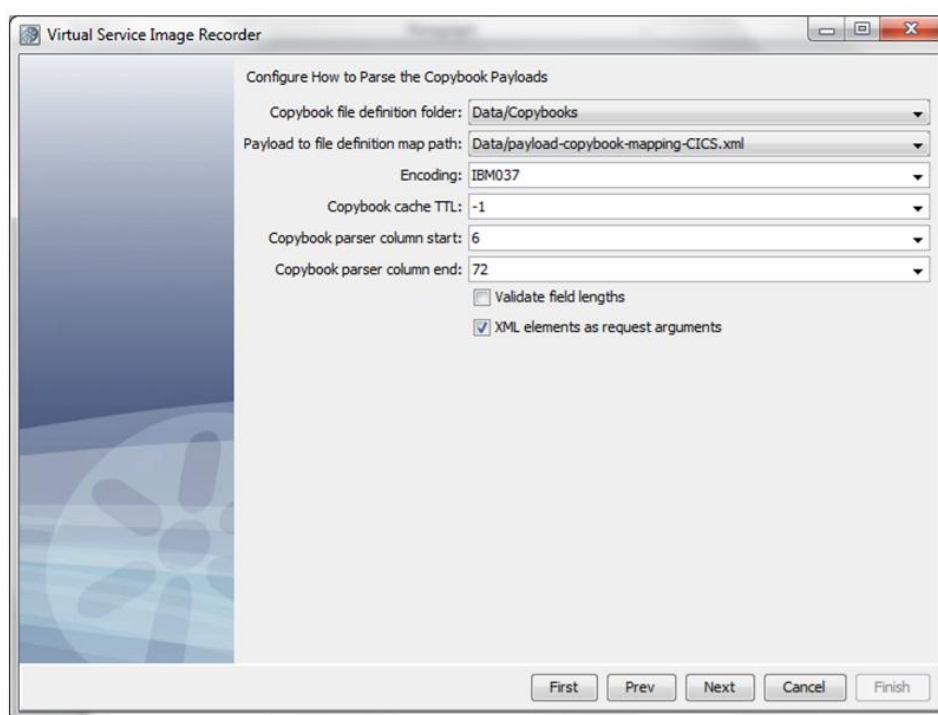
Comme l'illustre l'exemple suivant, le fichier DEMOGETB.txt contient la définition de copybook COBOL :

Copybooks/DEMOGETB.TXT

05 DEMOGETB-COMMAREA.

10	DEMOGETB-RETURN	PIC X.
10	DEMOGETB-MESSAGE	PIC X(70).
10	BALANCE-REC.	
15	BALANCE-ACCOUNT-NBR	PIC X(8).
15	FILLER	PIC X.
15	BALANCE-BALANCE	PIC X(14).
15	FILLER	PIC X.
15	BALANCE-AVERAGE	PIC X(14).

Le panneau suivant indique le dossier contenant DEMOGETB.txt et le fichier XML de mappage. La page de code est remplacée par IBM037.



Cliquez sur Next (Suivant). Le copybook traite les données.

Cliquez sur Next (Suivant) pour arrêter le traitement et effectuer l'enregistrement de l'image de service.

Lorsque l'image de service virtuel est ouverte dans l'éditeur d'images de service, vous pouvez observer que le copybook a mappé les données de la demande, puis les a converties en attributs. Par exemple :

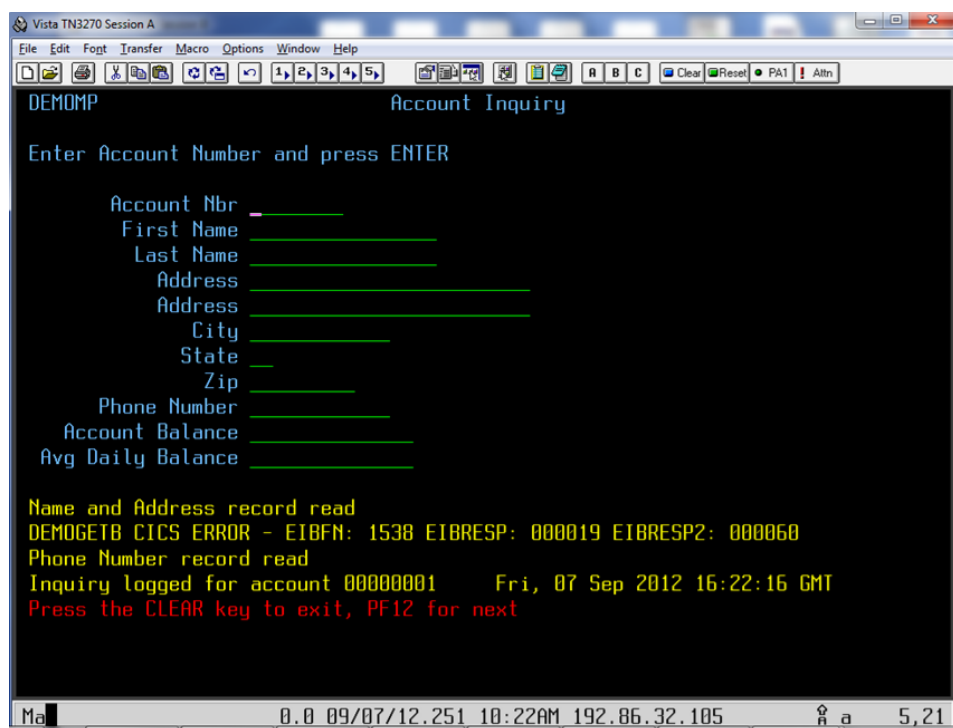
▼ Request Data					
Arguments Attributes Meta Data					
Name	Name in Session	Comparison Operator	Value	Magic String	Date Pattern
called_by		=	DEMOINQ	<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_DEMOGETB-RETURN		=		<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_DEMOGETB-MESSAGE		=		<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_BALANCE-REC_BALANCE-ACCOUNT-NBR		=	00000001	<input checked="" type="checkbox"/>	
Body_DEMOGETB-COMMAREA_BALANCE-REC_FILLER_1		=		<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_BALANCE-REC_BALANCE-BALANCE		=		<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_BALANCE-REC_FILLER_2		=		<input type="checkbox"/>	
Body_DEMOGETB-COMMAREA_BALANCE-REC_BALANCE-AVERAGE		=		<input type="checkbox"/>	

Le copybook a également formaté la réponse. Le graphique suivant illustre la réponse à renvoyer lors de l'utilisation de la virtualisation :

▼ Response 1 of 1					
Body Meta Data					
XML Document					
Node	Type	Occurs	Nil	Nilable	Value
copybook-payload			<input type="checkbox"/>		
Body			<input type="checkbox"/>		
DEMOGETB-COMMAREA			<input type="checkbox"/>		
DEMOGETB-RETURN			<input type="checkbox"/>	Y	
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	1	
DEMOGETB-MESSAGE			<input type="checkbox"/>	Account Balance record read	
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	70	
BALANCE-REC			<input type="checkbox"/>		
BALANCE-ACCOUNT-NBR			<input type="checkbox"/>	{{=request_Body_DEMOGETB_COMMAREA_BALANCE_REC_BALANCE_ACCOUNT_NBR;/"00000001"/}}	
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	8	
FILLER			<input type="checkbox"/>		
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	1	
BALANCE-BALANCE			<input type="checkbox"/>	\$1,000.00	
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	14	
FILLER			<input type="checkbox"/>		
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	1	
BALANCE-AVERAGE			<input type="checkbox"/>	\$962.16	
left-pad-length			<input type="checkbox"/>	0	
origin-length			<input type="checkbox"/>	14	

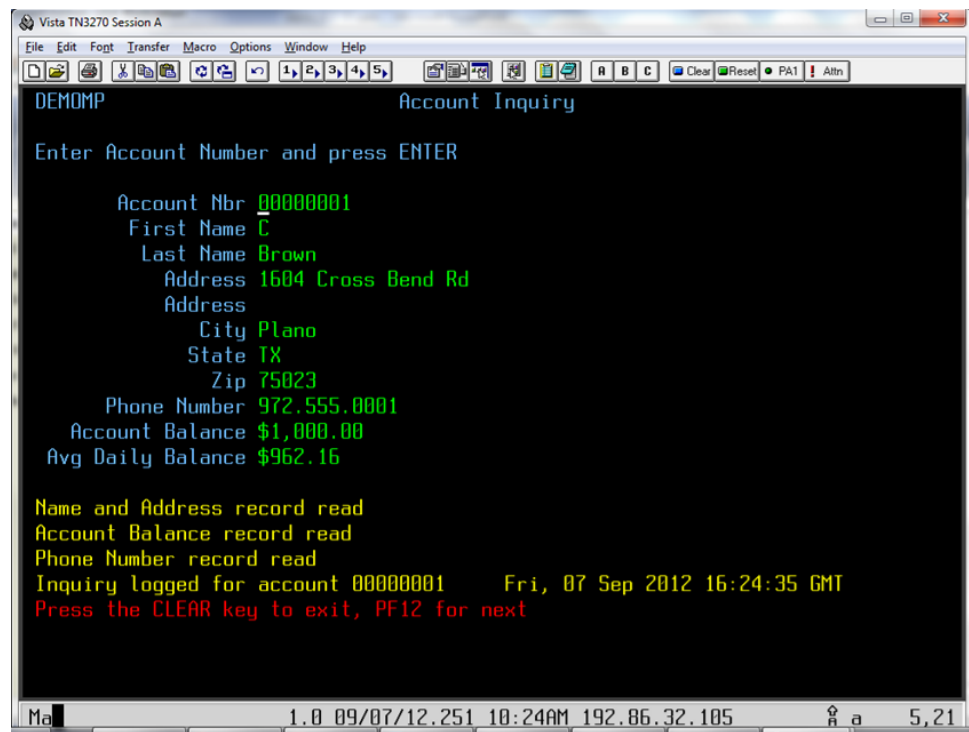
Exemple

Dans cet exemple, le DEMOBAL du fichier VSAM devient indisponible et le programme DEMO3 est exécuté. DEMO3 échoue en raison de l'indisponibilité de DEMOBAL.



Ensuite, le service virtuel est déployé.

Lorsque le programme DEMO3 est exécuté à nouveau, il fonctionne, même si le fichier VSAM est encore indisponible.

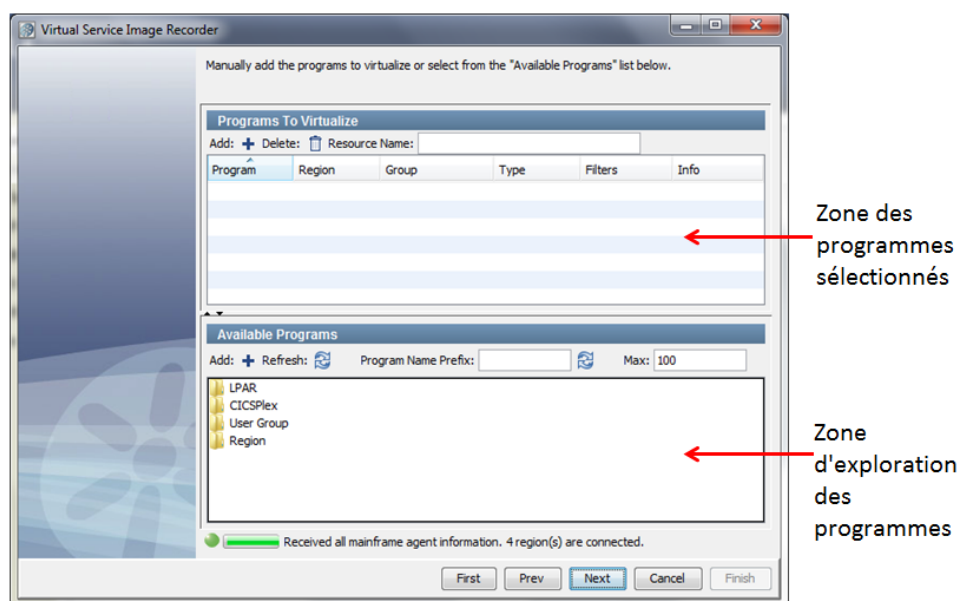


Panneau CICS Programs to Virtualize (Programmes CICS à virtualiser)

Utilisez le panneau CICS Programs to Virtualize (Programmes CICS à virtualiser) pour effectuer les opérations suivantes :

- Explorer les régions CICS connectées à des agents de partition logique pour DevTest
- Explorer les programmes définis pour ces régions CICS
- Sélectionner les programmes à virtualiser
- Ajouter manuellement les programmes à virtualiser
- Définir les filtres pour affiner les instructions CICS LINK, les transactions et les utilisateurs CICS virtualisés
- Etablir une appartenance de groupe pour virtualiser des programmes CICS dans un groupe CICS

Le graphique suivant présente les zones principales du panneau CICS Programs to Virtualize (Programmes CICS à virtualiser) :



Available Programs (Programmes disponibles)

La zone Available Programs (Programmes disponibles) permet d'explorer des régions CICS connectées à DevTest avec l'agent CICS pour DevTest et de consulter les programmes définis.

Vous pouvez explorer les régions CICS et programmes connus par :

- Partition logique (partition logique z/OS)

- CICSplex
- Groupement défini par l'utilisateur
- Région

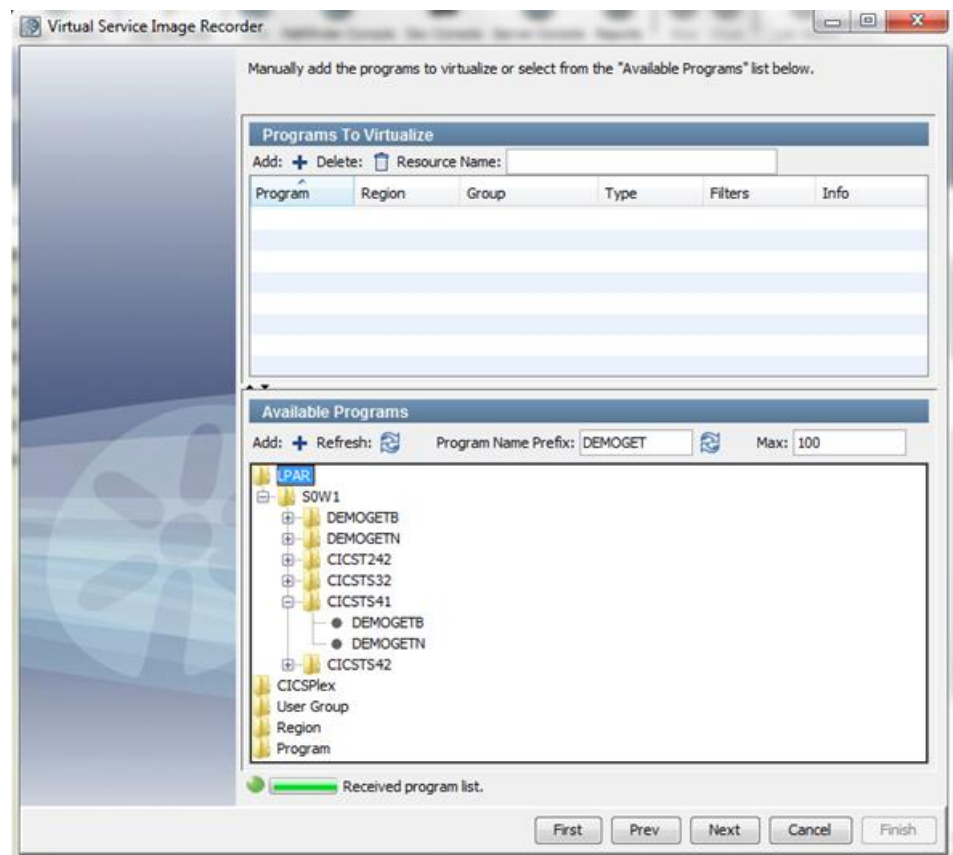
Les VTAM APPLIDs (ID d'application VTAM) identifient les régions CICS.

Program Name Prefix (Préfixe du nom de programme)

Permet de définir le préfixe utilisé par l'enregistreur pour filtrer les programmes à afficher. La zone Available Programs (Programmes disponibles) affiche uniquement les programmes qui commencent par le préfixe spécifié. Ajoutez toujours un préfixe aux programmes.

Max

Nombre maximum de programmes à afficher pour une région CICS

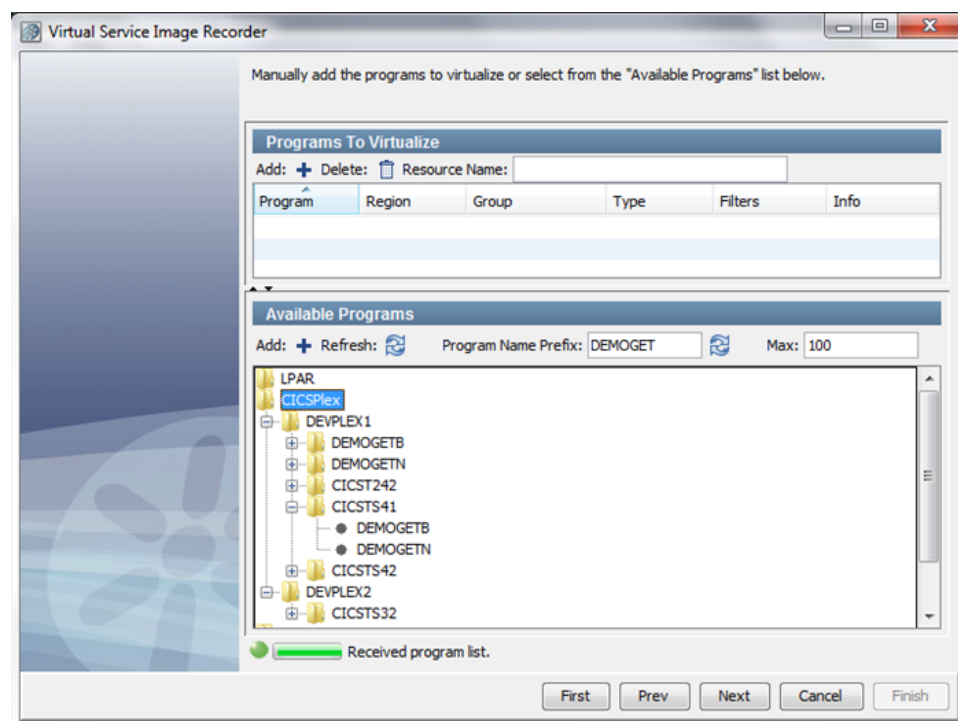


Cliquez sur LPAR (partition logique) pour afficher les agents LPAR connus (dans ce cas, SOW1).

Cliquez sur le signe plus à côté de SOW1 pour afficher les régions CICS connues dans SOW1: CICST242, CICSTS32, CICSTS41 et CICSTS42.

Cliquez sur le signe plus à côté de CICSTS41 pour afficher les programmes connus qui correspondent au préfixe DEMOGET dans CICSTS41 (à savoir, DEMOGETB et DEMOGETN).

DEMOGETB et DEMOGETN apparaissent également dans la liste sous SOW1, car il s'agit de programmes détectés dans la partition logique.



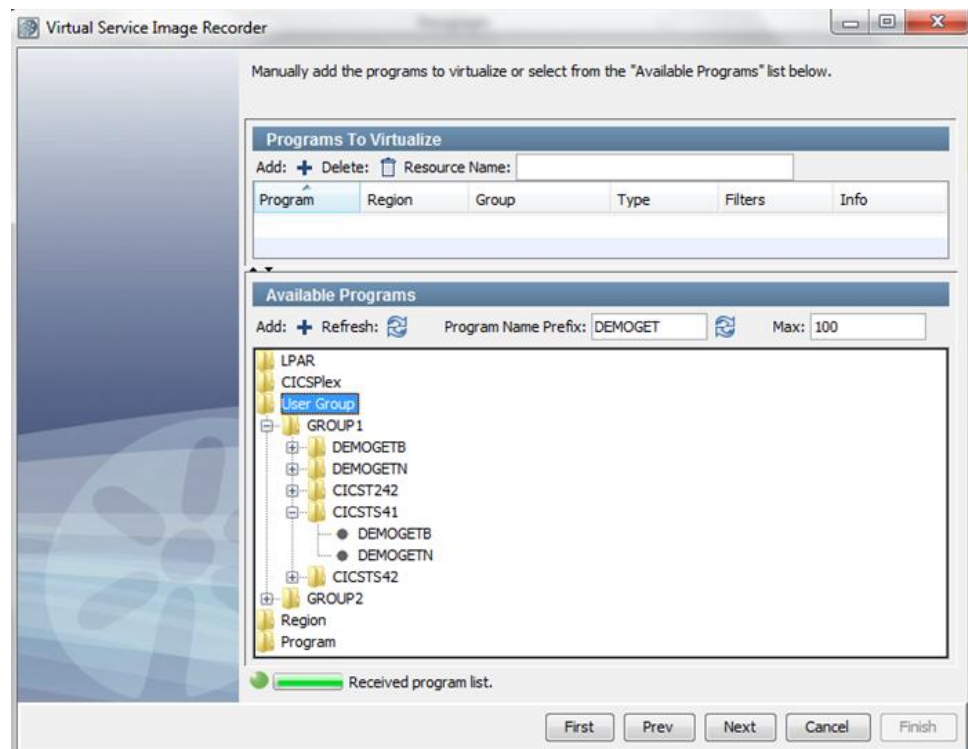
Cliquez sur CICSPlex pour afficher les CICSPlexs connus, ainsi que DEVPLEX1 et DEVPLEX2.

Cliquez sur le signe plus à côté de DEVPLEX1 pour afficher les régions CICS connues, à savoir CICST242, CICSTS41 et CICSTS42.

Cliquez sur le signe plus à côté de CICSTS41 pour afficher les programmes connus qui correspondent au préfixe dans CICSTS41 (à savoir, DEMOGETB et DEMOGETN).

DEMOGETB et DEMOGETN apparaissent également dans la liste sous DEVPLEX1, car il s'agit de programmes détectés dans CICSPlex.

Remarque : Pour convertir une région CICS en membre d'un groupe CICSPlex, une sortie d'utilisateur de l'agent CICS pour DevTest doit être modifiée. Pour plus d'informations, consultez la section Sorties d'utilisateur de l'agent CICS dans la rubrique *Agents*.



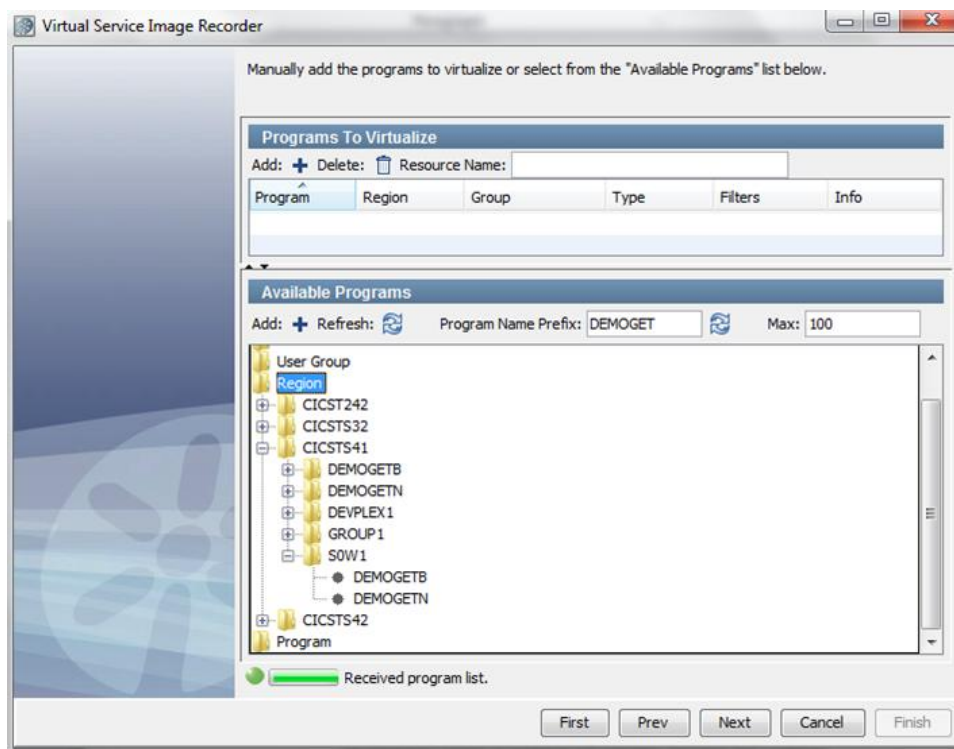
Cliquez sur User Group (Groupe d'utilisateurs) pour afficher les groupes d'utilisateurs connus et les groupes GROUP1 et GROUP2.

Cliquez sur le signe plus à côté de GROUP1 pour afficher les régions CICS connues, à savoir CICST242, CICSTS41 et CICSTS42.

Cliquez sur le signe plus à côté de CICSTS41 pour afficher les programmes connus qui correspondent au préfixe dans CICSTS41 (à savoir, DEMOGETB et DEMOGETN).

DEMOGETB et DEMOGETN apparaissent également dans la liste sous GROUP1, car il s'agit de programmes détectés dans le groupe d'utilisateurs.

Remarque : Pour convertir une région CICS en membre d'un groupe d'utilisateurs, une sortie d'utilisateur de l'agent CICS pour DevTest doit être modifiée. Pour plus d'informations, consultez la section Sorties d'utilisateur de l'agent CICS dans la rubrique *Agents*.

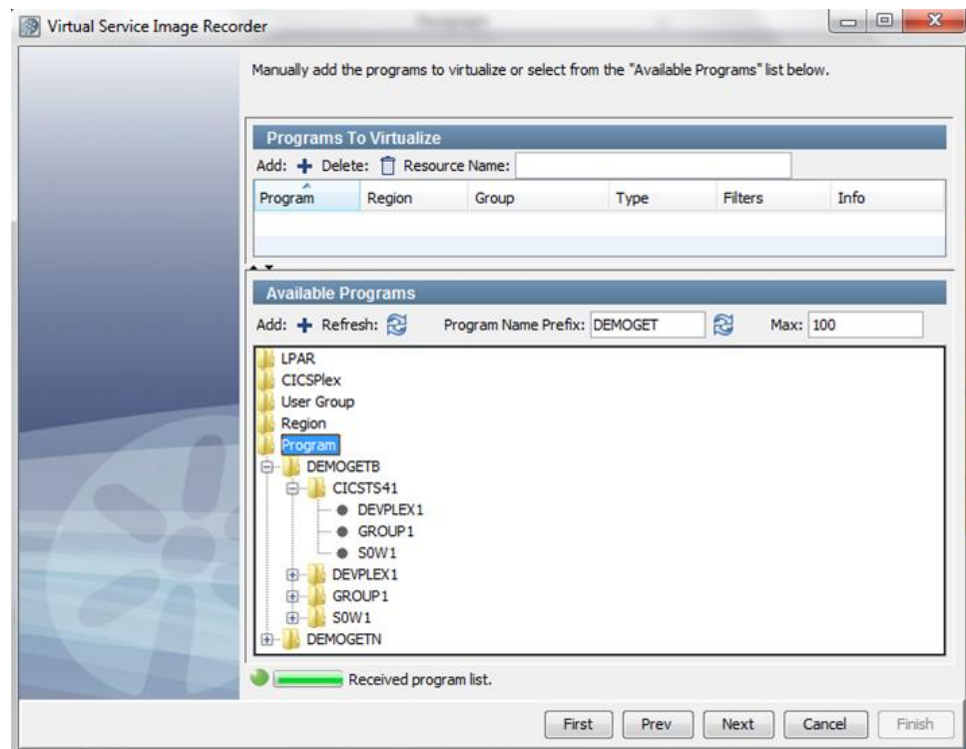


Cliquez sur Region pour afficher les régions CICS connues et les régions CICST242, CICSTS32, CICSTS41 et CICSTS42.

Cliquez sur le signe plus près de CICSTS41 pour afficher les groupes contenant cette région CICS.

Cliquez sur le signe plus à côté de SOW1 (groupe de partition logique dont cette région est membre) pour afficher les programmes connus correspondant au préfixe dans CICSTS41 (DEMOGETB et DEMOGETN).

DEMOGETB et DEMOGETN apparaissent également dans la liste sous GROUP1, car il s'agit de programmes détectés dans le groupe d'utilisateurs.




Après avoir récupéré les informations sur les programmes en cliquant sur une région CICS, le dossier de groupement par programmes s'affiche.

Cliquez sur Program pour afficher tous les programmes CICS connus ; dans le cas présent, DEMOGETB et DEMOGETN.

Cliquez sur le signe plus à côté de DEMOGETB pour développer les régions CICS connues et les groupes contenant DEMOGETB.

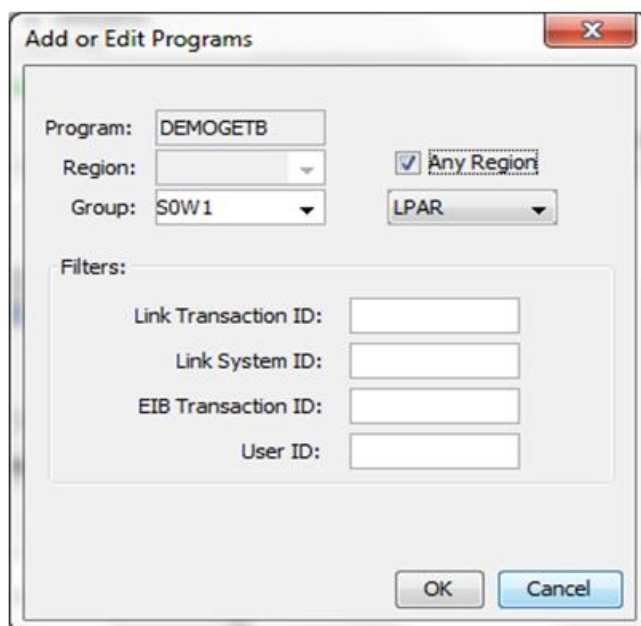
Programmes à virtualiser

Pour ajouter des programmes, double-cliquez dessus : ils s'afficheront dans le panneau Programs to Virtualize (Programmes à virtualiser). Vous pouvez également cliquer sur Add (Ajouter)  dans le panneau Programs to Virtualize (Programmes à virtualiser) pour ajouter un programme manuellement.

Lorsqu'un programme est répertorié dans le panneau Programs to Virtualize (Programmes à virtualiser), double-cliquez dessus pour ajouter des filtres et des groupements.

Dans cet exemple, le programme DEMOGETB est virtualisé dans la région CICS CICSTS41. Le groupe S0W1 et la partition logique sont ignorés, sauf si la case à cocher Any Region (Région quelconque) est sélectionnée.

Les filtres permettent de spécifier les liens CICS LINK vers DEMOGETB qui sont virtualisés.



Dialog box titled "Add or Edit Programs".

Fields:

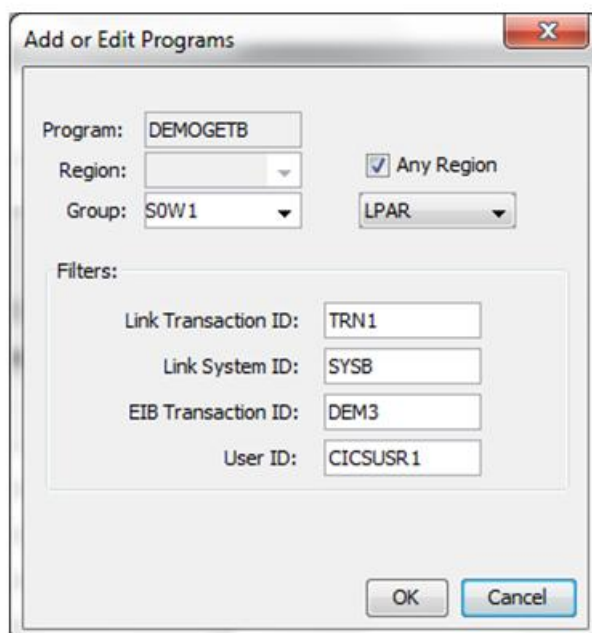
- Program: DEMOGETB
- Region: [Empty dropdown]
- Group: S0W1
- LPAR: [Empty dropdown]
- Any Region: ☒

Filters:

- Link Transaction ID: [Empty text box]
- Link System ID: [Empty text box]
- EIB Transaction ID: [Empty text box]
- User ID: [Empty text box]

Buttons: OK, Cancel

Sélectionnez la case à cocher Any Region (Région quelconque) pour effacer la zone de texte Region et indiquer que DEMOGETB sera virtualisé dans toutes les régions CICS du groupe de partition logique S0W1 (toutes les régions CICS résidant dans la partition logique S0W1).



Dialog box titled "Add or Edit Programs".

Fields:

- Program: DEMOGETB
- Region: [Empty dropdown]
- Group: S0W1
- LPAR: [Empty dropdown]
- Any Region: ☒

Filters:

- Link Transaction ID: TRN1
- Link System ID: SYSB
- EIB Transaction ID: DEM3
- User ID: CICSUSR1

Buttons: OK, Cancel

Dans le graphique précédent, les filtres sont utilisés pour réduire le nombre de programmes à virtualiser.

Link Transaction ID (ID de transaction de la liaison)

Indique que seuls les liens CICS LINK vers DEMOGETB codés avec TRANSID(TRN1) seront virtualisés.

Link System ID (ID de système de la liaison)

Indique que les liens CICS LINK vers DEMOGETB seront virtualisés uniquement s'ils sont codés avec SYSID(SYSB).

EIB Transaction ID (ID de transaction SAP)

Indique que seuls les liens CICS LINK vers DEMOGETB exécutés sous la transaction DEM3 seront virtualisés.

User ID (ID de l'utilisateur)

Indique que seuls les liens CICS LINK vers DEMOGETB exécutés sous l'ID d'utilisateur CICSUSR1 seront virtualisés.

Enregistrement d'images de la passerelle de transaction CICS (ECI)

Pour enregistrer des images de la passerelle de transaction CICS (ECI) :

1. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, sélectionnez CICS Transaction Gateway (ECI) (Passerelle de transaction CICS (ECI)) comme protocole de transport.
2. Remplissez les champs dans l'onglet Basics (Paramètres de base) et cliquez sur Next (Suivant).

L'étape suivante de l'enregistreur s'ouvre.

3. Entrez les informations du port et de l'hôte pour cette étape.

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Indique le numéro du port cible écouté par le serveur.

Expect SSL From Clients (Attendre une connexion SSL des clients)

Si cette option est sélectionnée, l'enregistreur attend que des clients s'y connectent à l'aide d'une connexion SSL. Le référentiel de clés et le mot de passe associés (s'ils ont été fournis) sont utilisés pour obtenir le matériel de sécurité (par exemple, des certificats).

Initiate SSL to Server (Initialiser une connexion SSL au serveur)

Si cette option est sélectionnée, l'enregistreur se connectera au système réel à l'aide d'une connexion SSL. Le référentiel de clés et le mot de passe associés (s'ils ont été fournis) sont utilisés pour obtenir le matériel de sécurité (par exemple, des certificats).

SSL keystore file (Fichier de référentiel de clés SSL)

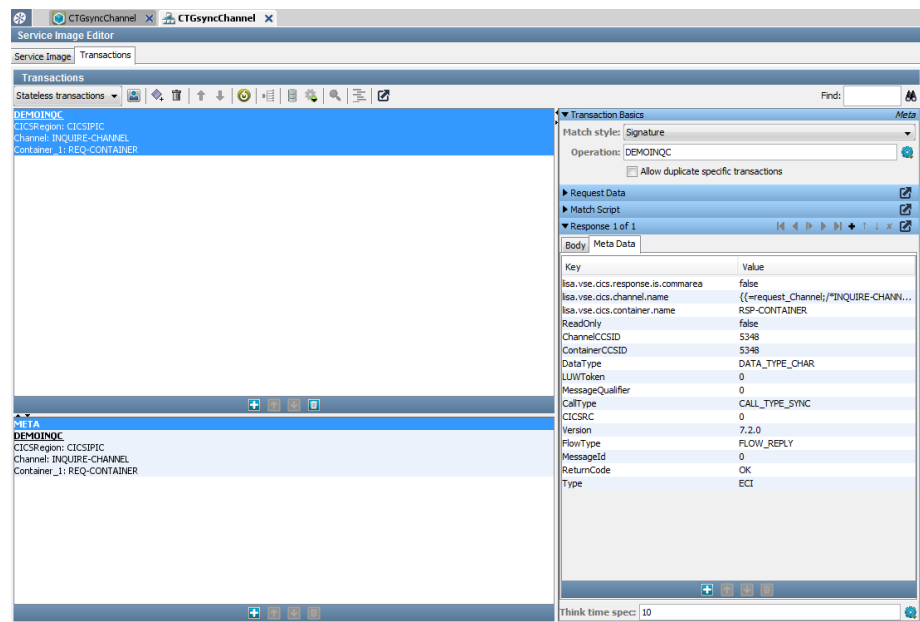
Indique le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

4. Pour afficher la fenêtre d'enregistrement, cliquez sur Next (Suivant).
5. Lancez l'application qui communique avec le serveur CTG et effectuez les opérations que vous voulez enregistrer.
6. Lorsque les transactions enregistrées apparaissent dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel, cliquez sur Next (Suivant)).
La fenêtre Data Protocols (Protocoles de données) apparaît.

7. Cliquez sur Next (Suivant).
La fenêtre Conversation Starter (Démarrage de conversations) s'affiche.
8. Cliquez sur Next (Suivant).
9. Sélectionnez les options pour afficher le modèle de service virtuel et l'image de service virtuel, puis cliquez sur Finish (Terminer).
10. L'éditeur d'images de service affiche l'image de service CTG.



Enregistrement d'images de service IMS Connect

Pour enregistrer des images de service IMS Connect :

1. Avant de lancer l'enregistreur, ajoutez les propriétés suivantes dans **local.properties**, le cas échéant :
 - Si le système testé envoie des réponses dont le message contient un champ de longueur LLLL de quatre octets au début, définissez la propriété suivante :
`lisa.vse.protocol.ims.response.includes.llll=true`
 - Si le système testé reçoit des demandes dont le message contient un champ de longueur LLZZ de quatre octets au début, définissez la propriété suivante :
`lisa.vse.ims.connect.llzz.request=true`
 - Si le système testé envoie des réponses dont le message contient un champ de longueur LLZZ de quatre octets au début, définissez la propriété suivante :
`lisa.vse.ims.connect.llzz.response=true`
 - Si le système testé utilise des copybooks pour analyser les charges utiles, ajoutez la propriété suivante :
`lisa.vse.copybook.unknown.passthrough=true`

Ainsi, les réponses binaires qui représentent des messages d'accusé de réception (sans données de charge utile) seront omises du traitement de copybook.

2. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, sélectionnez IMS Connect comme protocole de transport.
3. Remplissez les champs dans l'onglet Basics (Paramètres de base) et cliquez sur Next (Suivant).

L'étape suivante de l'enregistreur s'ouvre.

4. Entrez les informations du port et de l'hôte :

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Permet de définir le numéro du port cible écouté par le serveur. Si vous prévoyez de sélectionner un style d'authentification directe proxy, laissez ce champ vide.

Valeur par défaut : 80 (HTTP) et 443 (HTTPS)

IMS Format File (Fichier au format IMS)

Définissez l'un des éléments suivants :

- Le nom du fichier prenant en charge les demandes d'IMS Connect qui utilisent l'en-tête de 160 octets utilisé par l'exemple d'application IBM
- Un en-tête de 80 octets

Les définitions de champ pour les en-têtes de 160 octets et 80 octets pris en charge sont fournies dans le fichier **ims.format** disponible dans le répertoire LISA_HOME.

Pour utiliser la prise en charge d'IMS Connect fournie avec DevTest par défaut, laissez ce champ vide.

Character Set (Jeu de caractères)

Définit la méthode de codage des données par l'application à virtualiser.

Valeurs :

- ASCII - CP1252
- EBCDIC - CP037

5. Pour afficher la fenêtre d'enregistrement, cliquez sur Next (Suivant).
6. Lancez l'application qui communique avec le serveur IMS Connect et effectuez les opérations que vous voulez enregistrer.

Vérifiez que l'hôte et le port de cette application correspondent à l'hôte et au port de l'enregistreur (localhost:8001 en général).
7. Lorsque les transactions enregistrées s'affichent dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel), cliquez sur Next (Suivant).

La fenêtre Data Protocols (Protocoles de données) s'ouvre.

Par défaut, si le champ d'en-tête TransactionCode (Code de transaction) existe, il est défini comme opération de la demande. De même, TransactionCode (Code de transaction), DestinationId (ID de destination) et ClientId (ID de client) sont ajoutés comme arguments, si ces champs existent dans la définition du fichier de format.
8. Ajoutez des protocoles de données, y compris le protocole de données de copybook si nécessaire et cliquez sur Next (Suivant).

La fenêtre Conversation Starter (Démarrage de conversations) s'ouvre.
9. Cliquez sur Next (Suivant).
10. Sélectionnez les options pour afficher le modèle de service virtuel et l'image de service virtuel, puis cliquez sur Finish (Terminer).

L'éditeur d'images de service affiche l'image de service IMS Connect.

Fichiers de format personnalisés IMS Connect

Si le système testé utilise des en-têtes de demande différents de ceux pris en charge par défaut par DevTest, avant de lancer l'enregistrement, procédez comme suit :

1. Créez un fichier **<custom-format>.format** dans le dossier Data (Données) du projet DevTest.
2. Modifiez le fichier dans un éditeur de texte et ajoutez-y le texte suivant. Indiquez vos définitions de champ sous la zone RequestUserHeader (En-tête de demande d'utilisateur).

```
#-----
# The IMS™ request message (IRM) header contains a 28-byte fixed-format
# section that is common to all messages from all IMS Connect client applications
# that communicate with IMS TM.
#-----
RequestHeaderCommon {
    LLLL                                int;                                #total message length IRM + user data
    IRMFixedHeader {
        LL                                short;                                #IRM_LEN, total length of the header segment
including user header portion
        ZZ                                byte;                                #IRM_ARCH
        Flag0                            byte;                                #IRM_F0
        UserExitId                        string(8);                        #IRM_ID
        NakReasonCode                    byte(2);                        #IRM_NAK_RSNCDE
        Reserved1                        byte(2);                        #IRM_RES1
        MessageType                      byte;                                #IRM_MessageType
        WaitTime                         byte;                                #IRM_TIMER
        SocketConnectionType             byte;                                #IRM_SOCT
        EncodingSchema                   byte;                                #IRM_ES
        ClientId                         byte(8);                        #IRM_CLIENTID
    }
}

#-----
# Format of user portion of IRM some custom header
#
# Following the 4-byte length field and the 28-byte fixed portion of the
# IMS™ request message (IRM) header in IMS Connect client input messages,
# user-written client applications can include a user-defined section in the IRM.
#
#-----
RequestUserHeader {
    MyFlag1                            byte;
    MyFlag2                            byte;
    MyFlag3                            byte;
    MyFlag4                            byte;
    TransactionCode                     string(8);
    DestinationId                      string(8);
}
```

```

    LogicalTerminal      string(8);
    Miscellaneous        byte(20);
}

#-----
# Format of data segments of request message
#-----
RequestPayloadSegment {
    LL          short;
    ZZ          byte(2);
    Data        byte(LL:inclusive);
}

#-----
# Format of header for response message. Some responses will have it, some won't
#-----
ResponseMessageHeader {
    LLLL          int;          # followed by multiple ResponsePayloadSegment
}

#-----
# Format of data segments of request message
#-----
ResponsePayloadSegment {
    LL          short;
    ZZ          byte(2);
    Data        byte(LL:inclusive);
}

```

Une fois le fichier créé et enregistré dans le dossier Data (Données), il s'affichera dans le champ IMS Format File (Fichier de format IMS) de l'enregistreur. Vous pouvez sélectionner le fichier de format pour des applications qui utilisent des en-têtes de demande autre que ceux par défaut.

Enregistrement de RFC SAP via JCo

Procédez comme suit:

1. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, sélectionnez SAP RFC via JCo comme protocole de transport.
2. Remplissez les champs dans l'onglet Basics (Paramètres de base) et cliquez sur Next (Suivant).

L'étape suivante de l'enregistreur s'ouvre.

3. Spécifiez les informations de connexion.

Client System Name (Nom du client)

Spécifie un nom unique pour identifier le client SAP.

Client System Connection Properties (Propriétés de connexion au client)

Le fichier de propriétés de connexion au client contient les propriétés de connexion à la destination sur le système client. Il doit s'agir d'un fichier .properties du répertoire Data (Données) de votre projet qui contient les propriétés génériques d'un fichier .jcoServer. Ce fichier ne doit pas spécifier la propriété `jco.server.repository_destination`. Pour plus d'informations sur les propriétés prises en charge, consultez le document JavaDocs

com.sap.conn.jco.ext.ServerDataProvider dans le dossier **doc** de votre répertoire d'installation.

RFC Repository Name (Nom de référentiel RFC)

Spécifie un nom unique pour identifier le système SAP comprenant le référentiel pour le modèle RFC. Ce nom est souvent le même que le système de destination.

Propriétés de connexion au référentiel RFC

Spécifie le fichier de propriétés de connexion qui contient les propriétés de connexion au système contenant le référentiel. Il doit s'agir d'un fichier .properties du répertoire Data (Données) de votre projet et contenir les propriétés génériques d'un fichier .jcoDestination. Pour plus d'informations sur les propriétés prises en charge, consultez le document JavaDocs

com.sap.conn.jco.ext.DestinationDataProvider dans le dossier **doc** de votre répertoire d'installation.

Destination System Name (Nom du système de destination)

Définit un nom unique pour identifier le système SAP sur lequel le RFC est exécuté. Cette valeur est souvent identique au nom du référentiel.

Destination System Connection Properties (Propriétés de connexion au système de destination)

Spécifie le fichier de propriétés de connexion au système de destination qui contient les propriétés de connexion permettant de se connecter système contenant le référentiel. Il doit s'agir d'un fichier `.properties` du répertoire Data (Données) de votre projet et contenir les propriétés génériques d'un fichier `.jcoDestination`. Pour plus d'informations sur les propriétés prises en charge, consultez le document JavaDocs **com.sap.conn.jco.ext.DestinationDataProvider** dans le dossier **doc** de votre répertoire d'installation. Il peut s'agir du même fichier que le fichier de propriétés de connexion de référentiel.

Function Name (Nom de la fonction)

Spécifie le nom du référentiel RFC que DevTest doit intercepter.

4. Pour afficher la fenêtre d'enregistrement, cliquez sur Next (Suivant).
5. Lancez l'application qui communique avec le serveur SAP et effectuez les opérations que vous voulez enregistrer.
6. Lorsque les transactions enregistrées apparaissent dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel, cliquez sur Next (Suivant).
La fenêtre Data Protocols (Protocoles de données) s'ouvre.
7. Cliquez sur Next (Suivant).
La fenêtre Conversation Starter (Démarrage de conversations) s'ouvre.
8. Cliquez sur Next (Suivant).
9. Sélectionnez les options pour afficher le modèle de service virtuel et l'image de service virtuel, puis cliquez sur Finish (Terminer).

L'éditeur d'images de service affiche l'image de service RFC SAP.

Record JCo IDoc (Enregistrer une image de service virtuel de service Web)

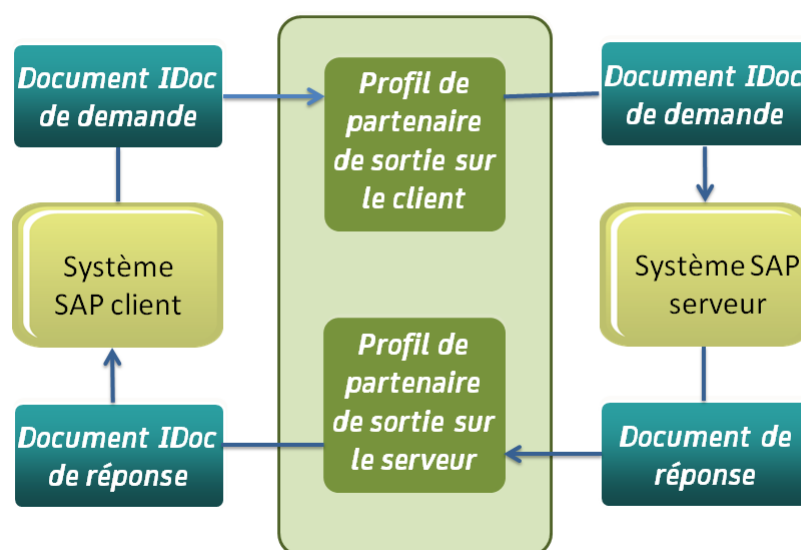
Le protocole de transport JCo SAP IDoc vous permet de créer des services virtuels qui utilisent ce protocole.

Le protocole JCo IDoc prend en charge la création d'un service virtuel qui enregistre et lit les fichiers IDoc envoyés à partir d'un système SAP à un autre à l'aide d'une destination RFC.

Le système SAP qui initialise les communications et envoie les fichiers IDoc est appelé le client SAP. Les fichiers IDoc envoyés sont appelés des fichiers IDoc de demande.

Le système SAP qui reçoit les fichiers IDoc de demande est appelé un serveur SAP. Les fichiers IDoc que le serveur SAP renvoie sont appelés des fichiers IDoc de réponse.

Le graphique suivant présente le flux standard des fichiers IDoc entre deux systèmes SAP.



Conditions requises : L'utilisation de DevTest avec cette application requiert que la mise à disposition d'un ou plusieurs fichiers pour DevTest. Pour plus d'informations, consultez la section Conditions requises pour les fichiers des applications tierces de la rubrique *Administration*.

Pour enregistrer des images de service JCo IDoc :

1. Dans l'onglet Basics (Paramètres de base) de l'enregistreur d'image de service virtuel, sélectionnez JCo IDoc Protocol (Protocole IDoc JCo) comme protocole de transport.
2. Remplissez les champs dans l'onglet Basics (Paramètres de base) et cliquez sur Next (Suivant).

L'étape suivante de l'enregistreur s'ouvre.

3. Spécifiez les informations de connexion.

Client RFC Connection Properties (Propriétés de connexion RFC du client)

Définit le fichier de propriétés de connexion RFC du client qui contient les propriétés de connexion que VSE utilise pour s'enregistrer sous un ID de programme dans une passerelle SAP et recevoir des fichiers IDoc. Les propriétés doivent être identiques à celles spécifiées dans un fichier .jcoServer.

Client RFC Destination Name (Nom de destination RFC sur le client)

Spécifie un nom unique qui identifie la destination RFC.

Client System Connection Properties (Propriétés de connexion au client)

Spécifie le fichier de propriétés de connexion au client qui contient les propriétés de connexion permettant de renvoyer les fichiers IDoc au client SAP. Ces propriétés doivent être identiques à celles spécifiées dans un fichier .jcoDestination que vous pouvez utiliser pour vous connecter au client SAP.

Client System Name (Nom du client)

Spécifie un nom unique pour identifier le client SAP.

Request Identifier XPath Expressions (Expressions XPath d'identificateur de demande)

Spécifie les expressions XPath que le protocole utilise avec la demande XML IDoc pour générer un identificateur. Les expressions XPath d'identificateur de demande peuvent être une expression XPath unique. Cet identificateur est utilisé pour mettre en corrélation une demande IDoc avec une réponse IDoc. Les expressions XPath peuvent également consister en une liste d'expressions XPath séparées par des virgules ; auquel cas, les valeurs résultantes de plusieurs expressions sont concaténées (séparées par des traits d'union) et utilisées comme identificateur.

Server RFC Connection Properties (Propriétés de connexion RFC du serveur)

Spécifie le fichier de propriétés qui contient les propriétés de connexion que VSE utilise pour s'enregistrer sous un ID de programme dans une passerelle SAP et recevoir des fichiers IDoc. Les propriétés doivent être identiques à celles spécifiées dans un fichier .jcoServer pour démarrer un programme de serveur JCo qui reçoit des fichiers IDoc à partir du serveur SAP.

Server RFC Destination Name (Nom de destination RFC sur le serveur)

Spécifie un nom unique pour identifier la destination RFC sur le serveur.

Server System Connection Properties (Propriétés de connexion au serveur)

Définit le fichier de propriétés de connexion au serveur qui contient les propriétés de connexion permettant de renvoyer les fichiers IDoc au serveur SAP. Ces propriétés doivent être identiques à celles spécifiées dans un fichier .jcoDestination que vous pouvez utiliser pour vous connecter au serveur SAP.

Server System Name (Nom du serveur)

Spécifie un nom unique pour identifier le serveur SAP.

Response Identifier XPath Expressions (Expressions XPath d'identificateur de réponse)

Définit les expressions XPath que le protocole utilise avec la réponse XML IDoc pour générer un identificateur. Les expressions XPath d'identificateur de réponse peuvent être une expression XPath unique. Cet identificateur est utilisé pour mettre en corrélation une réponse IDoc avec une demande IDoc reçue antérieurement. Les expressions XPath peuvent également consister en une liste d'expressions XPath séparées par des virgules ; auquel cas, les valeurs résultantes de plusieurs expressions sont concaténées (séparées par des traits d'union) et utilisées comme identificateur.

4. Pour ouvrir la fenêtre d'enregistrement, cliquez sur Next (Suivant).
5. Lancez l'application qui communique avec le serveur SAP et effectuez les opérations que vous voulez enregistrer.
6. Lorsque les transactions enregistrées s'affichent dans la fenêtre Virtual Service Image Recorder (Enregistreur d'image de service virtuel), cliquez sur Next (Suivant).

La fenêtre Data Protocols (Protocoles de données) s'ouvre.

Le protocole de données XML est sélectionné pour les protocoles de données côté demande, car VSE stocke les fichiers IDoc au format XML. Supprimez le protocole de données XML, car il remplace le nom d'opération défini par le protocole IDoc JCo.

7. Cliquez sur Next (Suivant).

La fenêtre Conversation Starter (Démarrage de conversations) s'ouvre.

8. Cliquez sur Next (Suivant).
9. Sélectionnez les options pour afficher le modèle de service virtuel et l'image de service virtuel, puis cliquez sur Finish (Terminer).

L'éditeur d'images de service affiche l'image de service IDoc JCo SAP.

Opaque Data Processing (Traitement de données opaques)

Le traitement de données opaques permet à CA Service Virtualization de virtualiser des détails suffisants de données lorsque le format des demandes et des réponses n'est pas connu. Le traitement de données opaques supprime la nécessité d'un nouveau gestionnaire de données chaque fois que vous rencontrez un nouveau format de message.

L'enregistrement de plusieurs demandes et des réponses correspondantes permet à CA Service Virtualization de déduire la structure de message. En d'autres termes CA Service Virtualization peut mettre en corrélation les octets d'une demande avec les octets correspondants d'une réponse, en attribuant le même comportement de chaîne magique disponible dans d'autres protocoles. CA Service Virtualization peut également suffisamment comprendre la structure de demande pour mettre en correspondance de manière intelligente de nouvelles demandes détectées lors de la lecture du service virtuel .

Plus bas, le traitement de données opaques utilise un algorithme breveté pour comparer une demande entrante avec toutes les demandes dans l'image de service traitement de données opaques enregistrée. La demande correspondance la plus proche est sélectionnée et la réponse correspondante est renvoyée, après la substitution d'une chaîne magique dynamique.

L'algorithme de correspondance du traitement de données opaques applique des poids entropiques lors du processus correspondant. Le processus de pondération entropique déduit quels sont les octets sont les plus importants du message. Par exemple, les octets qui correspondent au type d'opération, par opposition au reste de la charge utile ; une plus grande importance est attribuée à ces octets pendant le processus de correspondance. Le processus de pondération entropique fonctionne mieux avec des exemples plus volumineux de messages enregistrés (100 ou plus) et avec un échantillonnage divers de valeurs de paramètre.

L'algorithme de correspondance du traitement de données opaques et de réponse fonctionne avec des protocoles de message tant binaires que textuels. Les meilleurs résultats ont été obtenus avec des protocoles utilisant des champs de largeur fixe (par exemple IMS) ou des champs délimités (par exemple, protocoles XML). Une précision raisonnable a également été obtenue avec des formats de protocole codés en longueur (par exemple, ASN.1), mais ceux-ci constituent un défi majeur.

Le traitement des données opaques est pris en charge pour le trafic que vous pouvez capturer à l'aide de sockets TCP/IP bruts. Vous pouvez également importer les données à partir d'un fichier PCAP.


Utilisation du traitement de données opaques

Suivez les instructions suivantes pour enregistrer une image de service virtuel à l'aide du traitement de données opaques.

Pour plus d'informations sur les conditions préalables et la procédure de préparation, reportez-vous à la section [Virtualisation du protocole TCP](#) (page 183).

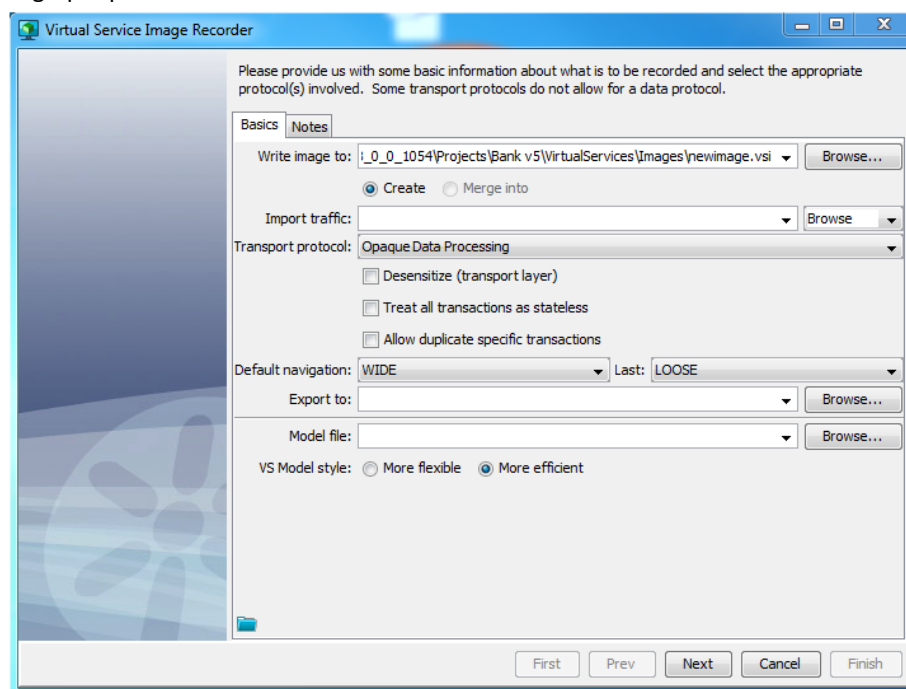
Procédez comme suit:

1. Ouvrez DevTest Workstation.
2. Pour lancer l'enregistrement d'une nouvelle image de service virtuel, effectuez l'une des étapes suivantes :

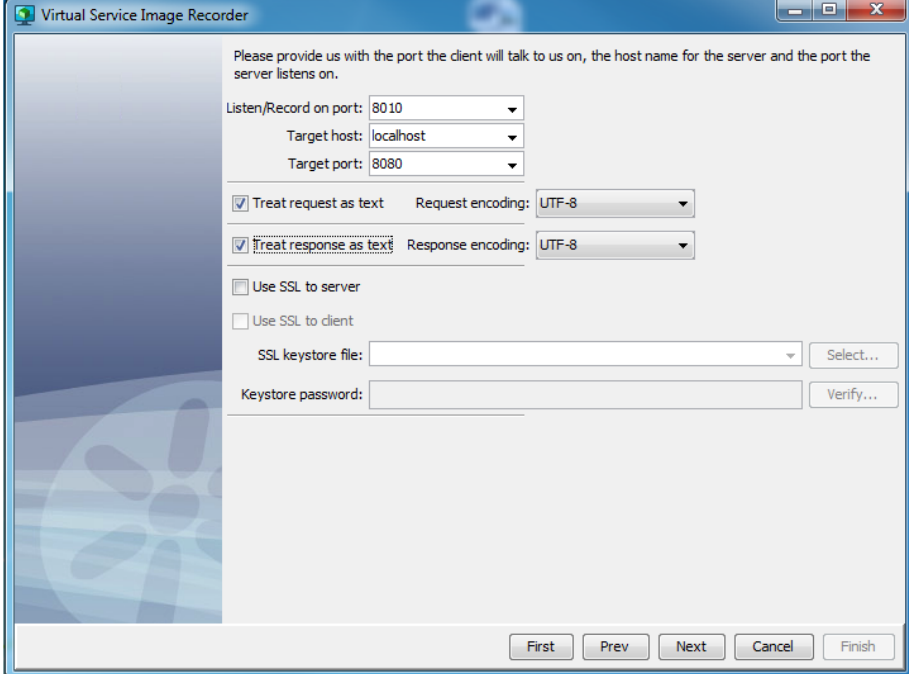
- Dans la barre d'outils principale, cliquez sur  VSE Recorder (Enregistreur de VSE).
- Cliquez avec le bouton droit de la souris sur le noeud VirtualServices dans le panneau Project (Projet) et sélectionnez Create a VS Image, by Recording (Créer une image de service virtuel à l'aide d'un enregistrement).

L'enregistreur d'image de service virtuel s'ouvre.

3. Sélectionnez l'onglet [Basics \(Paramètres de base\)](#) (page 131) comme indiqué dans le graphique suivant :



4. Cliquez sur Next (Suivant).
5. Saisissez les informations sur le client et le serveur, sélectionnez votre codage et ajoutez des paramètres de connexion SSL.



The screenshot shows the 'Virtual Service Image Recorder' dialog box. It contains the following fields and options:

- Listen/Record on port:** 8010
- Target host:** localhost
- Target port:** 8080
- ☒ **Treat request as text** Request encoding: UTF-8
- ☒ **Treat response as text** Response encoding: UTF-8
- ☐ **Use SSL to server**
- ☐ **Use SSL to client**
- SSL keystore file:** [empty field] Select...
- Keystore password:** [empty field] Verify...

At the bottom, there are navigation buttons: First, Prev, Next, Cancel, and Finish.

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Indique le numéro du port sur lequel le serveur écoute.

Treat request as text (Traiter la demande comme du texte)

Spécifie si la demande doit être traitée comme du texte. Pour plus d'informations, consultez la section [Préparation de la virtualisation du protocole TCP](#) (page 183).

Request Encoding (Codage de demande)

Répertorie les codages de demande disponibles sur l'ordinateur sur lequel la station de travail DevTest Workstation est exécutée. Il s'agit par défaut de la UTF8.

Treat response as text (Traiter la réponse comme du texte)

Spécifie si la réponse doit être traitée comme du texte. Pour plus d'informations, consultez la section [Préparation de la virtualisation du protocole TCP](#) (page 183).

Response Encoding (Codage de réponse)

Répertorie les codages de réponse disponibles sur l'ordinateur sur lequel la station de travail DevTest Workstation est exécutée. Il s'agit par défaut de la UTF8.

Use SSL to server (Utiliser une connexion SSL au serveur)

Permet de spécifier si DevTest utilise une connexion HTTPS pour envoyer la demande au serveur.

- Si cette option est **sélectionnée**, DevTest envoie une demande HTTPS (secured layer) au serveur.

Si vous sélectionnez Use SSL to server (Utiliser une connexion SSL au serveur) sans sélectionner Use SSL to client (Utiliser une connexion SSL au client), DevTest utilisera une connexion HTTP pour l'enregistrement. DevTest enverra alors ces demandes au serveur à l'aide d'une connexion HTTPS.
- Si l'option est **désélectionnée**, DevTest enverra une demande HTTP au serveur.

Use SSL to client (Utiliser une connexion SSL au client)

Spécifie si un référentiel de clés personnalisé doit être utilisé pour lire une demande SSL à partir d'un client. Cette option est activée uniquement si l'option Use SSL to server (Utiliser une connexion SSL au serveur) a été sélectionnée.

Valeurs :

- **Option sélectionnée** : vous pouvez spécifier un référentiel de clés client personnalisé et une phrase secrète. Si ces paramètres sont saisis, ils sont utilisés au lieu des valeurs par défaut codées de manière irréversible.
- **Option désélectionnée** : vous ne pouvez pas spécifier un référentiel de clés client personnalisé et une phrase secrète.

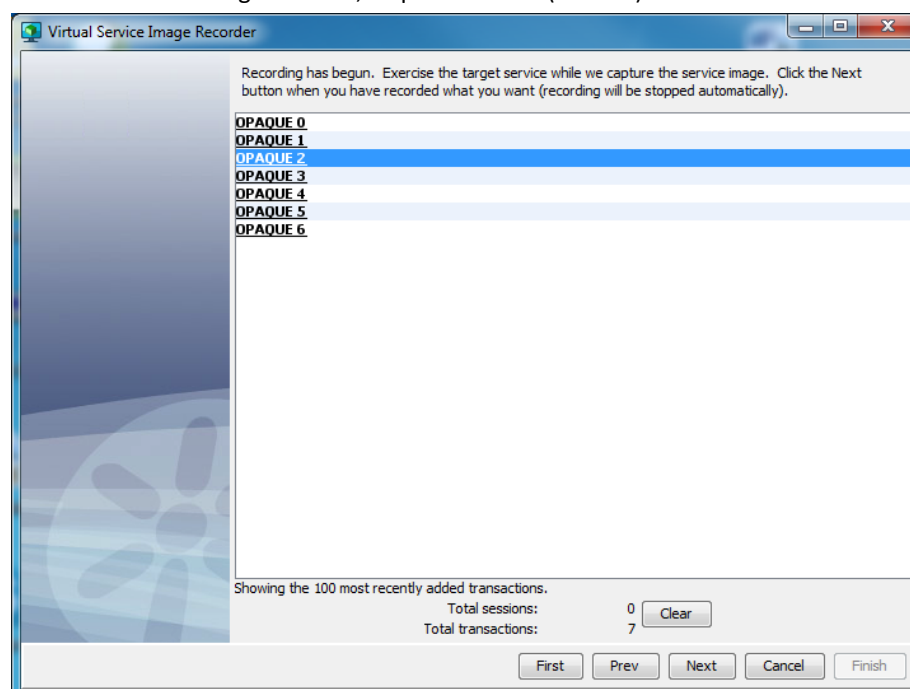
SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

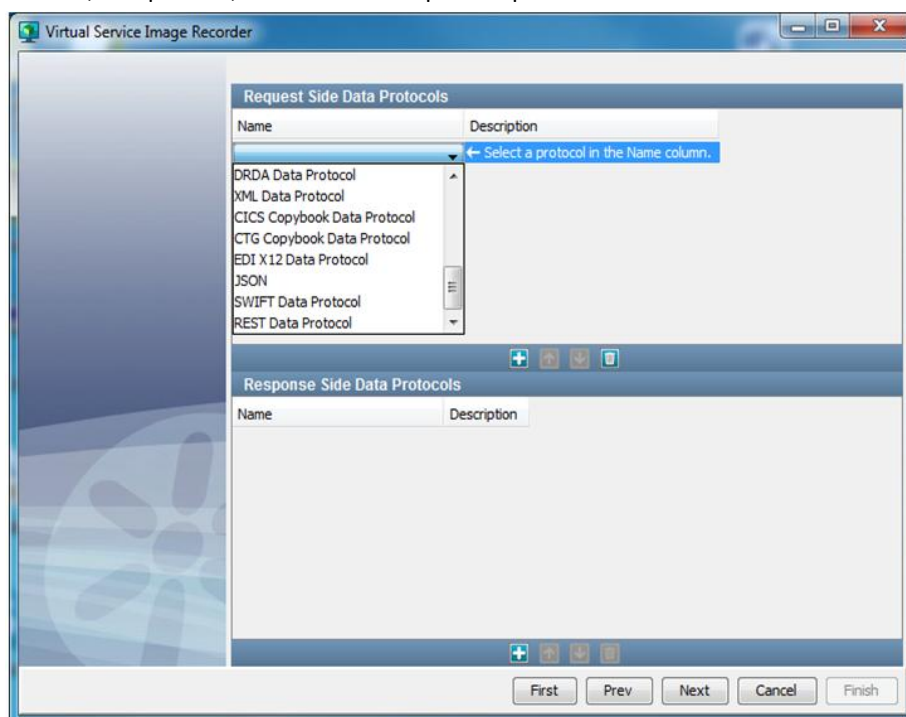
6. Pour démarrer l'enregistrement, cliquez sur Next (Suivant).



7. A l'issue de l'enregistrement, cliquez sur Next (Suivant).

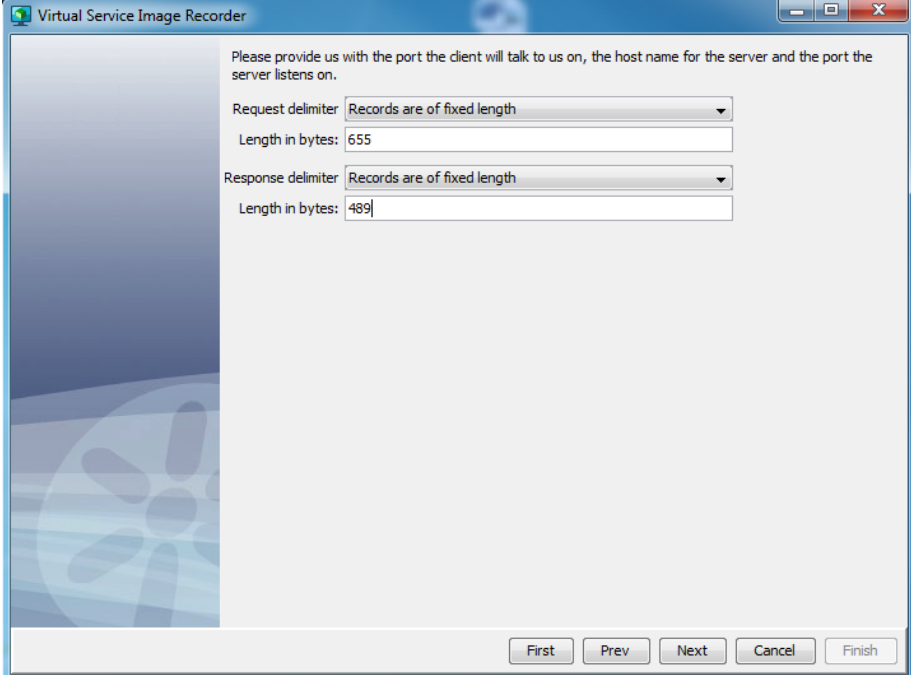
Le traitement de données opaques repose sur le corps de la demande, au lieu de l'opération et des arguments, pour la mise correspondance. Par conséquent, la plupart des protocoles de données de demande ne sont pas appropriés.

D'autre part, les protocoles de données de réponse conviennent si la réponse est chiffrée, compressée, ou codée d'une quelconque autre manière.



8. L'enregistreur tente de détecter des délimiteurs de message qui indiquent à DevTest qu'une demande ou une réponse complète a été lue. Confirmez ou corrigez ces délimiteurs dans cette fenêtre.

Remarque : Un délimiteur de demande est requis. Un délimiteur de réponse doit être sélectionné pour rendre l'appel dynamique disponible.



The screenshot shows the 'Virtual Service Image Recorder' window. It contains a text area on the left with a blue background and a right panel with configuration options. The right panel has a title bar with standard window controls. Below the title bar, there is a text prompt: 'Please provide us with the port the client will talk to us on, the host name for the server and the port the server listens on.' Below this, there are two sections: 'Request delimiter' and 'Response delimiter'. Each section has a dropdown menu set to 'Records are of fixed length' and a text input field for 'Length in bytes'. The 'Request delimiter' field contains '655' and the 'Response delimiter' field contains '489'. At the bottom of the window, there are five buttons: 'First', 'Prev', 'Next', 'Cancel', and 'Finish'.

9. Après l'enregistrement, l'enregistreur vérifie que les corps des demandes et des réponses sont du texte, s'ils sont marqués comme tel. Sinon, le type devient binaire.

Protocole SSL pour les enregistrements de VSE

Pendant l'enregistrement de VSE, l'application de serveur SSL est le système testé.

Use SSL to server (Utiliser une connexion SSL au serveur)

Si seule l'option Use SSL to server (Utiliser une connexion SSL au serveur) est sélectionnée pendant la configuration de l'enregistreur (sans l'option Use SSL to Client (Utiliser une connexion SSL au client)), le client enverra une connexion HTTP simple à DevTest, mais l'enregistreur enverra une demande HTTPS (connexion sécurisée via SSL) à l'application de serveur. Dans ce cas, *client* fait référence à l'application ou au scénario de test qui envoie la demande.

La clé publique de certificat pour l'application de serveur SSL doit se trouver dans le référentiel d'approbations de DevTest pour permettre l'authentification du certificat de serveur (ou fournir une authentification unidirectionnelle).

Si le serveur SSL demande l'authentification du client (authentification bidirectionnelle), LISA v6.0.8 (et version ultérieure) utilise le référentiel de clés identifié dans **ssl.client.cert.path** pour envoyer un certificat de client à partir de l'enregistreur vers l'application de serveur SSL. L'enregistreur simule un client pour effectuer l'authentification de client bidirectionnelle, lorsque l'option Use SSL to Server (Utiliser une connexion SSL au serveur) est sélectionnée.

Use SSL to the Server (Utiliser une connexion SSL au serveur) et Use SSL to Client (Utiliser une connexion SSL au client)

Pour effectuer des enregistrements, le client ou le scénario de test envoie des données au port d'écoute ou d'enregistrement configuré. L'hôte cible de l'enregistreur correspond au serveur SSL et le port cible au port SSL que le serveur SSL utilise (en général, 443 ou 8443).

Pendant l'enregistrement, l'établissement de liaison SSL est effectué entre le client (enregistreur) et le serveur SSL. Le serveur envoie son certificat et DevTest l'authentifie. Si le serveur demande l'authentification du client, le certificat qui se trouve dans le fichier **local.properties** est utilisé. S'il n'y a aucun référentiel de clés valide dans **ssl.client.cert.path** et le serveur demande l'authentification du client, une situation de certificat erroné est renvoyée, car l'enregistreur du client ne peut renvoyer aucun certificat au serveur.

Pendant l'enregistrement, l'établissement de liaison SSL est effectué entre le client (application ou scénario de test) et le serveur (enregistreur). L'enregistreur envoie le certificat spécifié dans le référentiel de clés Use SSL to Client (Utiliser une connexion SSL au client). Si le champ SSL Keystore File (Fichier de référentiel de clés SSL) est vide, le référentiel de clés par défaut ({LISA_HOME})\webrekeys.ks est utilisé. Le serveur d'enregistreur ne demande pas l'authentification du client. L'établissement d'une liaison est une authentification unidirectionnelle.

Lecture du modèle de service virtuel

Si vous sélectionnez l'option Use SSL to Client (Utiliser une connexion SSL au client), un établissement de liaison SSL est effectué entre l'application cliente ou le client du scénario de test et le modèle de service virtuel. Le référentiel de clés fourni dans l'étape Listen (Ecouter) du modèle de service virtuel est utilisé comme certificat du serveur pour l'authentification unidirectionnelle. Il n'y a aucun établissement de liaison SSL entre le client et le modèle de service virtuel. L'établissement de liaison est une connexion HTTP simple.

Si l'étape Live Invocation (Appel dynamique) a été exécutée, un établissement de liaison SSL est effectué entre le client de modèle de service virtuel et le serveur réel. Si le serveur réel demande l'authentification du client, le référentiel de clés de l'étape HTTP/S Protocol Live Invocation (Appel dynamique via le protocole HTTP/S) est utilisé.

Si le référentiel de clés contient plusieurs certificats, VSE utilise le premier.

Création et déploiement d'un service virtuel avec VSEasy

Vous pouvez rapidement créer et déployer un service virtuel à l'aide de la console de VSEasy.

VSEasy prend en charge les modes suivants:

- Protocole HTTP
- Fichier VRS

Vous pouvez utiliser VSEasy pour créer des services virtuels HTTP sans état avec des charges utiles SOAP, JSON, XML ou HTML.

Pour configurer des services virtuels autres que HTTP, utilisez des fichiers VRS.

Lorsque vous utilisez un fichier VRS, vous pouvez placer les protocoles de données dans une chaîne. Par exemple, vous pouvez utiliser les protocoles de données de l'analyseur de charge utile XML générique ou du gestionnaire de données de demande pour modifier l'image de service.

Le fichier VRS ne peut pas référencer de données externes. Par exemple, le fichier VRS ne peut pas référencer de copybooks ni de certificats pour la signature.

Dans la console VSEasy, vous devez assigner le service virtuel à un groupe de services virtuels. Lorsque vous sélectionnez un serveur VSE, les groupes disponibles s'affichent dans la liste **Service Group (Groupe de service)**. Le groupe par défaut est nommé **VSEasy**. Pour changer le nom du groupe par défaut, mettez à jour la propriété **lisa.vseasy.default.group.tag** dans le fichier **lisa.properties**.

Remarque : Si vous déployez un service virtuel dont le nom et le port sont identique à ceux d'un service virtuel déjà déployé dans le serveur VSE, le service virtuel existant sera écrasé.

Les navigateurs suivants sont pris en charge :

- Internet Explorer 10 ou 11
- Mozilla FireFox 24
- Google Chrome
- Safari 5

Procédez comme suit:

1. Vérifiez qu'un serveur de VSE est en cours d'exécution.
2. Dans DevTest Workstation, sélectionnez l'icône Server Console (Console de serveur) dans la barre d'outils.
3. A partir de la console de serveur, sélectionnez VSEasy.
La console VSEasy s'ouvre.

4. Suivez les instructions de l'Aide dans le panneau droit. L'Aide inclut également les exemples de fichier suivants :
 - Paire demande-réponse HTTP
 - Paires demande-réponse JMS et fichier VRS

Utilisation des modèles de service virtuel

Un modèle de service virtuel est un type de scénario de test spécialisé qui devient le terminal d'un service virtualisé. Pour lancer l'enregistreur d'image de service virtuel, créez un modèle de service virtuel.

Créez un fichier VSM (trap rule file, fichier de règles d'interruption)

Procédez comme suit:

1. Ouvrez un projet existant ou créez un projet.

Le projet s'ouvre et le panneau Project (Projet) à gauche contient le dossier et les sous-dossiers du projet.
2. Cliquez avec le bouton droit de la souris sur le noeud du sous-dossier VirtualServices et sélectionnez Create New VS Model (Créer un modèle de service virtuel).

Vous pouvez également créer un modèle de service virtuel dans un autre dossier, mais il est recommandé de le créer sous le dossier VirtualServices.

La fenêtre VS Model Editor (Editeur de modèles de service virtuel) s'ouvre.
3. Recherchez l'emplacement du nouveau modèle de service virtuel.
4. Entrez un nom unique pour le modèle de service virtuel dans le champ File Name (Nom du fichier). L'extension est **.vsm**.
5. Cliquez sur Save (Enregistrer).

Le nouveau modèle de service virtuel s'ouvre.

Lorsqu'un modèle de service virtuel est ouvert, le menu Commands (Commandes) affiche les éléments de menu pertinents pour un modèle de service virtuel.

Ouverture d'un modèle de service virtuel

Procédez comme suit:

1. Dans la barre d'outils, cliquez sur Open (Ouvrir), ou sélectionnez un projet dans la liste de projets Open Recent (Ouvrir un document récent) de la fenêtre Quick Start (Démarrage rapide).
2. Sélectionnez le modèle de service virtuel à partir de son dossier dans le panneau Project (Projet).

En règle générale, les modèles de service virtuel résident dans le dossier VServices ou VirtualServices.

Utilisation de protocoles de données

Un *protocole de données* est également appelé *gestionnaire de données* ou *gestionnaire de protocoles de données*. Un protocole de données est responsable 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. En conséquence, le protocole de données doit connaître la charge utile de la demande. Un protocole de transport peut comprendre plusieurs protocoles de données. Certains protocoles de transport (par exemple, JDBC) n'autorisent aucun protocole de données.

Pendant l'enregistrement, tous les protocoles de transport tentent de détecter la charge utile de la demande ou de la réponse et les protocoles de données appropriés par défaut côté demande ou réponse. Les gestionnaires de protocole de données indiqués sont ajoutés aux charges utiles suivantes :

- **SOAP** : gestionnaire de protocole de données (SOAP) de services Web
- **XML** : gestionnaire de protocole de données XML
- **HTML** : aucun gestionnaire de protocole de données
- **Signed SOAP (SOAP signé)** : un gestionnaire de protocole de données de demande WS-Security côté demande et de réponse WS-Security côté réponse
- **Encrypted SOAP (SOAP chiffré)** : un gestionnaire de protocole de données de demande WS-Security côté demande et de réponse WS-Security côté réponse
- **SOAP Security (Sécurité SOAP)** : un gestionnaire de protocole de données de demande WS-Security côté demande et de réponse WS-Security côté réponse

Vous pouvez ajouter d'autres protocoles de données, réorganiser ceux définis par défaut, ou supprimer les valeurs par défaut. Si vous créez une image de service à partir de paires demande/réponse, du fichier de trafic brut ou via une méthode différente de l'enregistrement et que vous avez spécifié des gestionnaires de protocole de données, les valeurs par défaut ne seront pas ajoutées.

Vous pouvez placer ces gestionnaires de protocole de données sous forme de chaîne pour les utiliser ensemble. Par exemple, côté demande, il est possible d'utiliser simultanément le protocole de données de texte délimité, l'analyseur de charge utile XML générique et le gestionnaire de données de demande.

Chaque protocole de données disponible est décrit dans les sections suivantes.

[Détection des transactions de hachage automatique](#) (page 230)
[Copybook CICS](#) (page 231)
[Copybook](#) (page 234)
[Copybook CTG](#) (page 251)
[Désensibilisation des données](#) (page 253)
[Delimited Text \(Texte délimité\)](#) (page 254)
[DRDA](#) (page 257)
[Protocole de données EDI X12](#) (page 258)
[Protocole de données d'analyseur de charge utile XML générique](#) (page 262)
[Protocole de données JSON](#) (page 267)
[Copieur de données de la demande](#) (page 268)
[Protocole de données Request Data Manager \(Gestionnaire de données de demande\)](#) (page 269)
[Protocole de données REST](#) (page 273)
[Protocole de données prenant en charge les scripts](#) (page 281)
[Protocole de données SWIFT](#) (page 284)
[Protocole de données Web Services Bridge](#) (page 286)
[Web Services \(SOAP\) \(Services Web \(SOAP\)\)](#) (page 287)
[Web Services \(SOAP Headers\) \(Services Web \(en-têtes SOAP\)\)](#) (page 288)
[Protocole de données WS-Security Request \(Demande WS-Security\)](#) (page 290)
[XML](#) (page 294)

Détection des transactions de hachage automatique

Le protocole de données Auto Hash Transaction Discovery (Détection des transactions de hachage automatique) identifie un message par le code de hachage des données. Même une légère modification des données entraîne un changement du code de hachage, ce qui rend toutes les demandes uniques. Ce protocole de données est utile lorsque vous exécutez le même petit ensemble de demandes au niveau du service.

L'objectif du protocole de données Auto Hash Transaction Discovery (Détection des transaction de hachage automatique) est simple. A mesure qu'il reçoit des demandes à gérer, la fonction de code de hachage Java standard est appliquée à la version texte du corps des demandes. La valeur du code de hachage obtenue est ensuite ajoutée comme nouvel argument dans la demande nommée **lisa.vse.auto.hashDiscovery**.

Cette fonction peut être utile, notamment lors de la virtualisation de la messagerie, pour créer une valeur raisonnable unique à utiliser pour identifier des conversations.

Ce protocole ne requiert aucune information de configuration et ne présente donc aucune fenêtre dans l'assistant d'enregistrement.

Copybook CICS

Le protocole de données CICS Copybook divise la demande enregistrée en fragments de conteneur égaux. Chaque fragment est ensuite envoyé au protocole de données Copybook et groupé avec le XML correspondant. Le XML obtenu est le XML cumulé des conteneurs.

Cet exemple illustre l'utilisation du protocole de données CICS Copybook.

Procédez comme suit:

1. Dans l'onglet [Basics \(Paramètres de base\)](#) (page 131) de l'enregistreur d'image de service virtuel, remplissez les champs obligatoires.

2. Cliquez sur Next (Suivant).

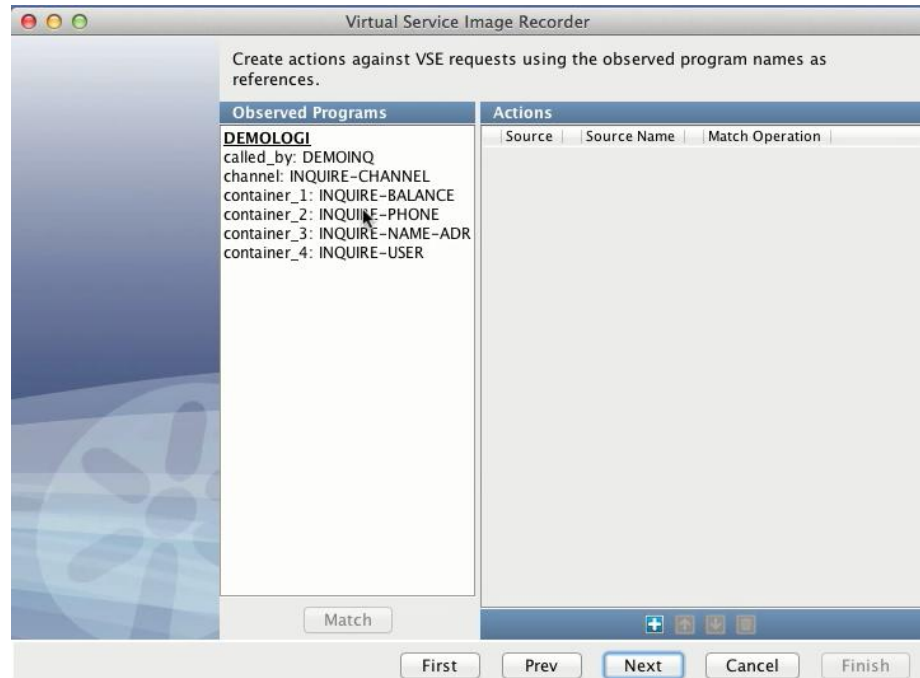
3. Entrez une adresse IP et un port, puis cliquez sur Next (Suivant).

La fenêtre de sélection Data Protocol (Protocole de données) s'ouvre.

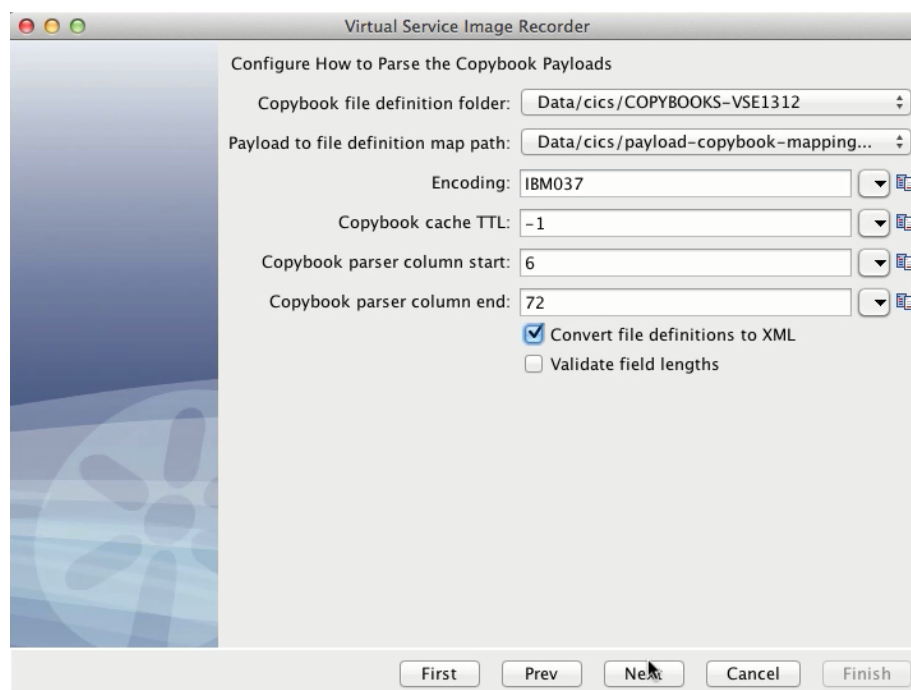
4. Sélectionnez le protocole de données CICS Request Data Access (Accès aux données des demandes CICS).

5. Sélectionnez le protocole de données CICS Container Copybook (Copybook de conteneur CICS) pour les côtés demande et réponse, puis cliquez sur Next (Suivant).

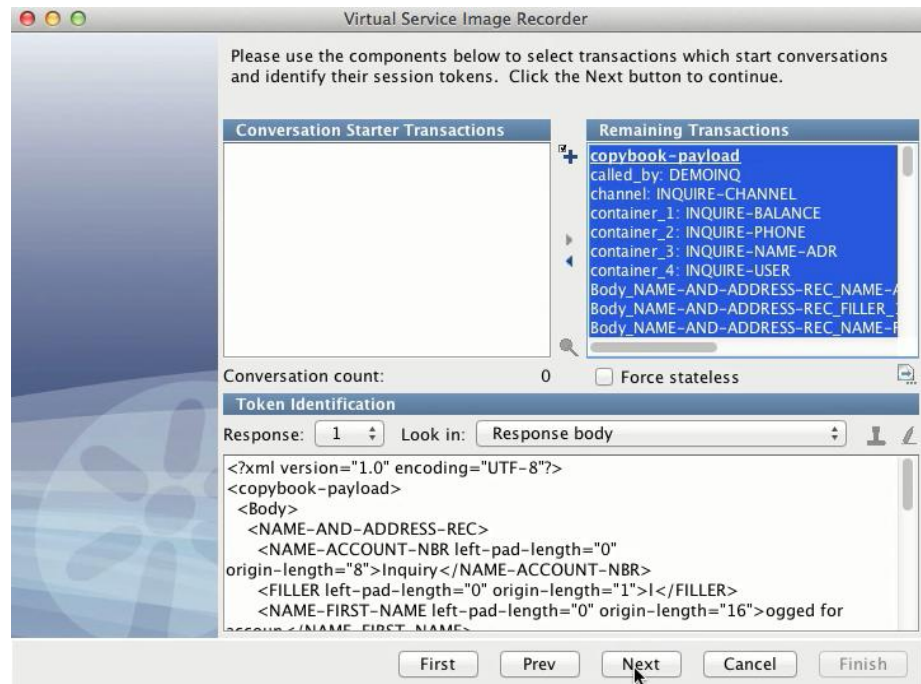
La liste Observed Programs (Programmes constatés) contient les quatre conteneurs CICS.



6. Cliquez sur Next (Suivant).

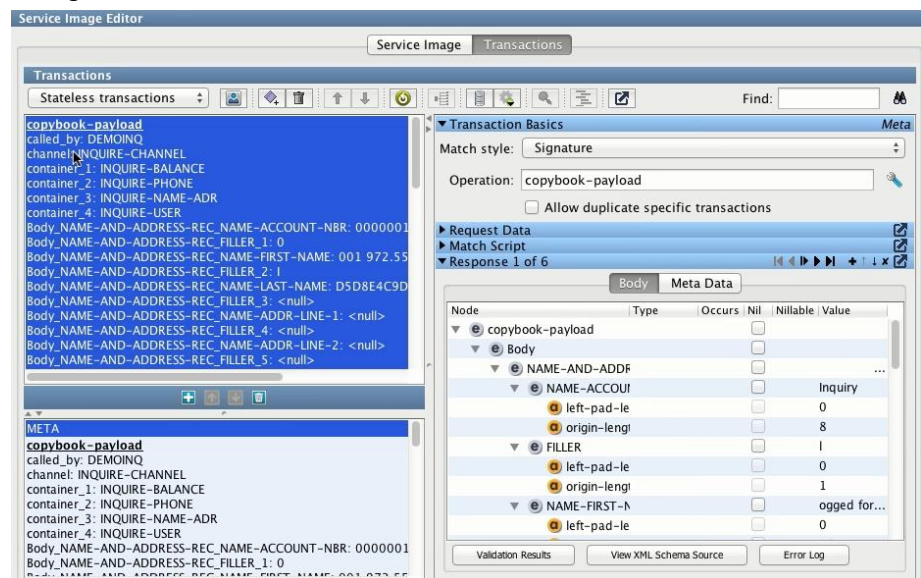


7. Entrez les paramètres suivants dans les définitions côté demande :
 - Dossier des définitions de fichier copybook
 - Chemin d'accès au mappage de charges utiles vers les définitions de fichier
 - Informations de codage
8. Sélectionnez Convert file definitions to XML (Convertir des définitions de fichier en XML) et cliquez sur Next (Suivant).



9. Pour remplir l'image de service, cliquez sur Next (Suivant).
10. Pour afficher l'image de service, sélectionnez View Service Image (Afficher une image de service) et cliquez sur Finish (Terminer).

L'image de service terminée s'ouvre.



Copybook

Le protocole de données Copybook permet au VSE de convertir des charges utiles de copybook en XML et inversement, lors de l'exécution et de manière transparente pour l'appelant. L'affichage de ces charges utiles au format XML permet à d'autres mécanismes de VSE standard de fournir des valeurs. Vous pouvez utiliser l'analyseur de charge utile XML générique pour identifier et sélectionner des arguments de demande et des opérations de correspondance, ainsi qu'organiser des transactions dans l'image de service. L'utilisation d'une structure XML standard permet de traiter des chaînes magiques dans les réponses, de sorte à pouvoir créer des données dynamiques de manière plus utile.

Cette section comprend les rubriques suivantes :

- [Termes](#) (page 235)
- [Installation](#) (page 235)
- [Utilisation](#) (page 236)
- [Fichier de mappage de charges utiles](#) (page 239)
- [Logique de correspondance](#) (page 247)

Termes

Afin de mieux comprendre le protocole de données Copybook, il est important de comprendre les termes suivants :

- **Copybook** : structure hiérarchique utilisée pour définir la disposition des données. Le copybook identifie les noms des champs, leur longueur et leur relation.
- **Copybook File Definition (Définition de fichier de copybook)** : fichier au format de texte brut contenant un copybook
- **Field (Champ)** : élément unique dans un copybook. Un champ contient un nom, la longueur et le type de données d'une donnée logique unique dans un enregistrement.
- **Payload (Charge utile)** : collection d'un ou plusieurs enregistrements que le VSE affiche dans une demande unique ou une réponse unique.
- **Payload Mapping File (Fichier de mappage de charges utiles)**, également appelé Payload Copybook Mapping (Mappage de copybook de charges utiles) ou Payload to File Definition Map (Mappage de charges utiles vers les définitions de fichier) : document XML décrivant la méthode d'identification du copybook ou des copybooks requis pour l'analyse d'une charge utile. Le fichier de mappage de charges utiles fournit la structure de base au XML résultant.
- **Record (Enregistrement)** : collection de données sans élément structural. Un ou plusieurs champs formant un copybook unique constituent un enregistrement. Un enregistrement peut ne pas contenir des données pour chaque champ défini par un copybook, mais la relation entre les enregistrements et les copybooks est toujours unique. Un enregistrement donné est défini uniquement par un seul copybook et un copybook unique définit un seul enregistrement.

Conditions préalables à la création d'un service virtuel de copybook

Pour préparer la création d'un service virtuel de copybook, effectuez les opérations suivantes :

- Rassemblez tous les copybooks qui définissent les enregistrements de vos charges utiles. Ces copybooks doivent être placés dans un répertoire de votre projet DevTest. Vous pouvez inclure la hiérarchie que vous jugez utile dans le répertoire que vous utilisez. Généralement, le dossier de définitions de fichier de copybook est placé à un endroit sous le dossier Data (Données) du projet.
- Créez un [fichier de mappage de charges utiles](#) (page 239).
- Préparez votre client pour l'enregistrement, rassemblez les paires demande/réponse, capturez un fichier PCAP du trafic, ou utilisez une autre méthode pour collecter les transactions à utiliser pour votre service virtuel.

Procédure d'utilisation du protocole de données de copybook

Vous pouvez utiliser le protocole de données de copybook avec l'un des protocoles de transport pris en charge par le VSE. En règle générale, le gestionnaire de protocole de données Copybook est utilisé avec un protocole de messagerie comme JMS ou MQ, ou avec le protocole de transport CICS. Un exemple d'application de copybook utilisant le protocole HTTP est disponible dans le serveur de démonstration.

Sélection

Lors de la sélection du protocole de données de Copybook, tenez compte des points suivants :

- Envisagez d'utiliser le gestionnaire de protocole de données côtés demande et réponse. Bien qu'il soit possible d'utiliser le protocole de données sur un seul côté, cela n'est pas habituel. Cette documentation ne couvre pas l'utilisation sur un côté de manière détaillée.
- Le protocole de données de Copybook convertit le corps de la demande (ou de la réponse) en XML. Le protocole de données de copybook n'ajoute pas automatiquement d'arguments à chaque champ de la demande. Pour utiliser à bon escient le protocole de données de copybook, vous devez inclure un autre gestionnaire de protocole de données côté demande. En règle générale, l'analyseur de charge XML générique est le deuxième gestionnaire de protocole de données. Toutefois, un gestionnaire de protocole de données qui effectue des opérations efficaces avec des charges utiles XML est acceptable.

Configuration

Après avoir capturé le trafic, ou importé des paires demande/réponse, un fichier PCAP ou de trafic brut, la fenêtre Copybook DPH Configuration (Configuration du gestionnaire de protocoles de données Copybook) suivante s'ouvre.

Entrez les paramètres suivants :

Copybook file definition folder (Dossier des définitions de fichier de copybook)

Indique le dossier de stockage de vos définitions de fichier de copybook dans votre projet. Ce dossier devient le chemin de base pour les chemins d'accès relatifs dans le fichier de mappage des charges utiles.

Payload to file definition map path (Chemin d'accès au mappage de charges utiles vers les définitions de fichier)

Il s'agit du document XML qui sert de fichier de mappage des charges utiles pour ce service virtuel.

Encoding (Codage)

Permet de spécifier un [jeu de caractères Java](#) valide. Si une valeur est indiquée, elle sera utilisée pour tenter de convertir les octets de la charge utile au format texte à utiliser dans la sortie XML. La valeur du jeu de caractères par défaut de VSE est UTF-8. Vous pouvez configurer le jeu de caractères par défaut en définissant `lisa.vse.default.charset` dans `local.properties`.

Copybook cache TTL (Durée de vie du cache du copybook)

Lors de l'exécution, VSE doit référencer un copybook spécifique et doit pour cela lire à partir du fichier et éventuellement le convertir en représentation XML du copybook. Ce paramètre définit la durée de conservation dans la mémoire d'une version mise en cache du copybook converti. Une fois la TTL atteinte, le copybook converti est supprimé du cache. Si ce fichier est de nouveau requis, il sera de nouveau lu et reconverti. Une valeur de TTL de 0 ou un nombre négatif indique que la mise en cache est désactivée. Les fichiers sont lus et analysés chaque fois qu'ils sont requis. Si la valeur de la TTL est un nombre positif, elle sera utilisée comme délai d'expiration (en secondes).

Copybook parser column start (Début de colonne de l'analyseur de copybook)

Les copybooks contiennent souvent des numéros de ligne au début de chaque ligne. Généralement, ces numéros de ligne sont composés de six chiffres ; la valeur par défaut de ce paramètre est donc 6. Cette option permet de spécifier la colonne au niveau de laquelle l'analyse de la définition de fichier de copybook doit démarrer. Le numéro correspond à un index démarrant à partir de zéro et est inclusif. Toutefois, vous pouvez le considérer comme un index ordinaire exclusif démarrant à partir de un. Si vous définissez cette valeur sur 6, les six premiers caractères seront ignorés et l'index commencera par le septième.

Copybook parser column end (Fin de colonne de l'analyseur de copybook)

Les copybooks contiennent parfois d'autres données de référence à la fin de chaque ligne. Lorsque c'est le cas, l'analyseur doit connaître la colonne au niveau de laquelle il doit s'arrêter. S'il n'existe aucune donnée supplémentaire à la fin des lignes du fichier, vous pouvez définir ce numéro sur une valeur supérieure à la longueur de la ligne la plus longue dans le fichier. Le numéro correspond à un index démarrant à partir de zéro et est exclusif. Toutefois, vous pouvez le considérer comme un index ordinaire inclusif démarrant à partir de un. Si vous définissez cette valeur sur 72, le 72e caractère dans la ligne sera lu, puis l'analyseur s'arrêtera (sans tenter de lire le 73e). Si ce numéro est supérieur à la longueur d'une ligne, l'analyseur s'arrêtera à la fin de la ligne.

Validate field lengths (Valider les longueurs de champ)

Dans le VSE, cette option est uniquement utilisée côté réponse. Lors de l'enregistrement, la charge utile est convertie au format XML, puis de nouveau en octets, afin d'assurer la conversion symétrique. De même, pendant la lecture, les réponses XML sont de nouveau converties en enregistrements/charges utiles avant leur envoi à l'appelant. Pour ces deux opérations, VSE peut valider l'exactitude de la valeur dans chaque champ avec la longueur spécifiée dans le copybook. Toutefois, cette validation n'est pas toujours souhaitable. Par exemple, si certains champs de votre enregistrement ne contiennent aucune donnée, leur longueur est considérée égale à 0, tandis que le copybook définit ces champs sur une longueur supérieure à 0. Dans ce cas, si cette option est sélectionnée, la validation par VSE échouera et une erreur sera signalée. Si vous savez que vos données ne s'alignent pas sur la longueur définie dans vos copybooks, laissez cette case à cocher désélectionnée. Toutefois, si vous voulez assurer que chaque enregistrement contienne exactement les données qu'il doit contenir, vous pouvez sélectionner cette option.

XML elements as request arguments (Éléments XML comme arguments de la demande)

Indique de vérifier si les demandes et les réponses sont des chaînes XML.

Valeurs :

- Si cette option est **sélectionnée**, elle identifie les variables des messages XML utilisés par l'enregistreur. Elle permet de vérifier que les demandes et les réponses sont des chaînes XML.
- Si elle est **désélectionnée**, elle ne vérifiera pas si les demandes et les réponses sont des chaînes XML.

Permet de vérifier que les demandes et les réponses sont des chaînes XML. Sélectionnez cette option pour identifier les variables dans les messages XML utilisés par l'enregistreur. Pour plus de détails sur l'identification des variables, reportez-vous à la section [Analyseur de charge utile XML générique](#) (page 262).

Ce même éditeur est disponible dans les filtres de protocole de données du modèle de service virtuel (un côté demande et un côté réponse). Il permet de changer la configuration après l'enregistrement, le cas échéant.

Fichier de mappage de charges utiles

Le *fichier de mappage de charges utiles* est un document XML qui décrit la mise en correspondance des charges utiles entrantes avec leur copybook correspondant. Le fichier de mappages de charges utiles est également appelé *payload copybook mapping* (mappage de copybook de charge utile) et *payload to copybook file definition map* (mappage de charges utiles vers les définitions de fichier de copybook). Ce fichier fournit également des conseils sur la structure du XML obtenu à des fins de clarification pour l'utilisateur.

Exemple

Un exemple de fichier de mappage est disponible [ici](#). Ce fichier contient des exemples de différentes configurations et de commentaires, décrivant chacun des attributs sur chaque noeud. Cet exemple de fichier est fourni à titre de référence pour la création de vos fichiers de mappage de copybook.

Structure

L'exemple suivant illustre la structure du fichier de mappage de charges utiles de base. Par souci de simplicité, cet exemple ne contient aucun attribut de valeurs.

```
<payloads>
  <payload>
    <key></key>
    <section>
      <copybook></copybook>
      ...
    </section>
    ...
  </payload>
  <payload>
    ...
  </payload>
</payloads>
```

<payloads>

Noeud XML racine du document. Vous ne pouvez pas répéter ce noeud.

<payload>

L'ensemble de cet élément décrit de manière complète une charge utile correspondante. Lorsqu'une charge utile est déterminée comme correspondant à un élément <payload> dans ce document, l'élément <payload> est utilisé pour décrire la charge utile entrante.

<key>

(Facultatif) Cette option rend le document XML plus lisible. Tous les détails que vous spécifiez dans cet élément peuvent également être spécifiés comme attributs dans l'élément <payload>.

<section>

Groupement logique de copybooks qui définit une partie de la charge utile. S'il existe plusieurs éléments <section>, ils seront traités dans l'ordre sans répétition.

<copybook>

Copybook unique qui décrit ou non un enregistrement dans la charge utile. Un élément <section> peut contenir plusieurs éléments <copybook>.

<payload>

Un élément <payload> peut contenir les attributs suivants :

```
<payload name="TEST" type="request" matchType="all" key="reqKey"
value="reqVal" keyStart="3" keyEnd="6" headerBytes="0"
footerBytes="0" saveHeaderFooter="false" definesResponse="false">
  ...
</payload>
```

Attribute (Attribut)	Requis/facultatif	Description
name	Obligatoire	Définit un nom unique permettant d'identifier le type de demande décrite ici. Name doit être unique parmi l'ensemble des charges utiles du même type (à savoir, demande ou réponse).
type	Obligatoire	Spécifie le type de charge utile. Valeur - il s'agira de l'une des options suivantes : <ul style="list-style-type: none">■ requête■ réponse

matchType	Facultatif	<p>L'attribut matchType s'applique différemment aux définitions de charge utile de type réponse. Par exemple, les réponses ne contiennent pas d'arguments ni de noms d'opération. Si une définition de charge utile de type réponse définit cet attribut sur argument, attribute ou operation, elle ne correspondra jamais à aucun élément (sauf si la demande correspondant définit definesResponse sur true, qui remplace cet attribut). Si une définition de charge utile de type réponse définit cet attribut sur all (tout), les phases argument, attribute et operation de correspondance seront ignorées.</p> <p>Valeur - il s'agira de l'une des options suivantes :</p> <ul style="list-style-type: none"> ■ argument : tentative de correspondance avec l'argument spécifié uniquement ■ attribute : tentative de correspondance avec l'attribut spécifié uniquement ■ metaData : tentative de correspondance avec les métadonnées spécifiées uniquement ■ operation : tentative de correspondance avec le nom d'opération uniquement ■ payload : tentative de correspondance avec le corps de la demande ■ all : tentative de correspondance dans l'ordre suivant : argument, attribute, metaData, operation et payload <p>Valeur par défaut : payload (charge utile)</p>
clé	Obligatoire	<p>Le comportement de cet attribut change selon le matchType (Type de correspondance) :</p> <ul style="list-style-type: none"> ■ Si matchType (Type de correspondance) est operation ou payload (charge utile), la valeur de cet attribut sera utilisée pour la correspondance. ■ Si matchType (Type de correspondance) est argument, attribute, metaData (Métadonnées), ou all (tout), la valeur de cet attribut est considérée comme la clé d'une entrée argument, attribute, ou metaData (Métadonnées). Si la valeur de matchType est all (tout), la valeur key (clé) ne sera pas utilisée pour correspondre avec operationName (nom de l'opération) ou le corps de charge utile. L'attribut value (valeur) sera utilisé à la place.

valeur	Optional/Requiert	<p>matchType définit le comportement de cet attribut et indique s'il est requis ou non :</p> <ul style="list-style-type: none"> ■ Si matchType est défini sur operation ou payload (charge utile), la valeur de cet attribut sera ignorée et peut être exclue. ■ Si matchType est défini sur argument, attribute, ou metaData, cet attribut sera considéré comme la valeur d'une entrée argument, attribute, ou metaData et sera requis pour la correspondance. ■ Si matchType est all (tout), cet attribut sera requis et son comportement sera identique à celui décrit pour les phases de correspondance argument, attribute et metaData. Toutefois, pendant les phases de correspondance operation et payload (charge utile), la valeur de cet attribut sera utilisée pour la correspondance (par opposition à l'utilisation de la valeur de l'attribut key comme c'est le cas lorsque matchType est défini sur operation ou payload).
keyStart	Facultatif	<p>Désigne la position de démarrage de la recherche de la clé dans la charge utile (index démarrant à partir de 1). La recherche inclut cette valeur.</p> <ul style="list-style-type: none"> ■ Si vous définissez keyStart sur 1, la recherche commencera par le premier octet. ■ Si vous ne définissez pas keyStart, la recherche sera effectuée dans la charge utile complète et keyEnd sera ignorée. ■ Si KeyStart est défini sur un nombre inférieur à 1, la valeur sera traitée comme 1. ■ Si KeyStart est défini sur un nombre supérieur à la longueur de la charge utile, la recherche sera effectuée dans la charge utile complète et la valeur de keyEnd sera ignorée. ■ Si KeyStart est défini sur une valeur égale ou supérieure à keyEnd, la recherche sera effectuée dans la charge utile complète. ■ Si le résultat de keyEnd - keyStart est inférieur à la longueur de la clé, la recherche sera effectuée dans la charge utile complète.

keyEnd	Facultatif	<p>Désigne la position de fin de la recherche de la clé dans la charge utile (index démarrant à partir de 1). La recherche exclut cette valeur.</p> <ul style="list-style-type: none"> ■ Si vous définissez keyEnd sur 3, seuls les octets 1 et 2 seront recherchés. ■ Si cette valeur n'est pas fournie, la recherche commencera à partir de la valeur de keyStart et se terminera à la fin de la charge utile. ■ Si cette valeur est supérieure à la longueur de la charge utile, la recherche se terminera à la fin de la charge utile. ■ Si cette valeur est égale ou inférieure à keyStart la recherche sera effectuée dans la charge utile complète. ■ Si le résultat de keyEnd - keyStart est inférieur à la longueur de la clé, la recherche sera effectuée dans la charge utile complète.
headerBytes	Facultatif	<p>Désigne le nombre d'octets à supprimer à partir du début de la charge utile. Par défaut, cette valeur est définie sur 0, si vous ne fournissez aucune valeur. Si l'attribut est présent, sa valeur doit être un nombre entier valide.</p>
footerBytes	Facultatif	<p>Désigne le nombre d'octets à supprimer à partir de la fin de la charge utile. Par défaut, cette valeur est définie sur 0, si vous ne fournissez aucune valeur. Si l'attribut est présent, sa valeur doit être un nombre entier valide.</p>
saveHeaderFooter	Facultatif	<p>Spécifie si les octets de l'en-tête et du pied de page supprimés seront conservés dans la version XML de la demande sous forme de chaînes hexadécimales sous les balises rawHeader et rawFooter.</p> <p>Valeurs :</p> <ul style="list-style-type: none"> ■ true : conserve les octets de l'en-tête et du pied de page supprimés. ■ false : ne conserve pas les octets de l'en-tête et du pied de page supprimés. <p>Valeur par défaut : false</p>
definesResponse	Facultatif	<p>Si cet attribut est défini sur true, la réponse pour cette demande recherchera un élément de charge utile avec un nom identique, mais de type response. Cet attribut sera ignoré si le type est response.</p> <p>Valeur par défaut : false</p>

Vous pouvez utiliser l'élément <key> facultatif pour remplacer les attributs liés aux clés. Pour améliorer la lisibilité, l'exemple de l'élément <payload> précédemment décrit peut également être écrit de la manière suivante :

```
<payload name="TEST" type="request" headerBytes="0" footerBytes="0"
saveHeaderFooter="false" definesResponse="false">
  <key matchType="all" value="reqVal" keyStart="3"
    keyEnd="6">reqKey</key>
  ...
</payload>
```

<section>

Exemple : Voici un exemple d'élément de section, avec tous les attributs qu'il contient ainsi que leur description.

```
...
<section name="Body">
  ...
</section>
...
```

Attribute (Attribut)	Obligatoire	Description
name	Obligatoire	Définit le nom d'un élément XML de groupement dans la version de sortie XML de la charge utile. Un ou plusieurs éléments de copybook convertis figureront dessous.

<copybook>

Exemple : Voici un exemple d'élément de copybook, avec tous les attributs qu'il contient ainsi que leur description.

```
...
<copybook key="cpk" order="1" max="1" name="TESTRECORD"
length-field="SOME-ID">TESTIN.CPY.TXT</copybook>
...
```

Attribute (Attribut)	Obligatoire	Description
clé	Obligatoire	Définit la chaîne unique dans l'enregistrement qui identifie le copybook. Techniquement, cet attribut est facultatif. Toutefois, si une clé n'est pas fournie, cela signifie que ce copybook sera le seul attribut qui pourra être appliqué à la charge utile. Il sera appliqué en continu jusqu'à ce que la charge utile ne contienne plus d'octets. Si plusieurs éléments de copybook ne disposent d'aucune clé, le premier sera toujours utilisé, sauf si l'attribut max est spécifié.
order	Facultatif	Définit un indice sur l'ordre de recherche des enregistrements dans la charge utile. Les numéros utilisés sont inappropriés, mais d'autres enregistrements ultérieurs dans la charge utile utiliseront un nombre entier supérieur. Vous pouvez baliser plusieurs copybooks avec le même numéro d'ordre, pour indiquer que ces enregistrements peuvent être présentés dans un ordre quelconque. Lorsqu'un enregistrement est détecté avec un numéro d'ordre donné, les recherches ultérieures concerneront uniquement des copybooks portant ce numéro d'ordre et un numéro supérieur. Il est possible d'inclure des copybooks dans un groupe qui ne correspondra jamais à la charge utile. Ils seront simplement ignorés. Toutefois, cela affecte les performances, car chaque copybook doit être vérifié. Valeur par défaut : 0
max	Facultatif	Définit le nombre d'applications d'un copybook à la charge utile. Les valeurs vides, les nombres négatifs, les valeurs autres qu'un nombre ou inexistantes et 0 seront considérés comme aucune limite.

name	Facultatif	<p>Définit une valeur pour remplacer le nom d'un enregistrement (c'est-à-dire, le niveau racine dans le copybook).</p> <ul style="list-style-type: none">■ Si cette valeur est définie, le noeud généré dans le XML pour ce copybook utilisera ce nom au lieu du nom d'enregistrement dans la définition du copybook.■ Si vous ne définissez pas cette valeur, par défaut le nom d'enregistrement sera recherché dans la définition du copybook et utilisé.■ Si vous définissez cette valeur sur le nom de l'enregistrement dans la définition de copybook, seule la lisibilité de ce fichier sera affectée.
length-field	Facultatif	<p>Définit la méthode de fractionnement de la charge utile de sorte que la recherche de l'enregistrement suivant commence à l'endroit correct.</p> <p>Si cet attribut n'est pas présent, le processeur tente de déterminer la longueur du copybook à partir de la définition. Si, pour une quelconque raison, la longueur ne peut pas être déterminée, le processeur considérera que le reste de la charge utile s'applique à ce copybook et mettra fin au traitement après l'application de ce copybook. S'il ne s'agit pas d'un champ numérique d'affichage non signé, le processeur l'ignorera.</p>

Logique de correspondance

Vous pouvez reconstituer le fonctionnement de la logique de correspondance en fonction des descriptions des éléments XML et des arguments. Toutefois, pour plus de clarté, cette section décrira la procédure de correspondance dans son intégralité.

Lorsque le VSE reçoit une charge utile, la mise en correspondance se produit dans les phases suivantes :

- Charge utile
- Section
- Copybook
- Finalisation

Charge utile

La charge utile est la première phase de correspondance qui a lieu lorsque le VSE reçoit une nouvelle charge utile. Cette phase détermine l'élément <payload> dans le fichier de mappage de charge utile correspondant à la charge utile reçue. VSE tente de mettre en correspondance un élément <payload> en traitant les éléments dans le fichier de mappage de charge utile dans l'ordre, à savoir de haut en bas. La première correspondance détectée sera sélectionnée.

Remarque : Si aucun élément de charge utile ne correspond à la charge utile, elle sera traitée comme charge utile inconnue. Le contenu sera hexadécimal et encapsulé dans une structure XML générique. Si nécessaire, les charges utiles inconnues seront automatiquement reconverties en octets lors de la lecture.

Pour les charges utiles de demande :

1. Si l'attribut type de cet élément est request (demande), continuez. Dans le cas contraire, cet élément ne correspond pas.
2. Si matchType est défini sur argument, recherchez un argument de demande dont la clé correspond à l'attribut key et la valeur correspond à l'attribut value dans cet élément. En cas de détection, cet élément de charge utile correspondra.
3. Si matchType est défini sur attribute, recherchez un attribut de demande dont la clé correspond à l'attribut key et la valeur correspond à la valeur attribute dans cet élément. En cas de détection, cet élément de charge utile correspondra.
4. Si matchType est défini sur metaData (Métadonnées), recherchez une entrée de métadonnées de demande dont la clé correspond à l'attribut key et la valeur correspond à l'attribut value dans cet élément. En cas de détection, cet élément de charge utile correspondra.

5. Si `matchType` est défini sur `operation`, vérifiez si le nom d'opération de la demande correspond à l'attribut `key` dans cet élément. Si c'est le cas, cet élément de charge utile correspondra.
6. Si `matchType` est défini sur `payload`, recherchez dans les limites spécifiées par cet élément les attributs `keyStart` et `keyEnd` pour la valeur dans l'attribut `key`. Si la clé est détectée dans ces limites, l'élément de charge utile correspondra.
7. Si `matchType` est défini sur `all`, les étapes 2 à 6 seront traitées dans l'ordre et le traitement s'arrêtera lorsqu'une correspondance sera détectée. La seule différence repose sur le fait que dans les étapes 5 et 6, l'attribut `value` est utilisé à la place de l'attribut `key`.

Pour les charges utiles de réponse (lors de l'enregistrement) :

1. Si l'élément de charge utile de demande précédemment décrit a défini l'attribut `definesResponse` sur **true**, renvoyez immédiatement l'élément de charge utile avec le même nom que l'élément de demande, mais avec le type de réponse. Si aucun élément de ce type n'est détecté, il n'y aura aucune correspondance.
2. Si l'élément de charge utile de demande précédemment décrit n'a *pas* défini l'attribut `definesResponse` sur `true` :
 - a. Si l'attribut `type` de cet élément est `response` (réponse), continuez. Dans le cas contraire, cet élément ne correspond pas.
 - b. Si `matchType` est défini sur `metaData` (Métadonnées), recherchez une entrée de métadonnées de réponse dont la clé correspond à l'attribut `key` et la valeur correspond à l'attribut `value` dans cet élément. En cas de détection, cet élément de charge utile correspondra.
 - c. Si `matchType` est défini sur `payload`, recherchez dans les limites spécifiées par cet élément les attributs `keyStart` et `keyEnd` pour la valeur dans l'attribut `key`. Si la clé est détectée dans ces limites, l'élément de charge utile correspondra.
 - d. Si `matchType` est défini sur `all`, effectuez les étapes b et c dans l'ordre et arrêtez le processus lorsqu'une correspondance est détectée. La seule différence repose sur le fait que dans l'étape c, l'attribut `value` est utilisé à la place de l'attribut `key`.

Pour les charges utiles de réponse (lors de la lecture) :

Lors de la lecture, aucune correspondance n'est effectuée pour les réponses. En revanche, lors de l'enregistrement, toute correspondance déterminée est enregistrée dans les métadonnées de l'objet de réponse. Puis, pendant la lecture, lorsque cet objet de réponse est renvoyé, le processeur récupère la valeur à partir des métadonnées de la réponse, recherche l'élément de charge utile avec ce nom (et le type `response`) et l'utilise. Si aucun élément de ce type n'existe, cela est considéré comme une erreur irrécupérable.

Section

Une fois l'élément de charge utile identifié, le processeur examine chaque élément de section dans l'ordre, c'est-à-dire de haut en bas. Aucune correspondance n'est établie à ce niveau. A l'issue du traitement complet d'une section par le processeur (en d'autres termes, aucun des copybooks de la section ne correspond), il passe à la section suivante sans réexaminer les sections précédentes.

Copybook

La dernière phase de correspondance consiste à déterminer le copybook dans cette section qui s'applique en premier, en deuxième, etc., jusqu'à ce que tous les enregistrements dans la charge utile soient traités. Cette phase de correspondance est la plus complexe. Le processus est le suivant :

1. Le processeur examine tous les éléments copybook dans la section et les trie dans l'ordre des numéros spécifiés dans leurs attributs d'ordre.
2. La liste d'éléments copybook avec le numéro d'ordre le plus bas est d'abord vérifiée (dans l'ordre spécifié dans le fichier). La valeur de l'attribut key dans cet élément copybook est recherchée dans la charge utile restante et l'index dans lequel elle a été détectée (le cas échéant) est enregistré. La valeur de l'attribut clé dans cet élément copybook est recherchée dans la charge utile restante et l'index où elle a été détectée (le cas échéant) est enregistré.
3. Une fois que tous les copybooks dans la liste d'ordre le plus bas ont été vérifiés, en cas de correspondance, le plus ancien de la charge utile sera utilisé.
4. En cas de correspondance d'un copybook :
 - a. Si une correspondance a déjà été établie pour ce copybook dans cette charge utile (un enregistrement antérieur dans la charge utile), le processeur vérifiera si l'attribut max peut être à nouveau appliqué. Si l'attribut max a été atteint, la correspondance ne sera pas traitée comme telle et la copybook correspondant suivant sera utilisé.
 - b. Si l'attribut max n'a pas été atteint, ce copybook sera appliqué à la charge utile et le nombre d'octets spécifiés par le copybook sera consommé. Si un attribut length-field est présent dans le copybook, la valeur de ce champ dans l'enregistrement sera utilisée pour déterminer le nombre d'octets à consommer et le processeur reviendra à l'étape 2.
5. Si aucune correspondance avec un copybook n'est établie, la liste de copybooks avec le numéro d'ordre le plus bas suivant sera vérifiée à l'aide des mêmes règles que celles des étapes 2, 3 et 4. Une fois que le processeur détermine qu'aucun copybook dans une liste de numéros d'ordre donnée ne correspond, il n'effectue aucune nouvelle tentative de traitement de cette liste de numéros d'ordre (pour cette section) pour cette charge utile.
6. Après la vérification de toutes les listes et si aucune correspondance n'est détectée (ou que la charge utile ne contient plus d'octets), le processeur passe à la section suivante.

Finalisation

S'il reste des octets dans la charge utile après le traitement de toutes les sections d'une charge utile, ils seront tronqués.

Copybook CTG

Le protocole de données CTG Copybook divise la demande enregistrée en fragments de conteneur égaux. Chaque fragment est ensuite envoyé au protocole de données Copybook et groupé avec le XML correspondant. Le XML obtenu est le XML cumulé des conteneurs.

L'exemple suivant illustre l'utilisation du protocole de données CTG Copybook.

Procédez comme suit:

1. Dans l'onglet [Basics \(Paramètres de base\)](#) (page 131) de l'enregistreur d'image de service virtuel, remplissez les champs obligatoires.

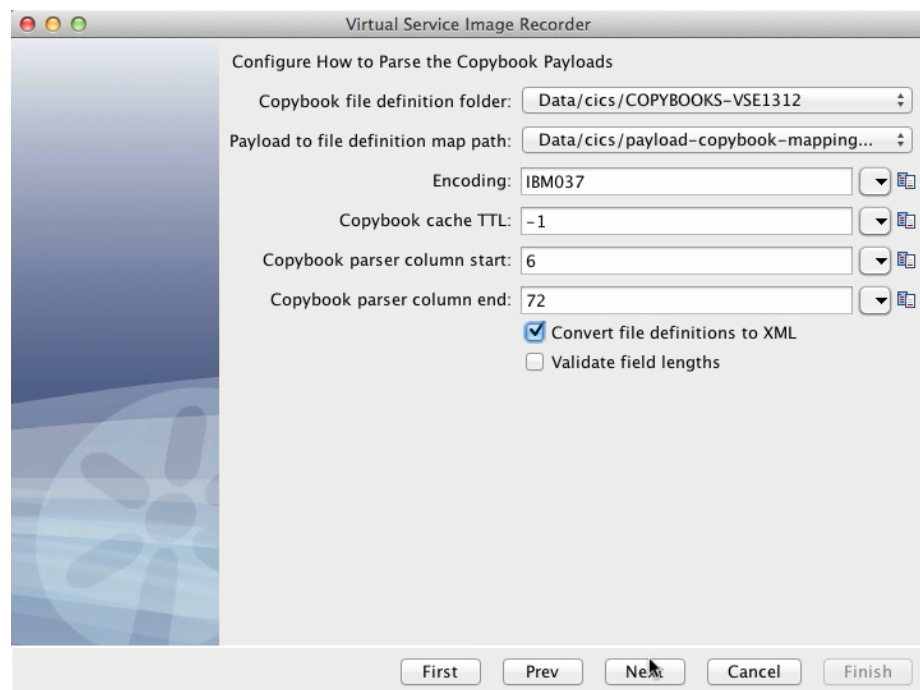
2. Cliquez sur Next (Suivant).

3. Entrez une adresse IP et un port, puis cliquez sur Next (Suivant).

La fenêtre de sélection Data Protocol (Protocole de données) s'ouvre.

4. Sélectionnez le protocole de données CTG Copybook pour les côtés demande et réponse, puis cliquez sur Next (Suivant).

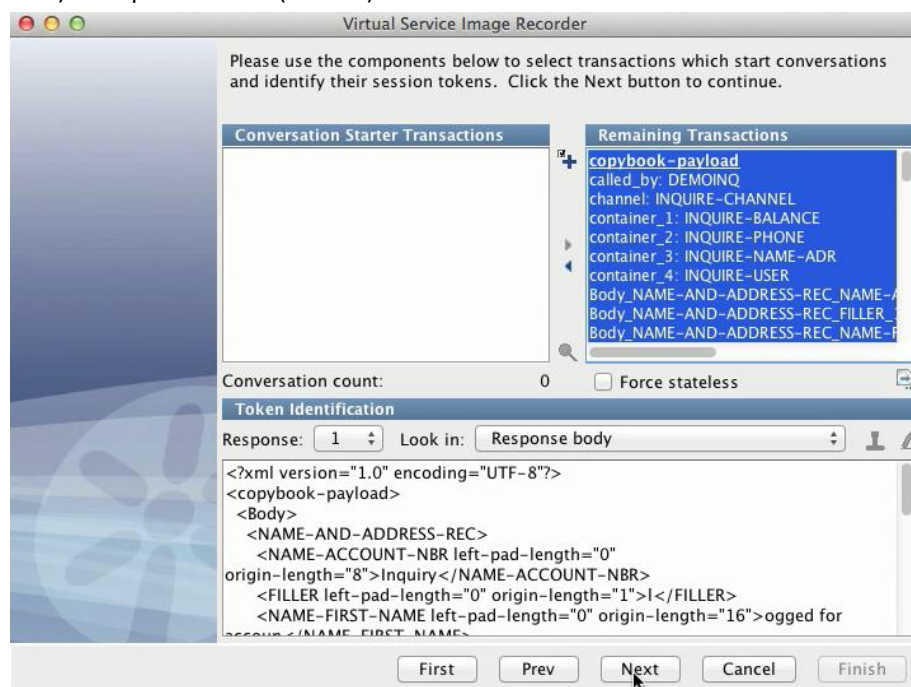
5. Cliquez sur Next (Suivant).



6. Entrez les paramètres suivants dans les définitions côté demande :

- Dossier des définitions de fichier copybook
- Chemin d'accès au mappage de charges utiles vers les définitions de fichier
- Informations de codage

7. Sélectionnez Convert file definitions to XML (Convertir des définitions de fichier en XML) et cliquez sur Next (Suivant).



8. Pour remplir l'image de service, cliquez sur Next (Suivant).
9. Pour afficher l'image de service, sélectionnez View Service Image (Afficher une image de service) et cliquez sur Finish (Terminer).

L'image de service terminée s'ouvre.

Désensibilisation des données

Dans les cas où la couche de transport ne peut pas accéder aux données à désensibiliser, le VSE fournira un gestionnaire de protocole de données. Par exemple, envisagez d'utiliser un gzip dans SOAP sur un protocole HTTP ou de messagerie. Si le VSE reçoit une demande (ou une réponse) SOAP au format gzip, la désensibilisation des données ne pourra pas être effectuée à ce niveau. Dans ce cas, vous pouvez configurer l'enregistreur avec un protocole de données pour effectuer l'opération gunzip nécessaire. Pour gérer la désensibilisation après la conversion du SOAP en texte brut, ajoutez le protocole de désensibilisation des données.

En général, le VSE tente de désensibiliser des données au niveau le plus bas possible. La case à cocher *Desensitize* (Désensibiliser) de la première fenêtre de l'assistant d'enregistrement du VSE indique au protocole de transport d'effectuer une tentative de désensibilisation dès que possible. Toutefois, il arrive que la charge utile d'une demande soit opaque au protocole de transport. Prenons l'exemple de messages HTTP dans lesquels la charge utile est compressée, à l'aide de gzip ou FastInfoSet. Dans ce cas, le protocole de transport ne peut pas appliquer les règles de désensibilisation. Ce protocole de données de désensibilisation des données est créé pour traiter ces situations.

Ajoutez le protocole de données de désensibilisation des données côté demande, côté réponse (ou les deux) d'une chaîne de protocole de données. Ajoutez le protocole dès qu'un autre protocole convertit la charge utile opaque en un format lisible. Ce protocole de données est destiné uniquement à l'enregistrement, non à la lecture. Chaque demande ou réponse étant présentée au gestionnaire de protocoles, toutes les règles de désensibilisation spécifiées dans **LISA_HOME/desensitize.xml** sont appliquées.

Ce protocole ne requiert aucune information de configuration et ne présente donc aucune fenêtre dans l'assistant d'enregistrement.

Delimited Text (Texte délimité)

Le protocole de données Delimited Text (Texte délimité) permet d'analyser des données de texte délimité dans des arguments et des valeurs. Ce protocole extrait le corps de la demande ou de la réponse, l'analyse, puis le remplace par une représentation XML des données analysées. Par exemple :

`"name1=val1;name2=val2"`

devient :

`<name1>val1</name1><name2>val2</name2>`

Procédez comme suit:

1. Remplissez l'onglet [Basics](#) (page 131) (Paramètres de base) de l'enregistreur de service virtuel.
2. Sélectionnez le Protocole de données de texte délimité côtés demande et réponse.
3. A l'issue de l'enregistrement, sélectionnez le format de votre demande de texte délimité :

Name/Value Pairs (Paires nom-valeur)

Définit le délimiteur entre les paires nom-valeur et le délimiteur entre le nom et la valeur. Par exemple, dans le cas des données suivantes :

`"name1=val1,name2=val2,name3=val3"`

le délimiteur entre les paires est une virgule et le délimiteur entre le nom et la valeur est le signe égal.

List of Values (Liste de valeurs)

Définit le délimiteur entre les valeurs. Vous pouvez utiliser deux types de listes :

- Liste de valeurs séparées par un délimiteur de texte

Par exemple, dans le cas des données suivantes :

`"val1,val2,val3,val4"`

le délimiteur est une virgule. Les paramètres sont nommés en fonction de leur position.

- Liste de valeurs séparées par de nouveaux caractères de ligne, par exemple un retour chariot ou un saut de ligne

Les paramètres sont nommés en fonction de leur position.

Fixed Width (Largeur fixe)

Définit (sous forme de nombre entier) la largeur du champ de données. Les paramètres sont nommés en fonction de leur position.

RegEx Delimited (Expression régulière de délimiteur)

Définit une expression régulière qui recherche les valeurs. Par exemple, dans le cas des données suivantes :

```
xxxx123xxx456xx789
```

L'expression régulière `\d\d\d` détecte les valeurs 123, 456 et 789 et ignore le reste. Les paramètres sont nommés en fonction de leur position.

Line Delimited (Séparé par des lignes)

4. Renseignez les champs suivants :

Field Names Path (Chemin d'accès aux noms de champ)

Définit l'emplacement d'un document de noms de champ, à savoir un document séparé par des lignes qui spécifie une liste de noms de champs triée. Par défaut, les champs sont nommés **value1**, **value2**, **value3**, etc. Pour spécifier des noms différents pour ces éléments XML, utilisez un document de noms de champ.

Delimiter Type (Type de délimiteur)

Définit le type de délimiteur utilisé pour séparer les paires de nom-valeur et les listes de valeurs. Les délimiteurs de texte ou de valeurs hexadécimales sont automatiquement sélectionnés en fonction du délimiteur spécifié.

Valeurs : Tous les caractères alphanumériques et les éléments suivants :

- `\r` (retour chariot)
- `\n` (nouvelle ligne)
- `\t` (tabulation)

Vous pouvez également utiliser la notation hexadécimale pour spécifier des délimiteurs.

Remarque : Vous pouvez uniquement utiliser des délimiteurs qui sont des XML 1.0 valides. Si vous spécifiez des caractères de contrôle non imprimables, l'image de service deviendra inutilisable.

XML Elements as request arguments (Éléments XML comme arguments de la demande)

Spécifie si des paramètres et des valeurs associées doivent être ajoutés aux demandes comme arguments.

Valeurs :

- Si l'option est **sélectionnée**, les paramètres et les valeurs associées sont automatiquement ajoutés à la demande comme arguments.
- Si elle est **désélectionnée**, vous devez utiliser l'[analyseur de charge utile XML générique](#) (page 262) (ou un protocole similaire) pour sélectionner les valeurs à convertir en arguments.

Remarque : Cette case à cocher ne s'applique pas lorsque le protocole de données de texte délimité est utilisé côté réponse.

5. Cliquez sur Next (Suivant).

Les paires nom-valeur sont représentées au format XML. Ici, vous pouvez double-cliquer sur une transaction pour afficher son contenu.

6. Configurez les délimiteurs côté réponse de la même manière que du côté demande.

A l'issue du traitement, vous pouvez afficher l'image de service virtuel et la charge utile convertie au format XML.

Informations complémentaires :

[Protocole de données d'analyseur de charge utile XML générique](#) (page 262)

DRDA

Le protocole de données DRDA convertit les charges utiles DRDA (Architecture de base de données relationnelles distribuée) au format XML lors de l'enregistrement. Ce protocole améliore l'alignement avec la fonctionnalité DevTest native, la lisibilité et la prise en charge de données dynamiques. Les réponses sont reconverties au format natif lors de la lecture.

Lorsque vous sélectionnez le protocole de transport de données DRDA, les champs des protocoles de données côté demande et réponse sont automatiquement remplis avec les sélections du protocole de données DRDA.

La fenêtre suivante de l'enregistreur permet de spécifier des informations de communications.

Virtual Service Image Recorder

Please provide us with the port the client will talk to us on, the server's host name, and the port the server listens on.

Listen/Record on port: 1306

Target host: 140.78.252.87

Target port: 1306

DB2 IP address: 140.70.252.10.190.3.1

LISA IP address: 10.22.162.147

Stored proc param delimiter:

First Prev Next Cancel Finish

Saisissez les paramètres de communications pour l'image de service.

Remarque : Si la charge utile contient des caractères autres que ASCII, la charge utile s'affiche au format binaire dans l'éditeur d'images de service.

Protocole de données EDI X12

Le protocole de données EDI X12 convertit les documents ANSI X12 EDI en représentation XML dans le corps de la demande. Le gestionnaire de protocole de données crée également une opération de demande VSE comprenant le type de document EDI (par exemple, 835) combiné à la version de document spécifique (par exemple, 004010) et une série d'arguments de demande de VSE. Ces arguments de demande sont la représentation aplanie du corps XML. Ils sont formés par la combinaison des éléments XML, dont les balises sont séparées par un trait de soulignement (_) et chaque attribut de balise est séparé par le signe @, pour le nom d'argument et la valeur d'élément pour sa valeur.

Par exemple, avec le document ANSI X12 EDI 850 suivant :

```
ISA*00*  *00*  *ZZ*0011223456  *ZZ*999999999
*990320*0157*U*00300*000000015*0*P*~$
GS*PO*0011223456*999999999*950120*0147*5*X*003040$
ST*850*000000001$
BEG*00*SA*95018017***950118$
N1*SE*UNIVERSAL  WIDGETS$
N3*375  PLYMOUTH PARK*SUITE  205$
N4*IRVING*TX*75061$
N1*ST*JIT  MANUFACTURING$
N3*BUILDING 3B*2001 ENTERPRISE PARK$
N4*JUAREZ*CH**MEX$
N1*AK*JIT  MANUFACTURING$
N3*400  INDUSTRIAL PARKWAY$
N4*INDUSTRIAL AIRPORT*KS*66030$
N1*BT*JIT  MANUFACTURING$
N2*ACCOUNTS PAYABLE DEPARTMENT$
N3*400  INDUSTRIAL PARKWAY$
N4*INDUSTRIAL AIRPORT*KS*66030$
P01*001*4*EA*330*TE*IN*525*VN*X357-W2$
PID*F****HIGH PERFORMANCE WIDGET$
SCH*4*EA****002*950322$
CTT*1*1$
SE*20*000000001$
GE*1*5$
```

l'objet de la demande VSE obtenu contiendra :

Operation : 850-00340

Arguments : voici un sous-ensemble des paires valeur-nom :

```
interchange_sender_address <null>
interchange_sender_address@Id      0011223456
interchange_sender_address@Qual    ZZ
interchange_receiver_address <null>
```

```

interchange_receiver_address@Id 999999999
interchange_receiver_address@Qual ZZ
interchange_group_transaction_segment_element_1 00
...

```

Body (Corps) : structure appliquée comme suit. Le document XML a été formaté ici à des fins de lisibilité. Le corps de la demande ne contiendra pas d'éléments de formatage tels que des fins de ligne et la mise en retrait.

```

<?xml version="1.0" encoding="UTF-8"?>
<ediroot>
  <interchange Standard="ANSI X.12" Date="990320" Time="0157"
StandardsId="U" Version="00300"
  Control="000000015">
    <sender>
      <address Id="0011223456 " Qual="ZZ"/>
    </sender>
    <receiver>
      <address Id="999999999 " Qual="ZZ"/>
    </receiver>
    <group GroupType="P0" ApplSender="0011223456"
ApplReceiver="999999999" Date="950120"
      Time="0147" Control="5" StandardCode="X"
StandardVersion="003040">
      <transaction DocType="850" Name="Purchase Order"
Control="000000001">
        <segment Id="BEG">
          <element Id="BEG01">00</element>
          <element Id="BEG02">SA</element>
          <element Id="BEG03">95018017</element>
          <element Id="BEG06">950118</element>
        </segment>
        <loop Id="N1">
          <segment Id="N1">
            <element Id="N101">SE</element>
            <element Id="N102">UNIVERSAL
WIDGETS</element>
          </segment>
          <segment Id="N3">
            <element Id="N301">375 PLYMOUTH
PARK</element>
            <element Id="N302">SUITE 205</element>
          </segment>
          <segment Id="N4">
            <element Id="N401">IRVING</element>
            <element Id="N402">TX</element>
            <element Id="N403">75061</element>
          </segment>
        </loop>
      </loop Id="N1">

```

```

        <segment Id="N1">
            <element Id="N101">ST</element>
            <element Id="N102">JIT
MANUFACTURING</element>
        </segment>
        <segment Id="N3">
            <element Id="N301">BUILDING 3B</element>
            <element Id="N302">2001 ENTERPRISE
PARK</element>
        </segment>
        <segment Id="N4">
            <element Id="N401">JUAREZ</element>
            <element Id="N402">CH</element>
            <element Id="N404">MEX</element>
        </segment>
    </loop>
    <loop Id="N1">
        <segment Id="N1">
            <element Id="N101">AK</element>
            <element Id="N102">JIT
MANUFACTURING</element>
        </segment>
        <segment Id="N3">
            <element Id="N301">400 INDUSTRIAL
PARKWAY</element>
        </segment>
        <segment Id="N4">
            <element Id="N401">INDUSTRIAL
AIRPORT</element>
            <element Id="N402">KS</element>
            <element Id="N403">66030</element>
        </segment>
    </loop>
    <loop Id="N1">
        <segment Id="N1">
            <element Id="N101">BT</element>
            <element Id="N102">JIT
MANUFACTURING</element>
        </segment>
        <segment Id="N2">
            <element Id="N201">ACCOUNTS PAYABLE
DEPARTMENT</element>
        </segment>
        <segment Id="N3">
            <element Id="N301">400 INDUSTRIAL
PARKWAY</element>
        </segment>
        <segment Id="N4">
```



```

        <element Id="N401">INDUSTRIAL
AIRPORT</element>
        <element Id="N402">KS</element>
        <element Id="N403">66030</element>
    </segment>
</loop>
<loop Id="P01">
    <segment Id="P01">
        <element Id="P0101">001</element>
        <element Id="P0102">4</element>
        <element Id="P0103">EA</element>
        <element Id="P0104">330</element>
        <element Id="P0105">TE</element>
        <element Id="P0106">IN</element>
        <element Id="P0107">525</element>
        <element Id="P0108">VN</element>
        <element Id="P0109">X357-W2</element>
    </segment>
    <loop Id="PID">
        <segment Id="PID">
            <element Id="PID01">F</element>
            <element Id="PID05">HIGH PERFORMANCE
WIDGET</element>
        </segment>
    </loop>
</loop>
<loop Id="SCH">
    <segment Id="SCH">
        <element Id="SCH01">4</element>
        <element Id="SCH02">EA</element>
        <element Id="SCH06">002</element>
        <element Id="SCH07">950322</element>
    </segment>
</loop>
</loop>
<loop Id="CTT">
    <segment Id="CTT">
        <element Id="CTT01">1</element>
        <element Id="CTT02">1</element>
    </segment>
</loop>
</transaction>
</group>
</interchange>
</ediroot>

```

Protocole de données d'analyseur de charge utile XML générique

L'analyseur de charge utile XML générique détermine si les demandes et les réponses sont des chaînes XML. Ce protocole permet d'identifier les variables dans les messages XML utilisés par l'enregistreur.

Remarque : Pour le [protocole de données Delimited Text \(Texte délimité\)](#) (page 254), le [protocole de données Copybook](#) (page 234) et le [protocole de données DRDA](#) (page 257), vous pouvez activer cette fonctionnalité en sélectionnant la case à cocher XML Elements as request arguments (Éléments XML comme arguments de la demande) dans les fenêtres de configuration de protocole de données.

Identification des conversations et des transactions

La qualité de l'image de service enregistrée dépend directement de la quantité d'informations dont dispose VSE pour identifier les conversations et chaque transaction qu'elles contiennent. Notamment, le VSE requiert de l'aide pour différencier les transactions les unes des autres :

- **Dans le cadre d'une conversation :** identification d'un ID unique représentant une session
- **Comme opération spécifique :** identification d'informations spécifiques à la sémantique de l'opération. Par exemple : distinction entre une opération de création de demande et une opération de liste des demandes.

En interne, le VSE connaît de nombreux modèles à rechercher. Par exemple, pour la virtualisation HTTP, si le serveur utilisé est un serveur Java, le VSE définit un cookie qui contient la valeur d'une variable spéciale appelée sessionid. La variable sessionid identifie la session de manière unique. Le VSE peut utiliser la variable pour distinguer différentes sessions.

Toutefois, le VSE requiert parfois une assistance supplémentaire, que vous pouvez fournir comme suit :

- Dans l'enregistrement HTTP, le VSE vous permet d'identifier des jetons.
- Dans le cas du protocole JMS, vous pouvez déterminer si le VSE utilise l'ID de corrélation JMS, des en-têtes JMS personnalisés ou tous les en-têtes JMS pour cette identification.
- Le VSE permet de spécifier un protocole de données dynamiques grâce auquel vous obtenez des informations explicites issues du message même. La section suivante décrit cette technique.

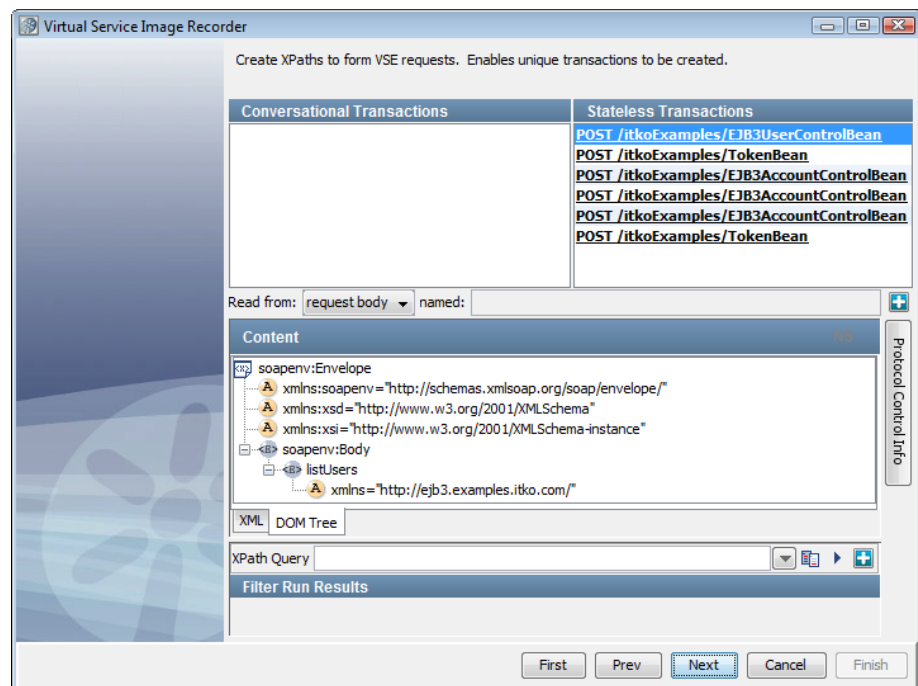
L'utilisation de l'analyseur de charge utile XML générique est une technique permettant d'aider le VSE à examiner le corps des messages enregistrés (charge utile) et d'en extraire des informations explicites pour identifier les transactions. Cette technique peut être la seule façon d'obtenir des informations explicite sur des conversations, notamment dans le cas de protocoles opaques tels que le protocole natif de WebSphere MQ.

Remarque : Lors de l'utilisation de l'analyseur de charge utile XML générique et du protocole de données Delimited Text (Texte délimité), ajoutez ce dernier avant l'analyseur de charge utile XML générique. Sinon, la demande ne s'affichera jamais comme analysée dans l'enregistreur.

Procédez comme suit:

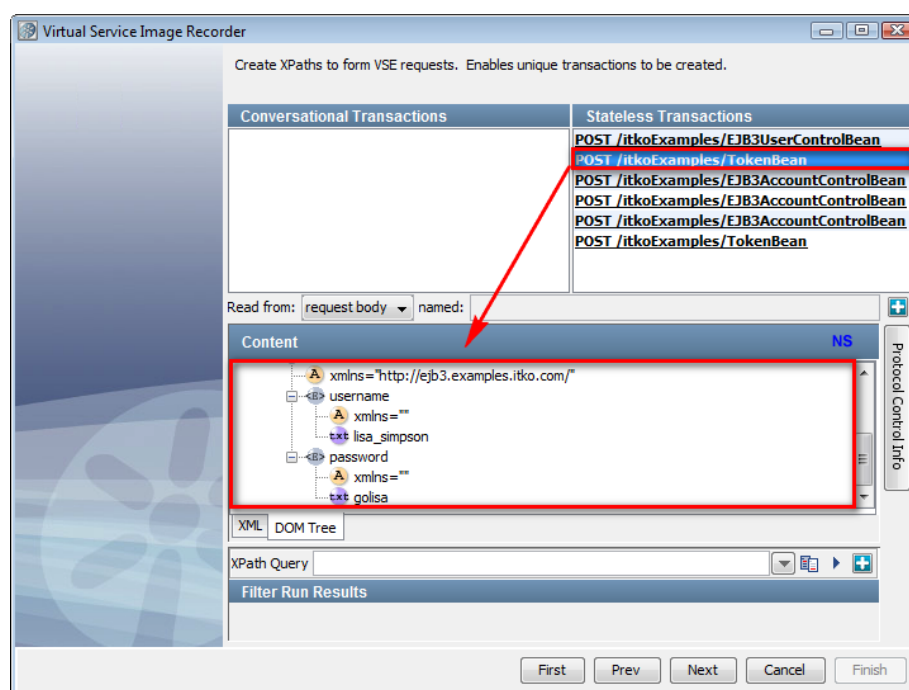
1. Pour utiliser un protocole de données dynamiques, dans l'onglet Data Protocols (Protocoles des données) de l'enregistreur d'image de service virtuel, sélectionnez Generic XML Payload Parse (Analyseur de charge utile XML générique), si la charge utile est au format XML correct.
2. Effectuez les étapes restantes, y compris l'étape de nettoyage.

Après l'étape de nettoyage, la fenêtre suivante s'affiche :




Par défaut, aucune transaction de démarreur n'est identifiée ; DevTest ne sait donc pas quelles sont les données à examiner pour identifier les conversations. Toutefois, la liste Other Transactions (Autres transactions) indique les transactions enregistrées.

3. Pour afficher la charge utile XML dans la zone Content (Contenu), cliquez sur la première transaction.




L'onglet XML dans la zone de Content (Contenu) contient la vue XML classique. L'onglet DOM Tree (Arborescence DOM) affiche la charge utile sous forme d'arborescence.

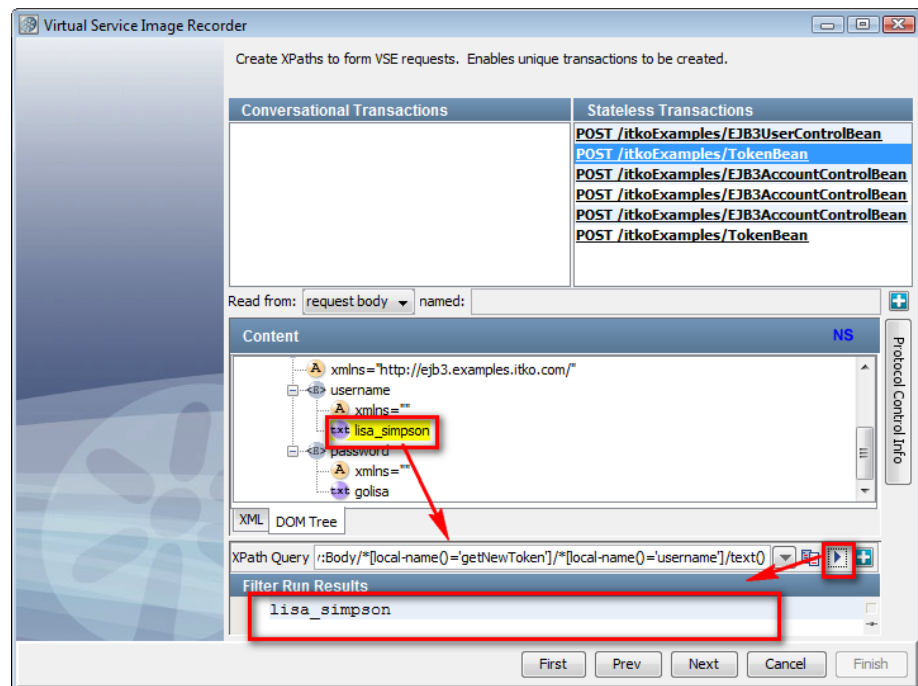
Remarque : Cette partie de la fenêtre est similaire au panneau qui s'affiche lors de la création d'un filtre XPath de résultat XML. Comme décrit dans la section *Filtre XPath de résultat XML* de la rubrique *Utilisation*, vous pouvez utiliser la propriété **lisa.xml.xpath.computeXPath.alwaysUseLocalName** pour ignorer l'espace de noms.



4. Pour identifier un noeud spécifique comme paramètre pour le VSE, cliquez sur le noeud dans l'onglet DOM Tree (Arborescence DOM). La zone XPath Query (requête XPath) affiche la requête qui correspond au noeud sélectionné. Pour évaluer la requête XPath sur la charge utile actuelle, cliquez sur .

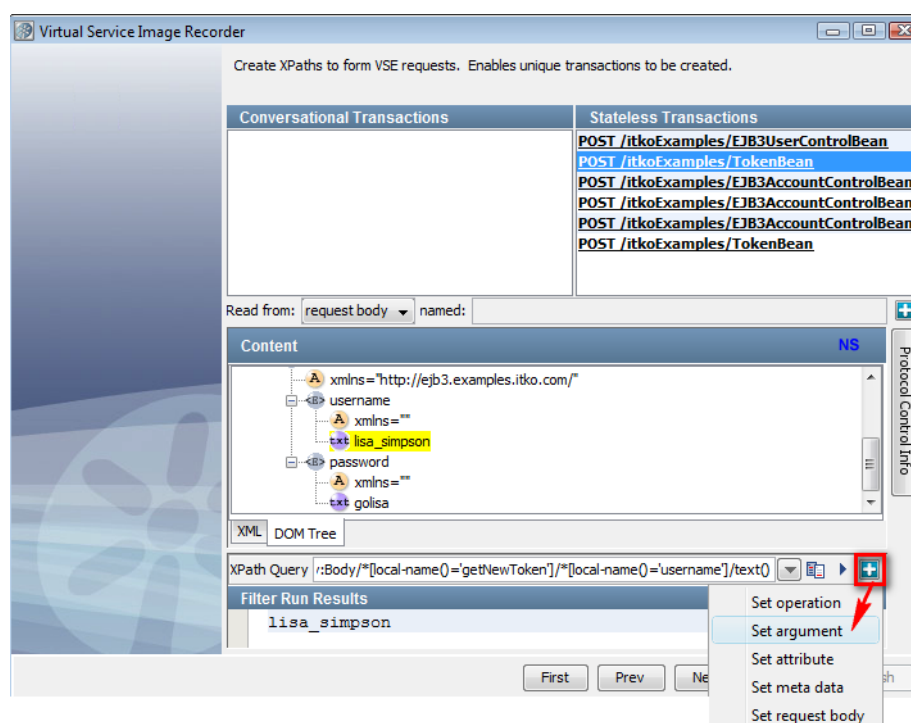
Le résultat de l'évaluation s'affiche sous l'en-tête Filter Run Results (Résultats de l'exécution du filtre).

5. Pour ajouter cette requête XPath particulière à la liste de paramètres, cliquez sur Add (Ajouter) .

Un menu s'affiche et permet de définir une opération, un argument, un métaparamètre, ou le corps de la demande. Par exemple, utilisez le menu pour sélectionner un document SOAP intégré au corps de la demande de sorte à utiliser le protocole WS SOAP pour analyser le document.



6. Cliquez sur Protocol Control Info (Informations sur le contrôle de protocole). DevTest a mémorisé la chaîne XPath identifiée et lui a attribué un nom d'argument. Vous pouvez renommer l'argument. Egalement dans ce panneau, utilisez les boutons Save (Enregistrer)  et Restore (Restaurez)  pour enregistrer votre liste d'expressions XPath dans un fichier, ou charger une liste d'expressions XPath à partir d'un fichier. L'enregistrement et le chargement permettent de copier et coller facilement d'un analyseur de charge utile XML à un autre.



De même, vous pouvez extraire des variables supplémentaires à partir de cette transaction ou d'autres. Il n'est pas nécessaire que toutes les transactions contiennent l'argument spécifique. Lors du traitement, si un argument particulier n'est pas présent dans la charge utile, il sera ignoré.

Si vous faites défiler le panneau Protocol Control Info (Informations sur le contrôle de protocole) vers la droite, une colonne NS apparaît et permet d'ajouter ou de supprimer des définitions d'espace de noms. Pour rechercher et importer un fichier XML à partir du système de fichiers pour utiliser les définitions d'espace de noms inclus dans ce fichier, utilisez le bouton Browse (Parcourir).

7. Double-cliquez sur une transaction dans l'onglet XML pour afficher une boîte de dialogue indiquant le contenu de la transaction.
8. Lorsque vous êtes satisfait de vos choix de variables, cliquez sur Next (Suivant). Vous serez ensuite dirigé vers la fenêtre post-traitement.

Protocole de données JSON

Le gestionnaire de protocoles de données JSON est utilisé pour convertir des données JSON au format XML, puis pour convertir les données XML au format JSON.

Par exemple, les données JSON suivantes

```
[
  {
    "organizationType": "W",
    "parentOrganizationType": "S",
    "parentUnitNbr": 777,
    "positionName": "S",
    "positionType": "S",
    "positionTypeId": 56,
    "unitNbr": 433,
    "l": [
      {
        "Roles": [
          "P", "P1", "P2", "S", "C", "AC", "ACF", "ACM"
        ]
      }
    ],
    "groupKey": "P",
    "groupName": "P"
  }
]
```

sont converties comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <element class="object">
    <groupKey type="string">P</groupKey>
    <groupName type="string">P</groupName>
    <l class="array">
      <element class="object">
        <Roles class="array">
          <element type="string">P</element>
          <element type="string">P1</element>
          <element type="string">P2</element>
          <element type="string">S</element>
          <element type="string">C</element>
          <element type="string">AC</element>
          <element type="string">ACF</element>
          <element type="string">ACM</element>
        </Roles>
      </element>
    </l>
    <organizationType type="string">W</organizationType>
    <parentOrganizationType type="string">S</parentOrganizationType>
    <parentUnitNbr type="number">777</parentUnitNbr>
    <positionName type="string">S</positionName>
    <positionType type="string">S</positionType>
    <positionTypeId type="number">56</positionTypeId>
    <unitNbr type="number">433</unitNbr>
  </element>
</root>
```

```
</element>
</root>
```

Dans le cas du gestionnaire de protocole de données JSON, si les données JSON contiennent un tableau avec un seul élément, il sera converti en objet avec pour clé element. Par exemple,

```
[{"question1": "answer1", "question2": "answer2"}]
```

est converti comme suit :

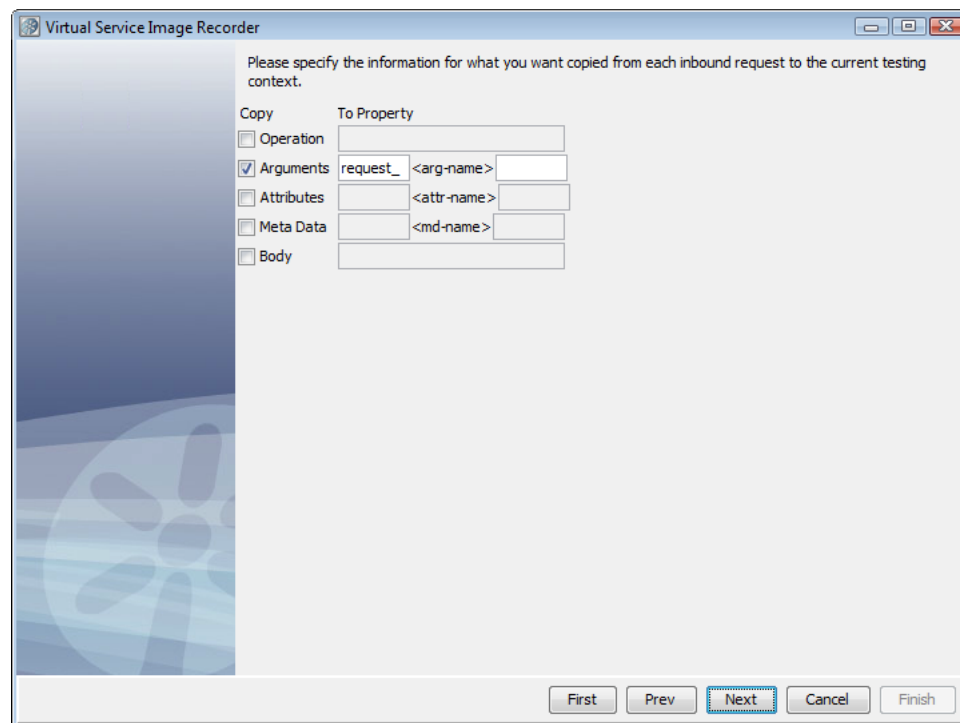
```
{"element": {"question1": "answer1", "question2": "answer2"}}
```

Le gestionnaire de protocole de données JSON réorganise les paires clé-valeur dans un objet dans l'ordre alphabétique des noms de clé.

Remarque : Si le type d'application est défini sur application/json, text/json ou text/javascript, l'enregistrement JSON est effectué normalement. Si un type d'application incorrect est défini, l'enregistrement n'est pas effectué.

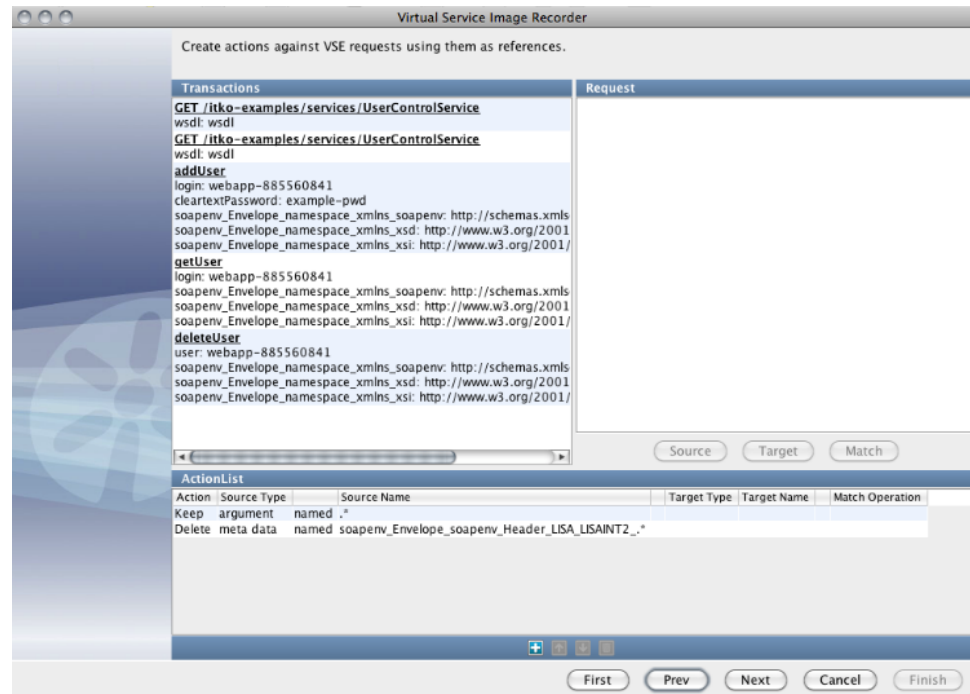
Copieur de données de la demande

Le Request Data Copier (Copieur de données de la demande) permet de copier les données de la demande entrante actuelle dans le contexte de test actuel. Cette fonctionnalité permet d'examiner plus facilement des informations de demande lorsque le comportement personnalisé est requis avant l'étape de recherche de réponse par le VSE dans le modèle de service virtuel.



Protocole de données Request Data Manager (Gestionnaire de données de demande)

Dans la fenêtre Data Protocols (Protocoles de données), sélectionnez Request Data Manager (Gestionnaire de données de demande) pour un protocole de données. A l'issue de votre enregistrement, la fenêtre suivante s'affiche :



Le protocole de données Request Data Manager (Gestionnaire de données de demande) permet de modifier des demandes du VSE pendant l'enregistrement ou la lecture.

Fondamentalement, ce protocole permet d'appliquer une liste d'actions au niveau d'une demande. Vous pouvez ajouter les actions suivantes dans la section ActionList (Liste d'actions) de la fenêtre :

Copy (Copier)

Permet de copier des données de la demande vers une autre partie de celle-ci.

Move (Déplacer)

Permet de déplacer des données de la demande vers une autre partie de celle-ci.

Supprimer

Permet de supprimer (ou d'effacer) des données de la demande.

Conserver

Permet de conserver des données d'une demande tout en supprimant tout autre élément dans ce groupe pour la valeur de données.

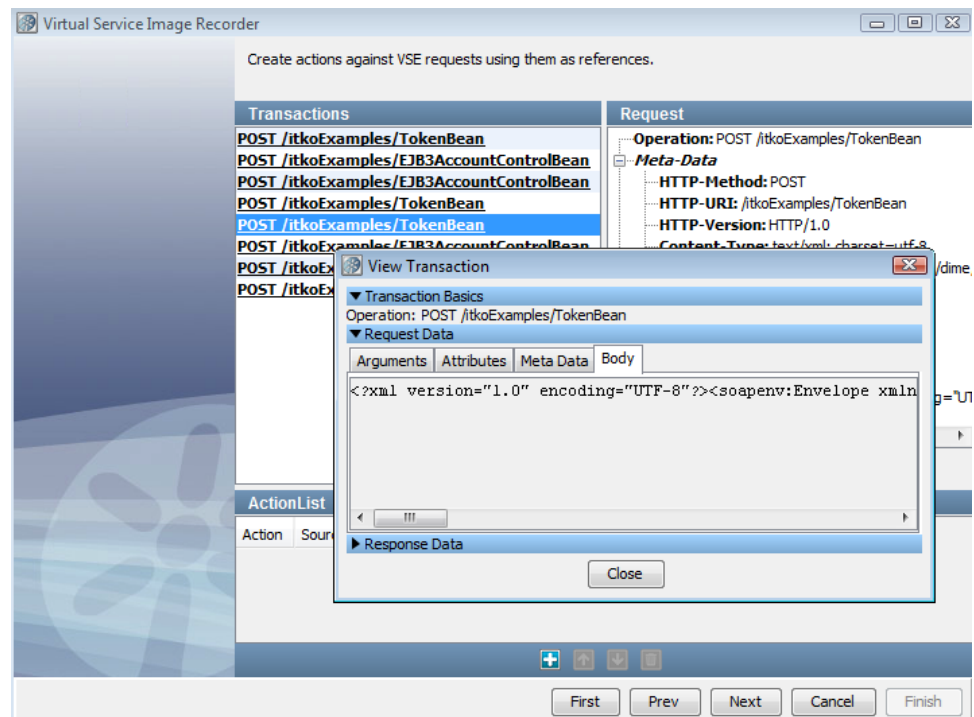
Toutes ces actions peuvent être appliquées à l'opération de demande, un argument, un attribut, une entrée de métadonnées ou le corps de la demande. Par exemple, lors de la virtualisation de Java, qui se finalise par des documents XML comme arguments, vous pouvez modifier ou copier la valeur d'un argument dans le corps de la demande, de sorte que d'autres protocoles de données puissent accéder à cet argument.

Remarque : Keep (Conserver) est l'action la plus explicite pour des arguments, des attributs et des métadonnées. Si vous conservez spécifiquement une valeur à partir de l'un de ces trois groupes, toute autre valeur dans ce groupe non référencée par une action dans la liste pour le protocole de données sera supprimée. Si vous conservez un argument, tous les autres seront supprimés, sauf s'ils étaient la cible d'un déplacement ou d'une copie. Cette technique permet de supprimer des arguments inutiles.

Vous pouvez également limiter l'application de chaque action aux demandes dont les opérations correspondent uniquement à une expression régulière spécifique.

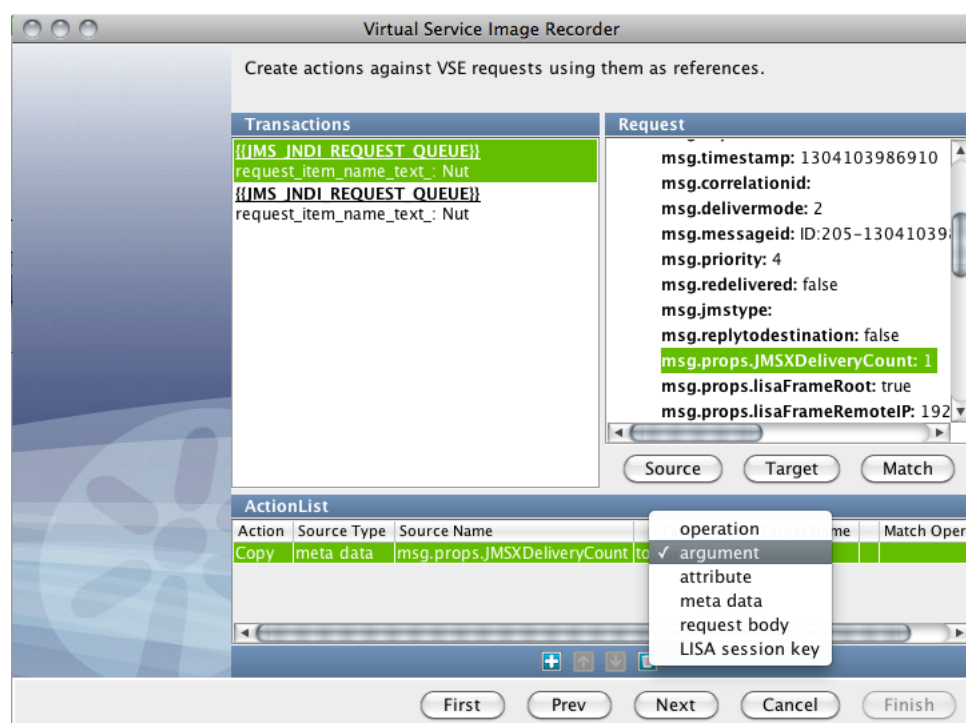
Dans l'assistant d'enregistrement ou l'éditeur de modèle contenant un gestionnaire de protocole de données de gestionnaire de données de demande, ajoutez une action Keep (Conserver), Delete (Supprimer), ou les deux. Sélectionnez argument/attribut/meta data (argument/attribut/métadonnées) et spécifiez une expression régulière pour établir une correspondance comme nom. Vous devez également modifier la cellule contenant *named (nommé)* par *matches (correspond)*. Lors de l'exécution du gestionnaire de protocole de données, il conserve ou supprime tous les éléments dans la liste d'arguments, d'attributs, ou de métadonnées dont le nom correspond au modèle. Laisser le modèle de correspondance d'opération pour une action vide affecte toutes les demandes.

Dans cette liste de transactions, double-cliquez sur l'une d'elles pour afficher une boîte de dialogue indiquant son contenu.



Utilisation du protocole de données Request Data Manager (Gestionnaire de données de demande) pour définir les propriétés de message JMS et MQ

A l'issue de l'enregistrement, utilisez la fenêtre Request Data Manager (Gestionnaire de données de demande) pour ajouter les propriétés de message JMS ciblées par les arguments de la demande. Les propriétés de message JMS sont disponibles sous les métadonnées de la demande et contiennent un préfixe **msg.** pour les propriétés JMS standard, telles que l'ID de corrélation et un préfixe **msg.props.** pour les propriétés de message personnalisées. Pour copier une propriété dans les arguments de la demande, sélectionnez argument dans la liste déroulante :



MQ fonctionne de la même manière.

Pour définir une clé de session avec état au lieu d'un argument, sélectionnez plutôt session key (Clé de session) dans la liste déroulante.

Vous pouvez utiliser un seul protocole de données Request Data Manager (Gestionnaire de données de demande) pour définir un nombre d'arguments et la clé de session simultanément.

Protocole de données REST

Le protocole REST permet d'appeler des services Web à l'aide du protocole HTTP.

Le gestionnaire de protocole de données REST analyse les demandes HTTP conformes au style d'architecture REST. Le protocole de données identifie les parties dynamiques des chaînes d'URI. Le résultat obtenu est un ensemble de règles. Pendant la phase de lecture, VSE utilise les règles pour virtualiser les demandes HTTP qui appellent les mêmes opérations.

Si VSE ne sélectionne pas automatiquement le protocole de données REST, vous pouvez le sélectionner manuellement dans la page des protocoles de données de l'enregistreur d'image de service Virtuel.

Vous pouvez inclure les demandes HTTP dans le trafic dynamique ou dans des paires demande-réponse.

Vous devez utiliser le protocole de transport HTTP/S pour les paires demande-réponse.

Vous pouvez configurer certains aspects du processus d'analyse.

Chaque règle contient un ou plusieurs paramètres, qui représentent les parties dynamiques. Par défaut, les paramètres commencent par **URLPARAM**.

Les exemples de règles suivants incluent les paramètres **URLPARAM0** et **URLPARAM1**. Les paramètres doivent être placés entre accolades dans la règle. Une fois que l'analyseur génère la règle, vous pouvez changer l'identificateur **URLPARAM**.

```
GET /Service/rest/user/{URLPARAM0}
GET /Service/rest/customer/{URLPARAM0}
GET /Service/rest/customer/{URLPARAM0}/order/{URLPARAM1}
```

Lors de la lecture, les chaînes d'URI suivantes correspondent à la première règle :

```
GET /Service/rest/user/100
GET /Service/rest/user/101
```

Lors de la lecture, la chaîne d'URI suivante correspond à la troisième règle :

```
GET /Service/rest/customer/1234/order/5678
```

Vous pouvez également combiner ces paramètres avec un texte constant supplémentaire si nécessaire. Par exemple :

```
GET /Service/rest/user/{USERNAME}/format.{type}
```

Dans cet exemple, le dernier jeton pourrait être "format.json" ou "format.xml".

Un jeton peut contenir le nombre de paramètres de votre choix à l'emplacement de votre choix. Par exemple :

GET

```
/Service/rest/users/format.{type}.sortorder.{order}.filter.{filter}
```

Vous pouvez utiliser l'éditeur de règles REST qui s'affiche après un enregistrement pour ajouter ou modifier des règles de manière similaire à l'exemple, en fonction de vos besoins. Vous pouvez également générer des règles similaires à partir de fichiers WADL ou RAML.

Format des paires demande-réponse

Cette rubrique décrit le format des paires demande-réponse pour le protocole de données REST.

Request (Demande)

Le fichier de demande doit inclure un en-tête HTTP valide. L'URL est la première ligne de l'en-tête. Les autres lignes d'en-tête sont facultatives.

Le gestionnaire de protocole de données REST prend en charge toutes les méthodes ou les syntaxes HTTP : GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT et PATCH.

Le format de la ligne URL est le suivant :

<méthode><espace><chemin_accès_API_REST><espace><version_HTTP>

Par exemple :

PUT /rest-example/control/users/save HTTP/1.1

Si la demande inclut un corps, séparez le corps de l'en-tête à l'aide d'une ligne vide.

L'exemple suivant présente une demande qui contient un corps au format JSON.

```
PUT /rest-example/control/users/save HTTP/1.1
accept: application/json
content-Type: application/json
Connection: Keep-Alive
User-Agent: LISA
```

```
{
  "user": {
    "emailAddress": "test@test.com",
    "firstName": "first-9",
    "lastName": "last-9",
    "password": "aaaaaaaa",
    "username": "dmxxx-009"
  }
}
```

Vous pouvez condenser le corps JSON ou XML sur une ligne unique.

```
{ "user": { "emailAddress": "test@test.com", "firstName": "first-9",
"lastName": "last-9", "password": "aaaaaaaa", "username":
"dmxxx-009"  }}
```

Response (Réponse)

Le fichier de réponse contient un code de réponse HTTP. Il peut contenir un en-tête et un corps. Le code de réponse doit être la première ligne dans le fichier, au format suivant :

<version_HTTP><espace><code_réponse_HTTP>

L'exemple suivant présente une réponse contenant un corps.

HTTP/1.1 200

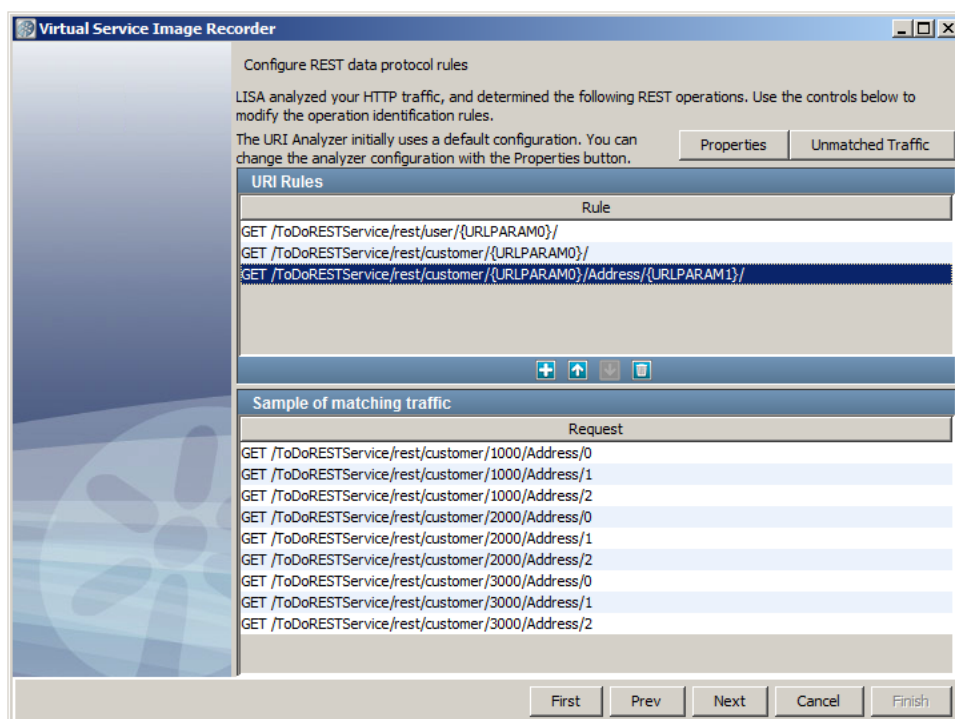
```
"user":{"emailAddress":"lisa.simpson@itko.com","firstName":"lisa",  
"lastName":"simpson","password":"60fAFoq+W0R4HrLgsfPodkWRw9I=",  
"phoneNumber":"","username":"lisa_simpson"}}
```

Si la réponse n'inclut pas une ligne de code de réponse, le code de réponse est remplacé par défaut par **200 (OK)**.

Révision et modification des règles

L'enregistreur d'image de service virtuel et l'interface du service virtuel à partir des paires demande-réponse inclut une page dans laquelle vous pouvez réviser et modifier les règles créées par le protocole de données REST.

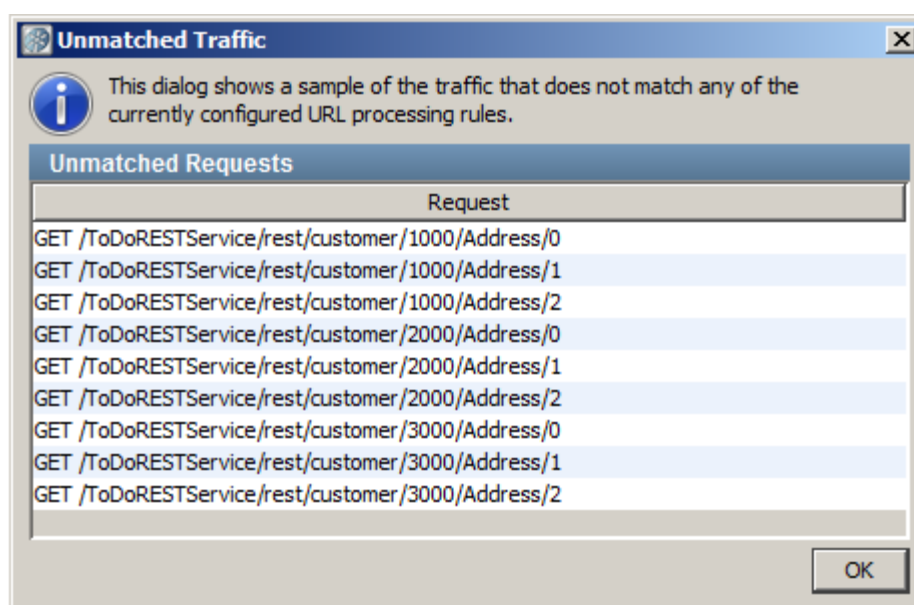
Le graphique suivant présente la fenêtre dans l'enregistreur d'image de service virtuel.



Le volet supérieur (URI Rules (Règles d'URI)) affiche les règles. Lorsque vous sélectionnez une règle, le volet inférieur (Sample of matching traffic (Exemple de trafic correspondant)) affiche un exemple de trafic qui correspond à la règle. Pour configurer le nombre maximum de lignes qui s'affichent, définissez la propriété **lisa.protocol.rest.editor.observedtraffic.max** dans le fichier **lisa.properties**.

Pour afficher une liste de demandes HTTP qui ne correspondent pas aux règles, cliquez sur Unmatched Traffic (Trafic non correspondant). Cette liste est vide si les règles correspondent à tout le trafic collecté. Pour configurer le nombre maximum de demandes qui s'affichent, définissez la propriété **lisa.protocol.rest.editor.unmatchedtraffic.max** dans le fichier **lisa.properties**.

Le volet URI Rules (Règles d'URI) vous permet d'ajouter, de mettre à jour, de réorganiser et de supprimer les règles.



Vous pouvez remplacer les paramètres dans les règles par des noms significatifs. Par exemple :

```
GET /Service/rest/customer/{customerid}/order/{orderid}
```

Vous pouvez créer plusieurs règles qui correspondent à la même opération. Les règles sont appliquées dans l'ordre affiché. En conséquence, une règle placée dans une position élevée dans la liste peut renvoyer une correspondance, alors que vous supposiez qu'une règle située plus bas dans la liste allait s'appliquer. Pour changer l'ordre, utilisez les boutons de réorganisation.

Si vous supprimez une règle, cliquez sur Unmatched Traffic (Trafic non correspondant) pour voir l'effet de la suppression de la règle sur le trafic capturé.

Pour changer les valeurs des propriétés de configuration suivantes, cliquez sur Properties (Propriétés).

Vous pouvez effectuer une nouvelle analyse du trafic en cliquant sur Properties (Propriétés) ; la page des propriétés s'affiche. Cliquez alors sur OK sans modifier les valeurs. Par exemple, vous pouvez effectuer cette action après avoir lié un protocole de données prenant en charge les scripts et un protocole de données REST.

Max Changes (Nombre maximum de modifications)

Définit le nombre maximum de modifications permises pour un jeton avant que la variabilité soit considérée suffisamment significative pour générer une règle.

Considérez un URI comme une liste de jetons séparée par le caractère "/". Par exemple, l'URI GET /rest/user/1234 contient les jetons :

- GET

- rest
- user (utilisateur)
- 1234

Pour modifier la valeur par défaut, définissez la propriété **lisa.protocol.rest.maxChanges** dans le fichier **lisa.properties**.

Start Position (Position de début)

Définit la position dans l'URL à partir de laquelle le protocole de données REST commence à rechercher des jetons variables.

La position de début est l'index d'un jeton dans la liste des jetons qui compose un URI. Par exemple, dans l'URI :

GET /service/rest/customers"

GET est à la position 0 et customers est à la position 3.

Pour modifier la valeur par défaut, définissez la propriété **lisa.protocol.rest.startPosition** dans le fichier **lisa.properties**.

Id Identification Regular Expression (Expression régulière d'identification d'ID)

Définit une chaîne d'expression régulière que le protocole de données REST utilise pour détecter des identificateurs de ressource dans les demandes HTTP. Pour modifier la valeur par défaut, définissez la propriété **lisa.protocol.rest.idPattern** dans le fichier **lisa.properties**.

URL Parameter Prefix (Préfixe de paramètre URL)

Définit le préfixe que le protocole de données REST utilise pour les paramètres dans des règles.

Le but de ce paramètre est d'aider l'analyseur à identifier les jetons qui suivent un modèle spécifique variable, mais que l'analyseur peut ne pas détecter automatiquement.

Par exemple, dans l'URI :

GET /rest/user/person-1234-dev

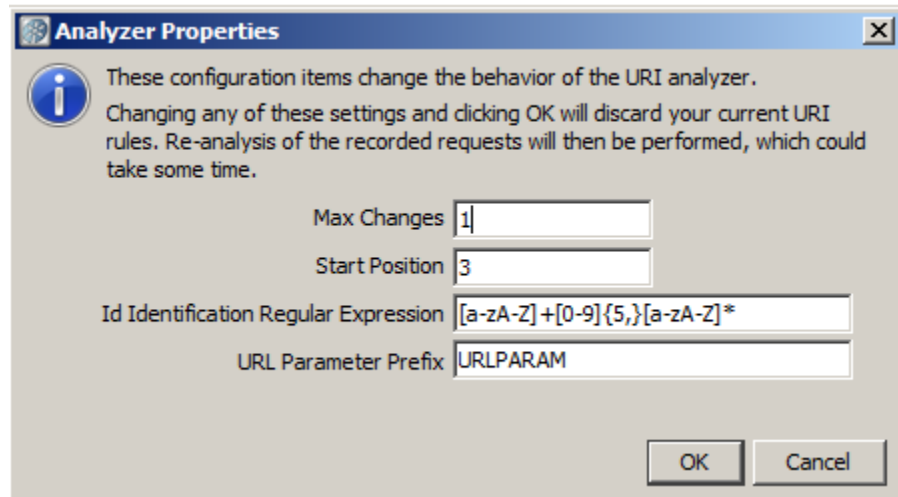
vous pouvez savoir qu'un ID d'utilisateur au format *person-nnnn-nnn* suit toujours user. Dans ce cas, vous pouvez définir une expression régulière pour détecter ce modèle directement. Dans cet exemple, l'expression régulière serait :

person-[0-9]{4}-[a-z]{3}

Pour modifier la valeur par défaut, définissez la propriété **lisa.protocol.rest.parameterBaseName** dans le fichier **lisa.properties**.

Si vous changez une ou plusieurs valeurs pour ces propriétés, le protocole de données réanalyse le trafic enregistré.

L'illustration suivante représente la fenêtre Analyzer Properties (Propriétés de l'analyseur).



Protocole de données prenant en charge les scripts

Le protocole de données Scriptable (prenant en charge les scripts) est disponible lorsque vous avez besoin d'une charge de traitement peu élevée sur la demande et/ou la réponse. Les exemples de scripts sont écrits en BeanShell. Vous pouvez écrire votre script dans le langage de script de votre choix pris en charge par CA Application Test.

Pour spécifier un langage autre que BeanShell pour le script, saisissez ce langage sur la première ligne du script.

Valeurs :

- applescript (OS X)
- beanshell
- freemarker
- groovy
- javaScript
- velocity

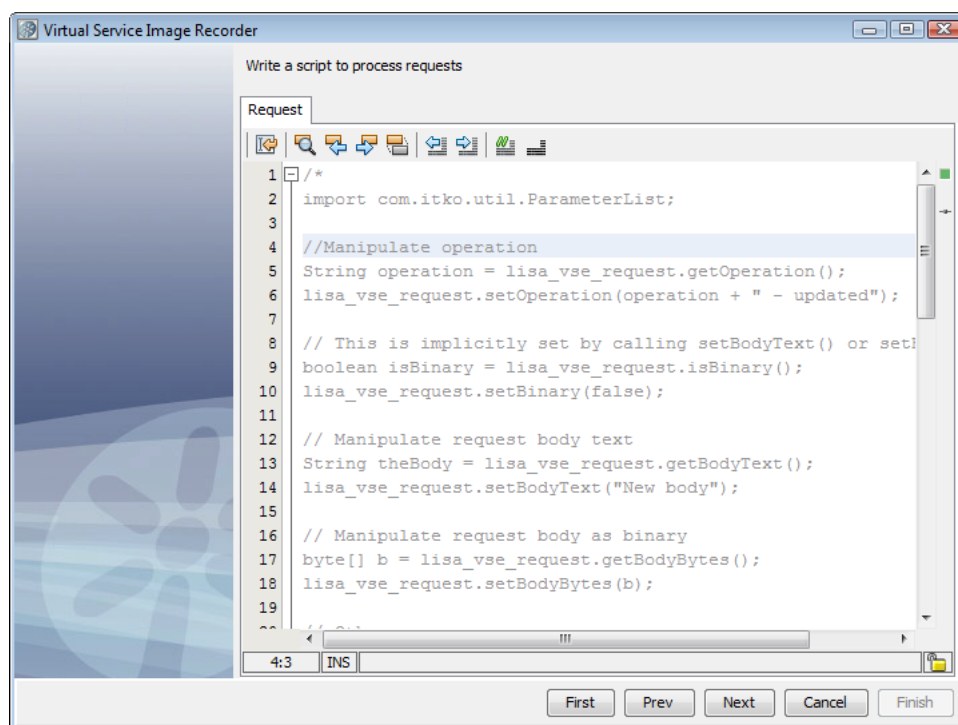
Format :

%Langage%

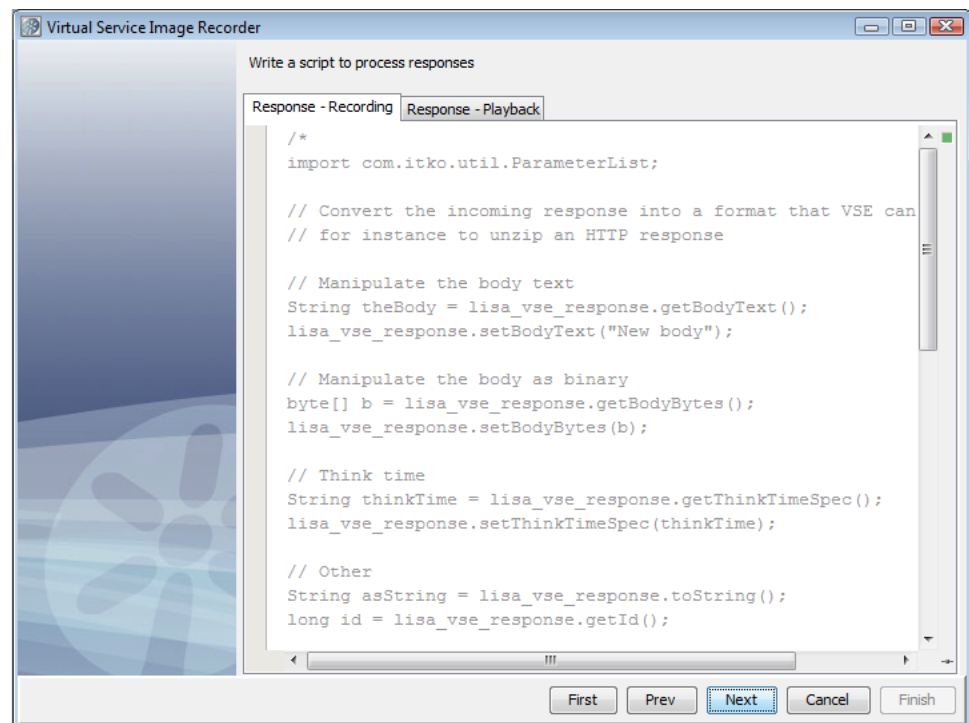
Valeur par défaut : beanshell

Pour utiliser des langages de script supplémentaires, reportez-vous à la section Activation de langages de script supplémentaires.

Lorsque vous spécifiez un protocole de données Scriptable (prenant en charge les scripts) côté demande, la fenêtre suivante s'affiche.



Deux scripts sont disponibles côté réponse ; l'un pour l'enregistrement, l'autre pour la lecture. Vous pouvez utiliser le script Response - Recording (Enregistrement de la réponse) pour modifier le format de la réponse enregistrée de sorte que le VSE puisse le traiter. Puis, une fois traité, vous pouvez utiliser le script Response - Playback (Lecture de la réponse) pour le renvoyer au format attendu par le système testé.



Vous pouvez ajouter votre propre script pour effectuer les actions de votre choix sur la demande et/ou la réponse.

Protocole de données SWIFT

Le gestionnaire de protocole de données SWIFT effectue les actions suivantes :

- Conversion d'un message SWIFT en une représentation XML
- Conversion de la représentation XML d'un message SWIFT à son format natif

Par exemple, le protocole de données convertit les messages SWIFT suivants :

```
{1:F01BANKDEFMAXXX2039063581}{2:01031609050901BANKDEFXAXXX89549829458949811609N}{3:{108:00750532785315}}{4:
```

```
:16R:GENL
```

```
:20C::SEME//YOUR REFERENCE
```

```
:16S:GENL
```

```
:16R:SETDET
```

```
:22F::SETR//TRAD
```

```
:16R:SETPRTY
```

```
:97A::SAFE//YYYY
```

```
:16S:SETPRTY
```

```
:16S:SETDET
```

```
-}
```

jusqu'au XML suivant :

```
<message>
<block1>
  <applicationId>F</applicationId><serviceId>01</serviceId>
  <logicalTerminal>BANKDEFMAXXX</logicalTerminal>
  <sessionNumber>2039</sessionNumber>
  <sequenceNumber>063581</sequenceNumber>
</block1>
<block2 type=\"output\">
  <messageType>103</messageType>
  <senderInputTime>1609</senderInputTime>
  <MIRDate>050901</MIRDate>
  <MIRLogicalTerminal>BANKDEFXAXXX</MIRLogicalTerminal>
  <MIRSessionNumber>8954</MIRSessionNumber>
  <MIRSequenceNumber>982945</MIRSequenceNumber>
  <receiverOutputDate>894981</receiverOutputDate>
  <receiverOutputTime>1609</receiverOutputTime>
  <messagePriority>N</messagePriority>
</block2>
```



```

<block3>
  <BLOCK3_108>00750532785315</BLOCK3_108>
</block3>
<block4>
  <GENL>
    <GENL_20C>:SEME//YOUR REFERENCE</GENL_20C>
  </GENL>
  <SETDET>
    <SETDET_22F>:SETR//TRAD</SETDET_22F>
    <SETPRTY>
      <SETPRTY_97A>:SAFE//YYYY</SETPRTY_97A>
    </SETPRTY>
  </SETDET>
</block4>
</message>

```

Si certains champs dans le message SWIFT incluent une date, le protocole de données reformate la date selon un format que DevTest peut identifier comme candidat de date magique.

L'exemple suivant présente les champs que le protocole de données vérifie pour les dates :

```

:98A::SETT//19911130
:98C::TRAD//20140117125901
:98E::PREP//20091107093238,02/N0230
:32A:870902JPY3520000,
:30:640123

```

Le protocole de données convertit ces lignes au format XML :

```

<BLOCK4_98A>:SETT//1991-11-30</BLOCK4_98A>
<BLOCK4_98C>:TRAD//2014-01-17 12:59:01</BLOCK4_98C>
<BLOCK4_98E>:PREP//2009-11-07T09:32:38.020-0230</BLOCK4_98E>
<BLOCK4_32A>1987-09-02 JPY3520000,</BLOCK4_32A>
<BLOCK4_30>2064-01-23</BLOCK4_30>

```

L'image de service représente les dates qu'il contient sous la forme de dates magiques.

SWIFT Conversations (conversations cumulées)

Pour générer une clé de session pour des transactions SWIFT, le protocole de données SWIFT extrait des informations du corps du message. Le protocole utilise les règles suivantes pour extraire la référence de message SWIFT (`mesg_ref`) et la référence d'accord (`deal_ref`), puis les combine pour former la clé de session.

1. Si un champ 70E est détecté au format **:SPRO///xxxREF/<mesg_ref>/<deal_ref>**, les références sont extraites à partir de ce champ.
2. Si un champ 20C est détecté au format **:RELA//<mesg_ref>** et un deuxième champ 20C est détecté au format **:TRRF//<deal_ref>**, les références sont extraites à partir de ces champs.
3. Si un champ 26H est détecté au format **<msg_ref>/<deal_ref>**, les références sont extraites à partir de ce champ.

Le protocole de données applique ces règles dans l'ordre. Si l'une d'entre elle est satisfaite, la clé de session est créée au format **<message ref>/<deal ref>**. Si aucune règle n'est satisfaite, le protocole de données traite la transaction comme étant sans état et ne crée pas de clé de session.

Protocole de données Web Services Bridge

Le protocole de données Web Services Bridge a été spécifiquement créé pour l'exemple de DevTest Travel. Il s'agit d'un protocole propre à l'exemple et n'est pas utile pour un cas général. En dehors de l'exemple de DevTest Travel, vous pouvez ignorer ce protocole.

Web Services (SOAP) (Services Web (SOAP))

Le gestionnaire de protocoles de données de services Web (SOAP) est utilisé pour convertir un document SOAP au format XML en un type d'opération/arguments approprié à la demande.

Une tentative d'analyse du format texte du corps de demande comme document XML est effectuée pour chaque demande présentée au protocole de données. Si le corps est un document XML valide et que l'élément de niveau supérieur est **Envelope (Enveloppe)**, le protocole recherche les éléments enfants **Header (En-tête)** et **Body (Corps)**.

Si l'enveloppe SOAP contient un élément d'en-tête, le protocole de données recherche un élément **ReplyTo (Réponse)** dans l'en-tête. Puis, sous cet élément, le protocole de données recherche un élément **Address (Adresse)**. Si cet élément est présent, sa valeur sera définie dans la liste des métadonnées pour la demande actuelle du VSE sous le nom **lisa.vse.reply.to**.

Si l'enveloppe SOAP contient un élément de corps, le nom de balise de son premier enfant deviendra le nom d'opération pour la demande de VSE. Si pour une raison quelconque, vous ne pouvez pas déterminer l'opération, aucune des actions suivantes ne se produira pour la demande actuelle.

Si l'élément d'opération contient des attributs XML, ils seront ajoutés à la liste des attributs pour la demande actuelle du VSE.

Puis, l'arborescence complète d'éléments XML sous l'élément d'opération sera examinée. Tous les éléments sans enfants seront ajoutés à la demande du VSE comme arguments. Pour assurer le caractère unique de chaque argument, leur nom est défini à l'aide de toutes les balises des éléments parents (jusqu'à l'élément d'opération). Si le même nom apparaît plusieurs fois, un suffixe numérique sera ajouté. Le suffixe numérique indique une structure de tableau et assure le caractère unique de l'argument spécifique.

Enfin, le document XML d'origine (le corps de demande) est copié dans un nouvel attribut de la demande du VSE nommée **recorded_raw_request**.

Ce gestionnaire de protocoles ne requiert aucune information de configuration et ne présente donc aucune fenêtre dans l'assistant d'enregistrement.

Web Services (SOAP Headers) (Services Web (en-têtes SOAP))

Le gestionnaire de protocoles de données SOAP Headers (En-têtes SOAP) convertit des éléments de l'en-tête d'un message SOAP en arguments pour les demandes dans une image de service virtuel. Ce protocole de données est compatible avec un protocole de transport qui génère des messages SOAP ; en règle générale, le protocole HTTP/S.

Pour utiliser le protocole de données SOAP Headers (En-têtes SOAP), générez une image de service virtuel via un enregistrement ou à partir de paires réponse-demande. En règle générale, vous utilisez le protocole de transport HTTP/S. Ajoutez le protocole de données SOAP Headers (En-têtes SOAP) comme protocole de données côté demande. Ce protocole de données ne requiert aucune configuration supplémentaire.

Vous pouvez l'utiliser avec le protocole de données SOAP pour traiter les en-têtes SOAP et le corps SOAP.

Les arguments sont nommés en fonction de la structure du XML.

Exemple

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
      </wsse:Security>
    <n1:ServiceControl xmlns:n1="http://localhost:8080/examples.xsd">
      <n1:VersionID>2.0</n1:VersionID>
      <n1:Asynchronous>
      <n1:ReplyRequiredIndicator>false</n1:ReplyRequiredIndicator>
      <n1:PassThroughData>
        <n1:Key>InteractionID</n1:Key>
        <n1:Value>444831</n1:Value>
      </n1:PassThroughData>
      </n1:Asynchronous>
    </n1:ServiceControl>
  </soapenv:Header>
  ...
```

Cet exemple serait analysé dans des éléments nommés comme suit :

- Security_UsernameToken_Username
- Security_UsernameToken_Password
- ServiceControl_VersionID

- ServiceControl_Asynchronous_ReplyRequiredIndicator
- ServiceControl_Asynchronous_PassThroughData_Key
- ServiceControl_Asynchronous_PassThroughData_Value

Tout élément dupliqué est nommé avec le suffixe _1, _2, etc.

Protocole de données WS-Security Request (Demande WS-Security)

Le protocole de données WS-Security Request (Demande WS-Security) prend en charge les messages SOAP qui incluent des en-têtes WS-Security. Ce protocole de données peut éliminer tout type de sécurité à partir de la demande SOAP avant de l'envoyer vers la structure de virtualisation. Le protocole de données WS-Security Request (Demande WS-Security) applique alors la sécurité aux réponses SOAP sortantes.

Lors de l'enregistrement d'un service Web avec des en-têtes WS-Security, ajoutez un protocole de données WS-Security Request (Demande WS-Security) côté demande (généralement, avant un protocole de données de services Web SOAP) et un protocole de données WS-Security Response (Réponse WS-Security) côté réponse.

Avant l'enregistrement, un ensemble de panneaux de configuration est présenté.

- Un pour le protocole de données WS-Security Request (Demande WS-Security)
- Deux pour le protocole de données WS-Security Response (Réponse WS-Security)

Protocole de données Request (Demande)

Pour le protocole de données Request (Demande), configurez le gestionnaire de sorte à traiter un message de demande envoyé par le client. Remplissez les actions Receive (Réception) utilisées pour décoder et valider les en-têtes. Cette configuration est utilisée pour l'enregistrement et la lecture.

Les options disponibles pour les messages Receive (Response) (Réception, réponse) sont les suivantes:

- Decryption (Déchiffrement)
- Signature Verification (Vérification de la signature)
- SAML Verifier (Vérificateur SAML)
- Username Token Verifier (Vérificateur de jetons de nom d'utilisateur)
- Timestamp Receipt (Réception d'horodatage)
- Signature Confirmation (Confirmation de signature)

Les options disponibles pour les messages Send (Request) (Envoi, demande) sont les suivantes:

- Timestamp (Horodatage)
- Username Token (Jeton de nom d'utilisateur)
- SAML Token (Jeton SAML)
- Signature Token (Jeton de signature)
- Chiffrement

Pour utiliser le chiffrement, entrez les informations suivantes :

Keystore file (Fichier du référentiel de clés)

Dans la liste déroulante ou dans le système de fichiers, sélectionnez le fichier du référentiel de clés à utiliser pour le chiffrement.

Keystore Type (Type de référentiel de clés)

- Sélectionnez Java Key Store (Référentiel de clés Java) ou Personal Information Exchange (PVKS #12) (Echange d'informations personnelles (PVKS #12)).

Keystore password (Mot de passe du fichier de référentiel de clés)

Spécifie le mot de passe associé au fichier de référentiel de clés sélectionné.

Keystore alias (Alias du référentiel de clés)

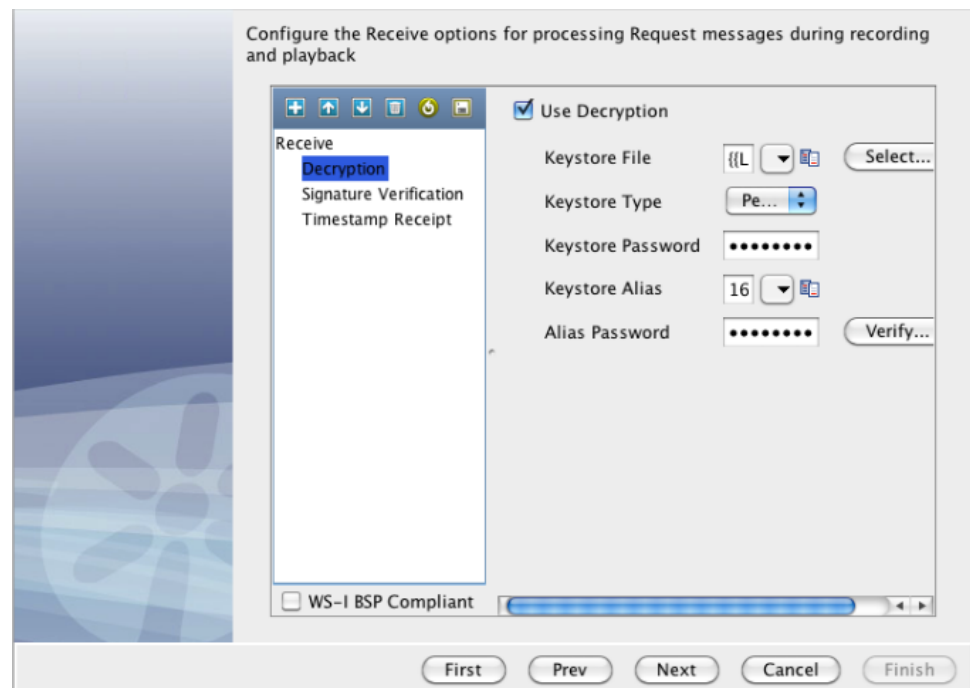
Indique un alias d'une clé publique.

Alias Password (Mot de passe d'alias)

Laissez ce champ vide ou saisissez la même valeur que Keystore Password (Mot de passe du référentiel de clés) dans les fichiers PKCS #12.

La case à cocher WS-I BSP Compliant (Conforme à WS-I BSP) indique si la conformité au WS-I Basic Security Profile (Profil de sécurité de base WS-I) doit être vérifiée, y compris à l'aide d'espaces de nom inclusifs et de la méthode de canonicalisation dans SignedInfo.

Pour valider vos informations de référentiel de clés, cliquez sur le bouton Verify (Vérifier).

**Protocole de données Response (Réponse)**

Dans le cas du protocole de données Response (Réponse), configurez le gestionnaire pour traiter les messages de réponse renvoyés à partir du service dynamique lors de l'enregistrement et les messages de réponse renvoyés à partir du modèle de service virtuel. Lors de la phase d'enregistrement, vous devez traiter le message de réponse de la même manière que le client.

Cochez la case Add Timestamp (Ajouter un horodatage).

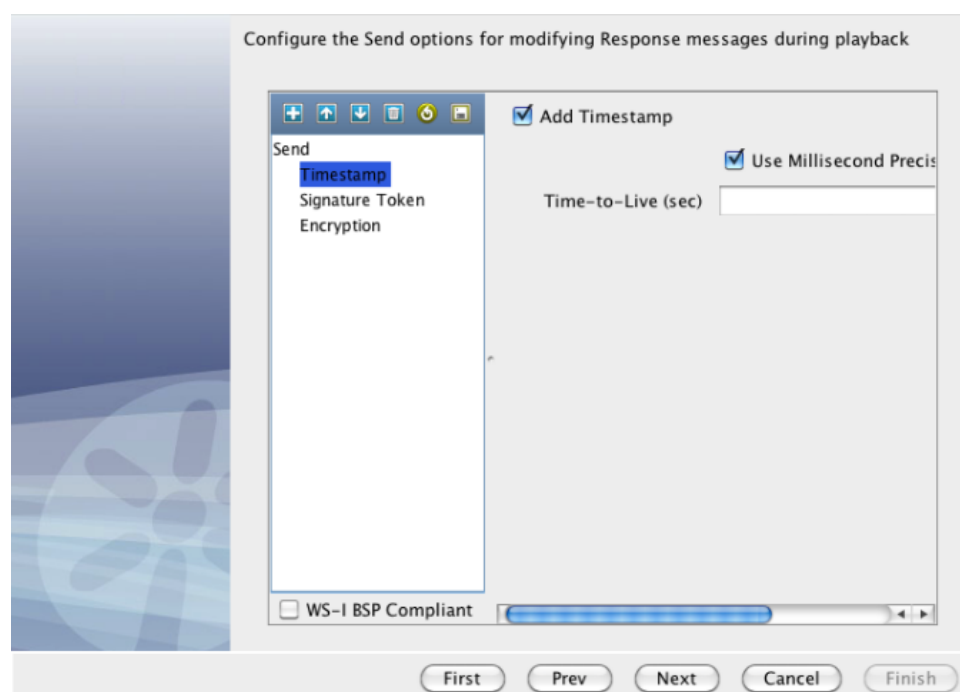
Time-To-Live (sec) (Durée de vie (en secondes))

Spécifie la durée de vie du message en secondes. Pour éviter l'inclusion d'un élément Expires (Expire), saisissez **0**.

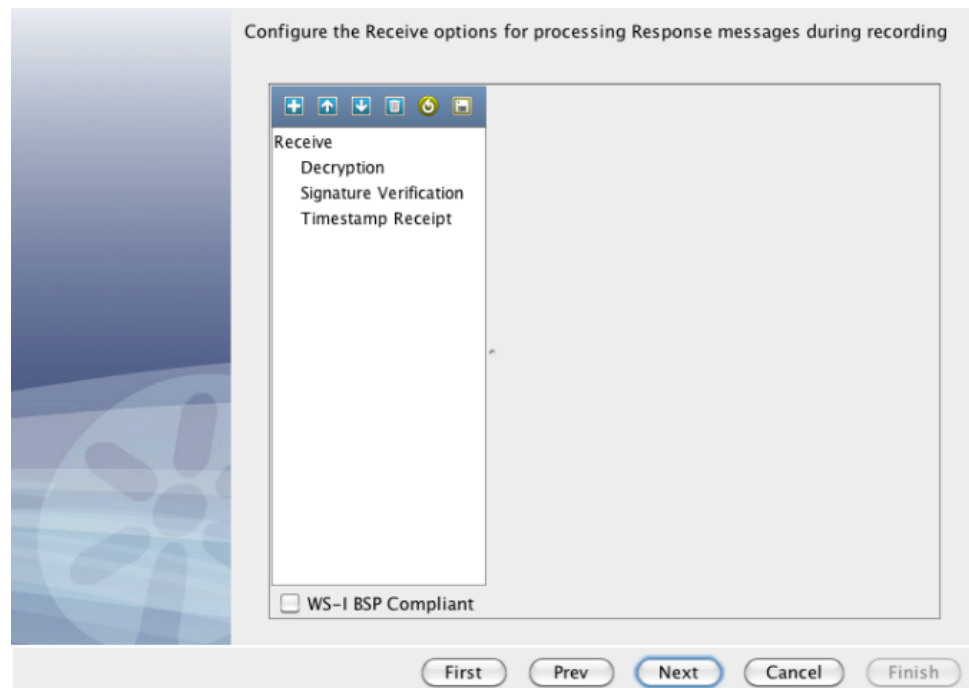
Use Millisecond Precision in Timestamp (Utiliser la précision à la milliseconde dans l'horodatage)

Spécifie si l'horodatage doit être indiqué en millisecondes.

Remarque : Certains services Web, notamment .NET 1.x/2.0 avec WSE 2.0, ne sont pas conformes au formatage de date de l'horodatage standard et ne permettent pas l'indication des millisecondes. Pour ces services Web, désélectionnez la case à cocher Use Millisecond Precision in Timestamp (Utiliser la précision à la milliseconde dans l'horodatage).

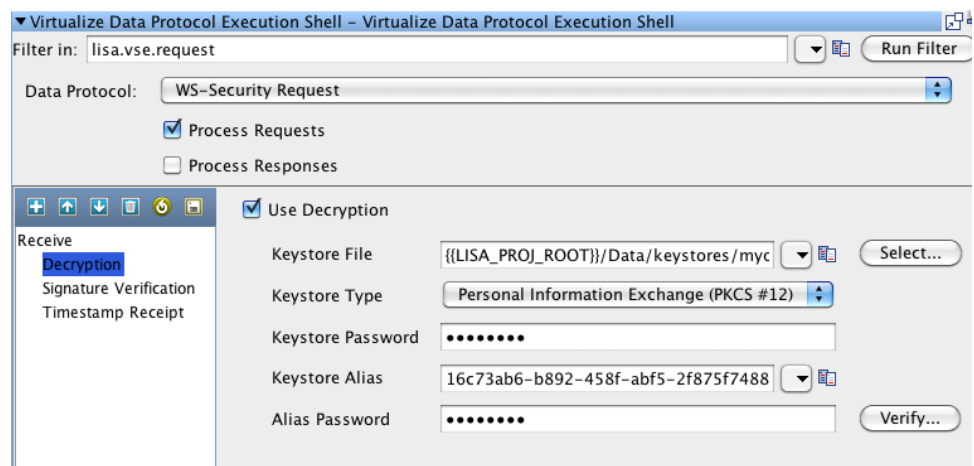


Pendant la lecture, vous devez traiter le message de la même manière que le serveur. Le message SOAP à partir du modèle de service virtuel ne contient aucun en-tête de sécurité tandis que cette configuration applique l'en-tête de sécurité.





A l'issue de l'enregistrement, un modèle de service virtuel est créé. Dans ce modèle, un filtre de protocole de données est ajouté à l'étape HTTP/S Listen (Écoute via HTTP/S) du protocole de données WS-Security Request (Demande WS-Security).

Mettez à jour toutes les informations de configuration de la sécurité pour la lecture. Par exemple, si vos paramètres WS-Security sont modifiés dans le service, vous pouvez les mettre à jour à ce stade au lieu de réenregistrer le service virtuel.



Dans le modèle de service virtuel, un filtre est également ajouté au protocole de données WS-Security Response (Réponse WS-Security) à l'étape HTTP/S Response (Réponse via HTTP/S).

Vous pouvez mettre à jour toutes les informations de configuration de la sécurité à lire pour le message de réponse.

Pour enregistrer vos paramètres de sécurité dans un fichier, ou charger un fichier enregistré contenant des paramètres de sécurité, utilisez les icônes Load (Charger)  et Save (Enregistrer). 

XML

Le gestionnaire de protocoles de données XML est utilisé pour convertir un document XML en un type d'opération/arguments approprié à la demande.

Ce gestionnaire de protocoles de données fonctionne exactement comme le protocole de données [Web Services \(Services Web \(SOAP\)\)](#) (page 287). Ce protocole de données ne requiert aucune information de configuration et ne présente donc aucune fenêtre dans l'assistant d'enregistrement.

Chapitre 9: Modification d'images de service

Ce chapitre traite des sujets suivants :

[Images de service héritées](#) (page 295)

[Ouverture d'une image de service à modifier](#) (page 296)

[Onglet Service Image \(Image de service\)](#) (page 297)

[Onglet Transactions](#) (page 301)

[Onglet Transactions pour les transactions sans état](#) (page 302)

[Onglet transactions pour les conversations](#) (page 315)

[Editeur de conversations](#) (page 317)

[Images de service pour le protocole de transport JMS](#) (page 328)

Images de service héritées

A partir de la version 6.0 de LISA, les images de service ne sont plus stockées dans une base de données. Si vous devez utiliser des images de service de LISA 5.0 dans LISA 6.0 ou une version ultérieure, exportez-les à l'aide de LISA 5.0. Puis, importez-les à l'aide de l'élément d'importation dans le menu contextuel de l'arborescence de projets dans la version ultérieure de LISA.

Remarque : L'extension d'une image de service LISA 5.0 exportée est **.xml**. Pour modifier cette image de service de 5.x exportée en image de service valide pour LISA 6.0 ou version ultérieure, remplacez l'extension par **.vsi**.

Ouverture d'une image de service à modifier

L'enregistreur d'image de service virtuel génère des images de service. Les images de service ont pour but de représenter vos enregistrements (une version manipulée ou modifiée du trafic brut enregistré).

Si vous disposez d'images de service de versions de VSE antérieures à LISA 6.0, exportez-les. L'exportation permet de les déplacer de la base de données de versions antérieures vers le système de fichiers de la version actuelle dans lequel elles sont stockées. Pour plus d'informations, consultez la rubrique [Images de service héritées](#) (page 295).

L'ouverture d'une image de service permet de modifier l'image dans l'éditeur d'images de service.

Procédez comme suit:

1. Dans le panneau Project (Projet), cliquez avec le bouton droit de la souris sur l'image de service et sélectionnez Open (Ouvrir).
L'éditeur d'images de service s'ouvre.
2. Vérifiez l'image de service sélectionnée et apportez les modifications nécessaires.

Remarque : Vous pouvez également accéder à l'éditeur à partir de l'étape Response Selection (Sélection d'une réponse) dans le modèle de service virtuel en cliquant sur Open (Ouvrir) à côté du nom de l'image de service.

Onglet Service Image (Image de service)

Les champs suivants sont disponibles dans l'onglet Service Image (Image de service) de l'éditeur d'images de service :

Image name (Nom de l'image)

Indique le nom de l'image de service actuelle.

Created On (Date de création)

Indique la date et l'heure de création de l'image de service.

Last modified (Dernière modification)

Indique la date et l'heure de la dernière modification de l'image de service.

Remarques

Documentation sur l'image de service.

Approximate memory usage (Utilisation approximative de la mémoire)

Estimation de la quantité de mémoire requise pour l'image de service

Response for Unknown Conversational Request (Réponse à la demande conversationnelle inconnue)

Donne des détails sur le corps, les métadonnées et le délai de réflexion pour une réponse à une demande conversationnelle inconnue pendant la lecture.


Response for Unknown Stateless Request (Réponse à la demande sans état inconnue)

Donne des détails sur le corps, les métadonnées et le délai de réflexion pour une réponse à une demande sans état inconnue pendant la lecture.

Pour plus d'informations sur les éléments spécifiques de cette fenêtre, reportez-vous aux sections suivantes :

- [Modification de réponses à des demandes inconnues](#) (page 298)
- [Personnalisation de l'éditeur de réponses](#) (page 299)

Modification de réponses à des demandes inconnues

Utilisez l'icône de flèche  pour agrandir les panneaux à leur taille maximum. Pour rétablir la taille d'origine, utilisez la même icône.

Dans les panneaux Response (Réponse), remplissez les champs suivants, si nécessaire :

Body (Corps)

Contient la réponse à renvoyer pour des demandes sans état inconnues pendant la lecture.

Response Meta Data (Métadonnées de la réponse)

Dans la zone Response Meta Data (Métadonnées de la réponse), utilisez la barre d'outils située au bas pour ajouter, déplacer, ou supprimer des paires clé-valeur si nécessaire.

Think time spec (Spécification du délai de réflexion)

Définit la durée du délai de réflexion requis (en millisecondes). Le *délai de réflexion* est le temps passé avant l'envoi de la réponse à une demande.

Valeurs : un nombre ou une plage, en millisecondes.

Si vous saisissez une plage, le délai de réflexion sera aléatoirement sélectionné dans cette plage.

Vous pouvez spécifier des mesures de temps en ajoutant un suffixe aux nombres. La casse est sans importance. Les suffixes valides sont les suivantes :

- **t** : millisecondes
- **s** : secondes
- **m** : minutes
- **h** : heures


Valeur par défaut : 0

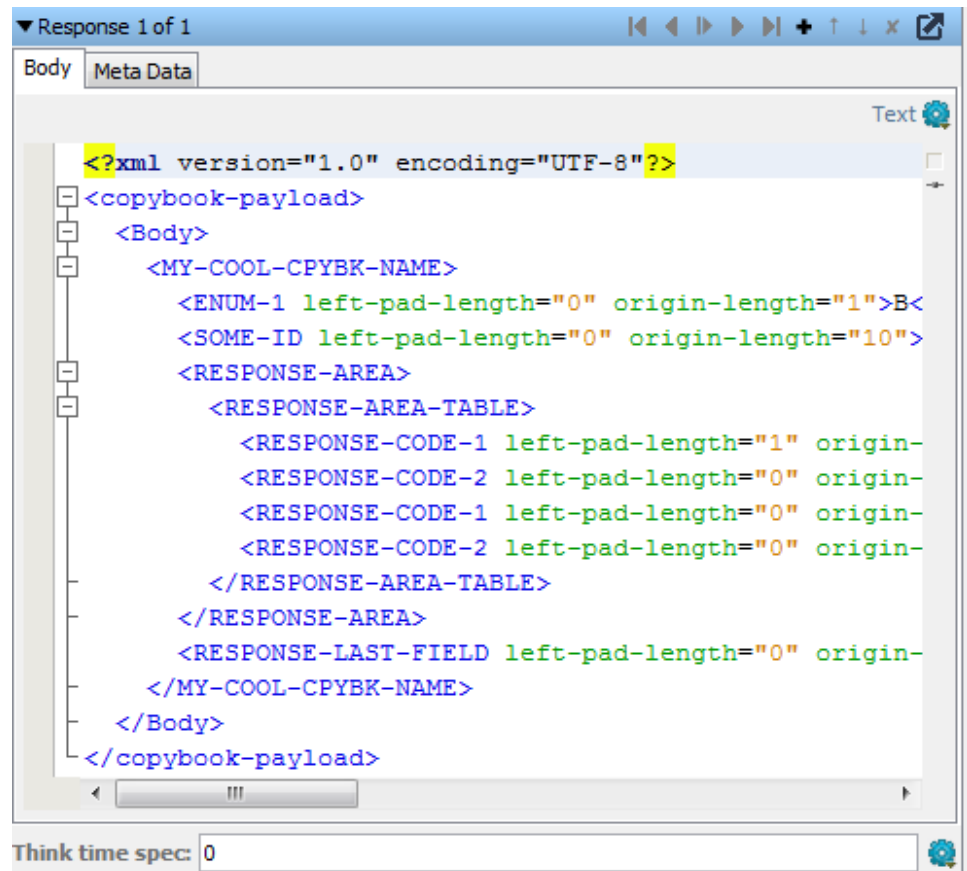
Exemples :

- 100 indique un délai de réflexion de 100 millisecondes.
- 100s indique un délai de réflexion de 100 secondes.
- 100-1000 spécifie un délai de réflexion aléatoire compris entre 100 et 1 000 millisecondes.
- 10t-5s indique un délai de réflexion aléatoire compris entre 10 millisecondes et 5 secondes.

Remarque : Une étape déduit sa propre durée de traitement du délai de réflexion pour obtenir une fréquence d'exécutions des tests cohérente.

Personnalisation de l'éditeur de réponses

Pour personnaliser l'éditeur de réponses, utilisez l'icône d'engrenage  située dans le coin inférieur droit du panneau.



Les options du menu de l'éditeur de réponses sont les suivantes :

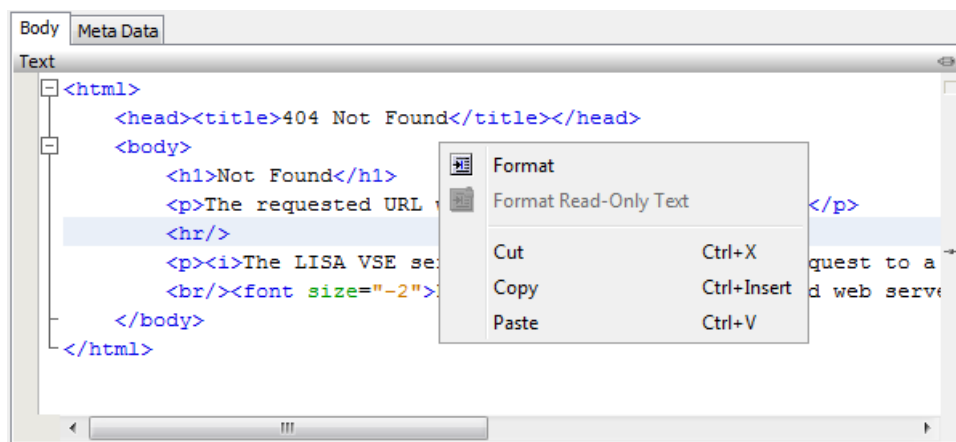
Set reference protocol to (Définir le protocole de référence sur)

Aucun protocole spécifique ou JDBC (basé sur le pilote)

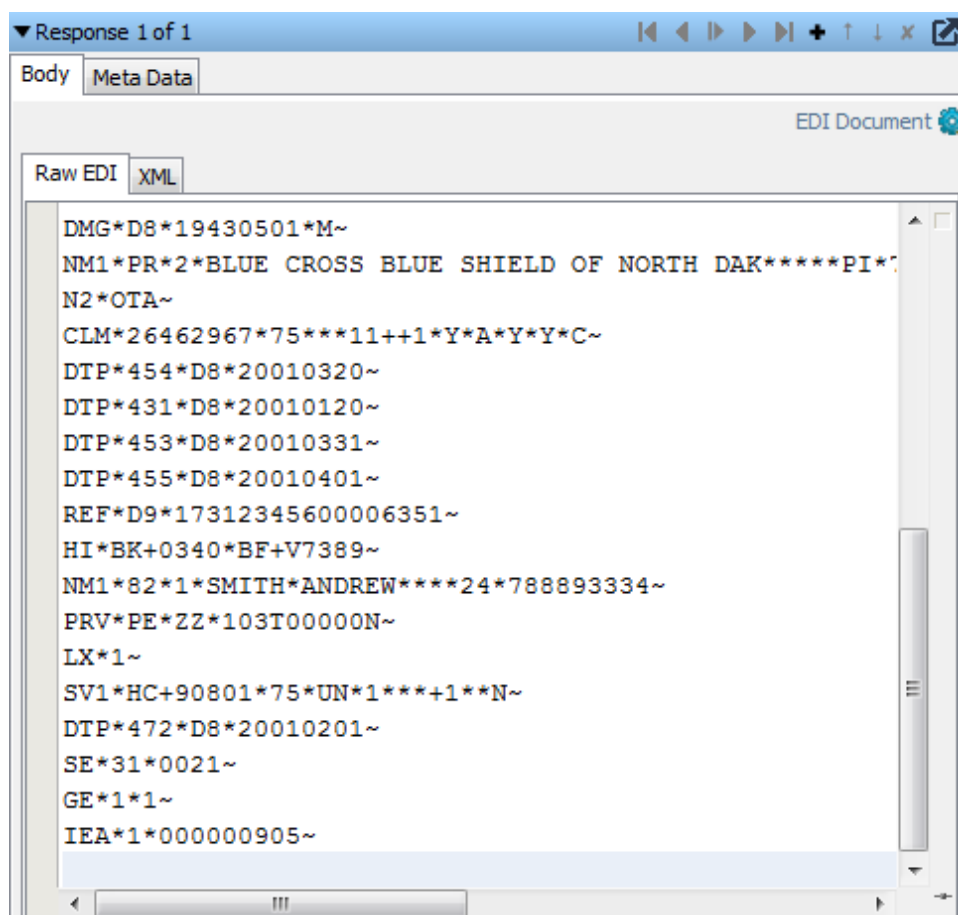
La barre de titre de l'onglet Body (Corps) de l'éditeur de réponses affiche la catégorie de la réponse renvoyée. Dans l'image précédente, elle indique une réponse Text. D'autres catégories de charges utiles de réponse sont XML, JSON, String (Chaîne), Large String (Chaîne longue) XML, Huge String (Chaîne très longue), Graphic Image (Image graphique) et Raw Bytes (Octets bruts).

Pour utiliser un éditeur différent ou modifier le type de charge utile, sélectionnez l'icône d'engrenage située dans le coin supérieur droit du panneau. Le menu est créé de manière dynamique de sorte à afficher uniquement d'autres éditeurs potentiellement valides en fonction de l'éditeur actuellement sélectionné.

Lorsque l'éditeur de corps de texte est défini sur Text, cliquez avec le bouton droit de la souris sur Response Body XML (Corps XML de la réponse) pour formater le texte de réponse.



Si la charge utile de réponse est détectée comme données EDI, elle sera convertie au format XML dans l'onglet XML.



Onglet Transactions

Vous pouvez accéder à l'onglet Transactions à partir de l'éditeur d'images de service. Cet onglet fournit des informations sur les transactions sans état et avec état (conversations), ainsi que des composants légèrement différents pour chaque type. Cette section décrit l'affichage des composants généraux de l'onglet Transactions qui sont identiques pour les deux types de transactions.

Pour sélectionner l'affichage des conversations ou des transactions sans état, sélectionnez Conversation by number (Conversation par numéro) ou Stateless Transactions (Transactions sans état) dans la liste déroulante.

Informations complémentaires :

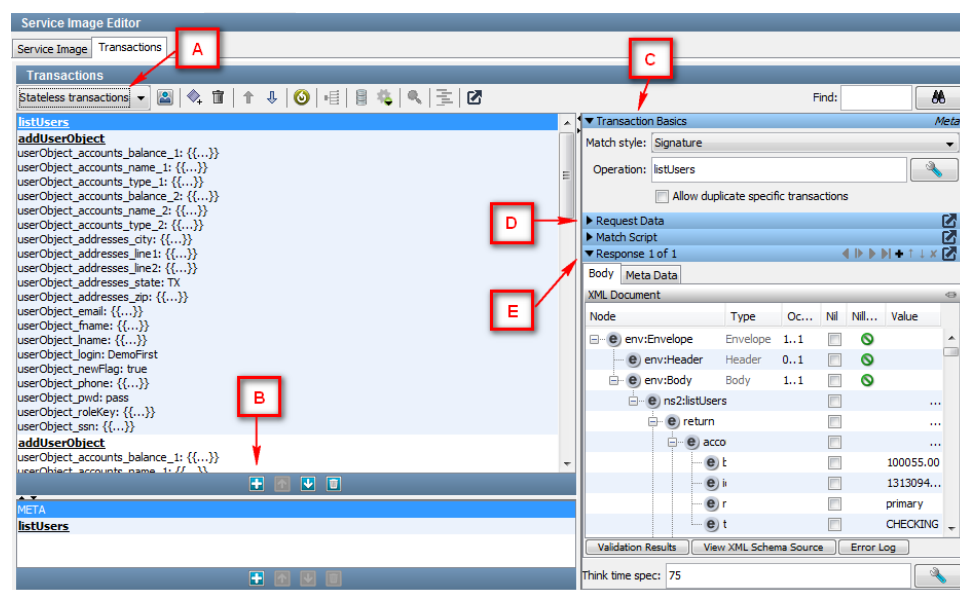
[Onglet Transactions pour les transactions sans état](#) (page 302)

[Onglet transactions pour les conversations](#) (page 315)

[Editeur de conversations](#) (page 317)

Onglet Transactions pour les transactions sans état

Lors de l'affichage d'une transaction sans état, les composants illustrés dans l'image suivante sont affichés.



- **A** : pour afficher et modifier des transactions sans état, utilisez la liste Stateless Transactions (Transactions sans état). Pour ajouter, déplacer, ou supprimer des transactions sans état, utilisez la barre d'outils située au bas du volet.
- **B** : une transaction logique (dans la liste Stateless Transactions (Transactions sans état)) contient exactement une métatransaction et un nombre de transactions spécifiques. La liste Transactions affiche ces transactions sous la transaction logique. Pour ajouter, déplacer, ou supprimer des transactions sans état, utilisez la barre d'outils située au bas du volet.
- **C** : pour afficher et modifier des demandes de transaction et des données de réponse pour des transactions spécifiques ou des métatransactions, qui sont sélectionnées dans la zone Transactions, utilisez la zone Transaction Basics (Concepts fondamentaux relatifs aux transactions). Vous pouvez sélectionner un style de correspondance pour la transaction concernée à ce stade.
- **D** : le panneau Request Data (Données de la demande) affiche les demandes sans état.
- **E** : le panneau Response (Réponse) affiche la réponse aux demandes sans état.

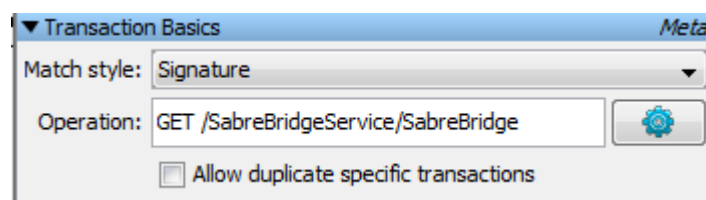
Remarque : Si vous ajoutez ou modifiez plusieurs transactions, cliquez sur Regenerate Magic Strings and Data Variables (Régénérer les variables des dates et des chaînes magiques). Les variables des chaînes et des dates magiques sont créées pour vous. Les chaînes et variables magiques existantes ne sont pas modifiées.

Pour plus d'informations sur les éléments spécifiques de cette fenêtre, reportez-vous aux sections suivantes :

- [Editeur de concepts fondamentaux relatifs aux transactions](#) (page 303)
- [Editeur de données de la demande](#) (page 304)
- [Editeur de scripts de correspondance](#) (page 309)
- [Barre d'outils de l'éditeur de scripts de correspondance](#) (page 312)
- [Editeur de données de réponse](#) (page 313)

Editeur de concepts fondamentaux relatifs aux transactions

Pour afficher et modifier des données de transaction pour des transactions spécifiques ou des métatransactions, utilisez l'éditeur de Transaction Basics (Concepts fondamentaux relatifs aux transactions). Sélectionnez une transaction spécifique ou META dans la liste Transactions.



L'éditeur Transaction Basics (Concepts fondamentaux relatifs aux transactions) permet de spécifier les informations suivantes :

Match Style (Style de correspondance)

- Les options valides sont **signature** ou **operation**.

Operation

Indique l'opération sélectionnée.

Allow duplicate specific transactions (Permettre la duplication des transactions)

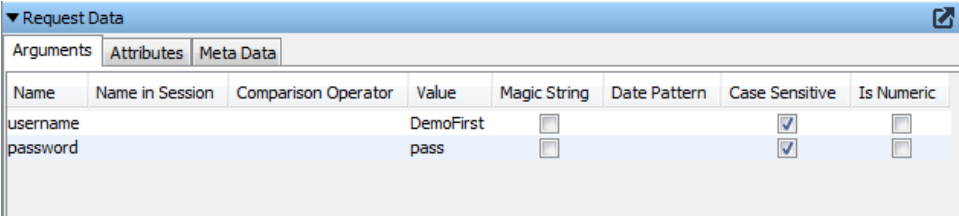
Indique si DevTest peut répondre plusieurs fois au même appel tout en sélectionnant une réponse différente.

Valeurs :

- **Option sélectionnée** : DevTest peut répondre plusieurs fois au même appel avec des réponses différentes. La correspondance de tourniquet se produit uniquement si cette case à cocher est sélectionnée.
- **Option désélectionnée** : DevTest peut répondre une seule fois à un appel spécifique.

Editeur de données de la demande

Le panneau de l'éditeur de données de la demande vous permet de mettre à jour les données associées à une demande.



Name	Name in Session	Comparison Operator	Value	Magic String	Date Pattern	Case Sensitive	Is Numeric
username			DemoFirst	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>
password			pass	<input type="checkbox"/>		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Arguments

VSE utilise le nom d'opération et des arguments pour rechercher une réponse correspondante pour une transaction entrante.

Ajout et suppression d'arguments

La transaction META est un modèle pour certaines transactions. Pour plus d'informations, consultez la rubrique [Transactions logiques](#). (page 62) Les arguments, par conséquent, peuvent être ajoutés et supprimés de la transaction META, et ces modifications s'appliquent à toutes les transactions appartenant à ce groupe logique. Les arguments peuvent ne pas être ajoutés directement à des transactions ou supprimés de celles-ci.

Modification des arguments

nom

Nom de l'argument analysé à partir de la demande. Dans la plupart des cas, ce champ ne doit pas être modifié. Vous pouvez uniquement le modifier au niveau de la transaction META et ces modifications se propagent aux transactions spécifiques.

Name in Session (Nom dans la session)

Valeur que l'application génère automatiquement lorsque les chaînes magiques sont identifiées. Vous pouvez référencer la valeur dans les réponses actuelles ou ultérieures à l'aide de la notation {{ }}. Si ce champ n'est pas vide, la valeur entrante est stockée dans la session à l'aide du nom spécifié. En général, cette valeur ne doit pas être modifiée.

Comparison Operator (Opérateur de comparaison)

Opérateur à utiliser dans la logique de correspondance. Par défaut, pour une transaction spécifique, tous les arguments doivent correspondre exactement. Pour modifier ce comportement, vous pouvez créer une logique de correspondance plus flexible en modifiant l'opérateur de comparaison. Pour en savoir plus sur les définitions d'opérateurs de comparaison, consultez la rubrique [Opérateurs de correspondance d'arguments](#) (page 64).

Magic string (chaîne magique)

Indique si la valeur spécifiée peut être utilisée pour les chaînes magiques.

Valeurs :

- **Option sélectionnée** : lorsque cette case à cocher est sélectionnée et que vous cliquez sur Regenerate magic strings (Régénérer les chaînes magiques) la valeur spécifiée peut être utilisée pour les chaînes magiques. Sélectionner cette option ne remplace pas les règles d'identification des chaînes magiques. Vous ne pouvez pas l'utiliser pour imposer l'utilisation d'une valeur comme chaîne magique, car elle dépend avant tout des propriétés de chaîne magique de VSE spécifiées dans le fichier **lisa.properties**.
- **Option désélectionnée** : lorsque cette case à cocher n'est pas sélectionnée et que vous cliquez sur Regenerate magic strings (Régénérer des chaînes magiques), la valeur spécifiée n'est pas utilisée pour les chaînes magiques. Si une valeur a été utilisée comme chaîne magique, mais que vous ne voulez pas que le remplacement de chaîne magique ait lieu, désélectionnez cette case à cocher et sélectionnez l'option Regenerate magic strings.

Date Pattern (Modèle de date)

Modèle selon lequel l'application interprète les valeurs entrantes et spécifiées en tant que dates. Cette valeur est automatiquement générée et ne doit pas être modifiée.

Ce modèle est un modèle de date et heure Java. Une interprétation stricte est appliquée, les valeurs doivent donc correspondre exactement au modèle. Si l'une des valeurs ou les deux ne peuvent pas être analysées en tant que date, l'argument ne renvoie aucune correspondance. Si les deux valeurs peuvent être analysées en tant que date, vous pouvez les comparer comme des dates à l'aide des opérateurs suivants :

- =
- !=
- <
- <=
- >
- >=

Vous pouvez utiliser les expressions Anything (Tout), Regular Expression (Expression régulière) et Property Expression (Expression de propriété). Si vous utilisez une expression régulière, la valeur entrante est considérée comme une chaîne.

Case sensitive (Respecter la casse)

Spécifie si la mise en correspondance doit respecter la casse.

Valeurs :

- **Option sélectionnée** : toutes les correspondances doivent respecter la casse.
- **Option désélectionnée** : les opérateurs de comparaison ignorent la casse.

Valeur par défaut : option sélectionnée

Is Numeric (Nombre)

Spécifie si les valeurs d'argument sont traitées comme des chaînes ou des nombres.

Valeurs :

- **Option sélectionnée :** les valeurs d'argument sont traitées comme des nombres.
- **Option désélectionnée :** les valeurs d'argument sont traitées comme des chaînes. Cela veut dire que 10000 est considéré comme une valeur inférieure à 9, car 1 est alphabétiquement antérieur à 9.

Valeur par défaut : option désélectionnée

Mass Change (Modification globale)

Pour effectuer des modifications globale d'arguments de demande, cliquez sur Mass



Change (Modifications globale). La boîte de dialogue Change Request Arguments (Arguments de la demande de modification) s'affiche.

Change Request Arguments

☒ If name equals

☐ If value equals

Set

☐ name to:

☐ name in session to:

☐ operator to: Anything

☐ value to:

☐ magic string to: unchecked

☐ date pattern to:

☐ case sensitive to: unchecked

☐ numeric string to: unchecked

Update Cancel

Pour spécifier des modifications globales, remplissez les champs de la boîte de dialogue Change Request Arguments, puis cliquez sur Update (Mettre à jour).

Attributes (Attributs)

Pour ajouter, modifier, déplacer et supprimer des paires clé-valeur, utilisez l'onglet Attributes (Attributs).

Meta Data (Métadonnées)

Pour ajouter, modifier, déplacer et supprimer des paires clé-valeur de métadonnées, utilisez l'onglet Meta Data (Métadonnées).

Editeur de scripts de correspondance

Pour insérer un exemple de script de correspondance pour vos informations, cliquez avec le bouton droit de la souris sur le panneau Match Script (Script de correspondance). De même, pour activer ou désactiver le script de correspondance, sélectionnez ou désélectionnez la case à cocher Do not use the script (Ne pas utiliser le script).

Pour désigner le langage de script, utilisez la liste déroulante des langages dans la partie inférieure droite du volet.

Langue

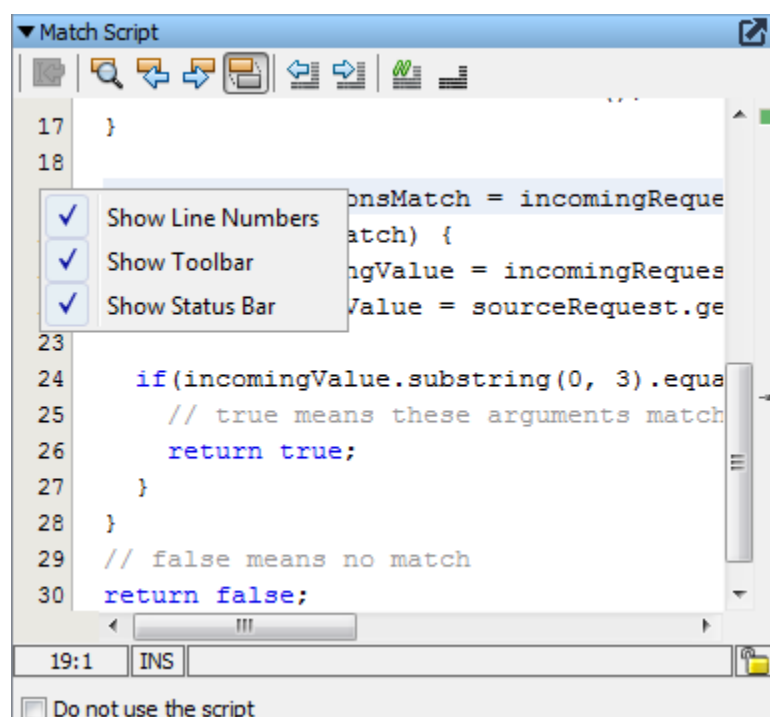
Désigne le langage de script à utiliser.

Valeurs :

- Applescript (OS X)
- Beanshell
- Freemarker
- Groovy
- JavaScript
- Velocity

Valeur par défaut : Beanshell

Pour masquer ou afficher les numéros de ligne, la barre d'outils de l'éditeur et la barre d'état d'éditeur, cliquez avec le bouton droit de la souris dans la partie gauche du panneau Match Script (Script de correspondance), puis sélectionnez les options appropriées dans le menu contextuel. Le graphique suivant présente toutes les options affichées.



Un script de correspondance définit la méthode utilisée par le VSE pour déterminer si une transaction donnée correspond à la transaction entrante. Pour renvoyer une correspondance basée sur la condition donnée, écrivez des scripts BeanShell qui effectuent des actions appropriées.

Par exemple :

```

/* always match name=joe */

ParameterList args = incomingRequest.getArguments();

if ("joe".equals(args.get("name"))) return true else return

defaultMatcher.matches();
  
```

Il n'est pas nécessaire de spécifier un niveau de tolérance de correspondance ou un opérateur de correspondance pour que le script de correspondance fonctionne. La correspondance est détectée en fonction de la condition dans le script de correspondance.

Par défaut (sans script de correspondance) une demande entrante est mise en correspondance avec une demande d'image de service, en comparant des opérations et/ou des arguments pour obtenir une réponse true/false à la question : Correspondent-elles ? Un script de correspondance remplace simplement cette logique cohérente et doit encore obtenir une réponse true/false à la question : Correspondent-elles ?

Le script peut utiliser la logique de correspondance par défaut. Dans le script, utilisez l'expression suivante `defaultMatcher.matches()`. Cette expression renvoie `true` ou `false` à l'aide de la logique de correspondance par défaut de VSE.

Le script de correspondance est similaire à une assertion mise en script. Il s'agit pour l'essentiel d'un script BeanShell standard, mais qui contient d'autres variables supplémentaires préchargées, ainsi que les propriétés habituelles et la variable `testExec` :

- `com.itko.lisa.vse.stateful.model.Request sourceRequest` (demande enregistrée)
- `com.itko.lisa.vse.stateful.model.Request incomingRequest` (demande dynamique entrante)
- `com.itko.lisa.vse.RequestMatcher defaultMatcher` (vous pouvez définir cette variable par défaut)

Renvoyez une valeur booléenne à partir du script ; `true` signifie qu'une correspondance a été détectée.

En cas d'erreur lors de l'évaluation du script, le VSE ignore délibérément l'erreur et applique par défaut la logique de correspondance standard. Pour confirmer que votre script a été exécuté, consultez le fichier journal du VSE.

L'intégration d'appels à l'enregistreur de correspondance du VSE est une méthode appropriée pour ajouter la journalisation et le suivi à vos scripts de correspondance. L'enregistreur de correspondance du VSE produit les messages dans le fichier **vse_xxx.log**, où `xxx` est le nom de l'image de service. Par exemple :

```
import com.itko.lisa.VSE;

VSE.info(testExec, "short msg", "a longer message");

VSE.debug(testExec, "", "I got here!!");

VSE.error(testExec, "Error!", "Some unexpected condition");

return defaultMatcher.matches();
```

Si vous journalisez des messages au niveau *INFO*, lors de l'application ultérieure des paramètres de production au fichier `logging.properties`, le niveau de journalisation sera *WARN* et vos messages apparaîtront sous forme d'événement de test DevTest (événement Log Message (Message de journal)).

Astuces du fichier **logging.properties** :










- Pour simplifier le débogage, définissez un journal différent pour les événements de correspondance/non-correspondance de transactions du VSE.
- Remplacez *INFO* par *WARN*, ou mettez la ligne suivante en commentaire pour les systèmes de production :
`log4j.logger.VSE=INFO, VSEAPP`

- En général, le niveau INFO signale chaque échec de correspondance.

Barre d'outils de l'éditeur de scripts de correspondance





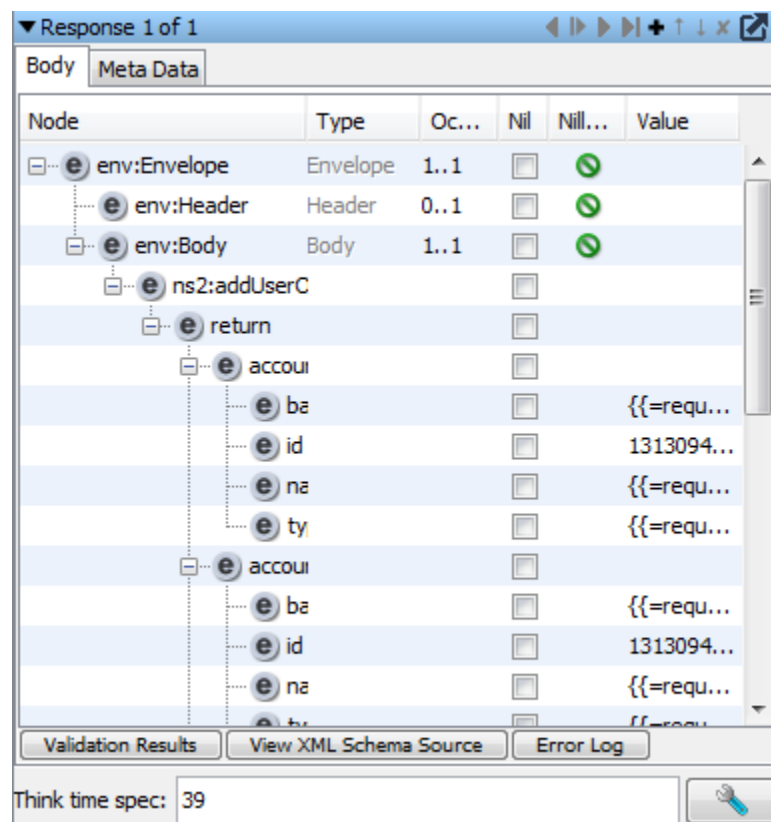
La barre d'outils de l'éditeur de scripts de correspondance permet d'effectuer les fonctions suivantes :

	Vous renvoie vers la dernière modification effectuée.
	Trouve l'occurrence suivante du texte mis en surbrillance.
	Trouve l'occurrence précédente.
	Trouve l'occurrence suivante.
	Active la mise en surbrillance de la recherche.
	Déplace la ligne actuelle de quatre espaces vers la gauche.
	Déplace la ligne actuelle de quatre espaces vers la droite.
	Insère des barres obliques de commentaires (//) à l'emplacement du curseur.
	Supprime les barres obliques de commentaires (//).

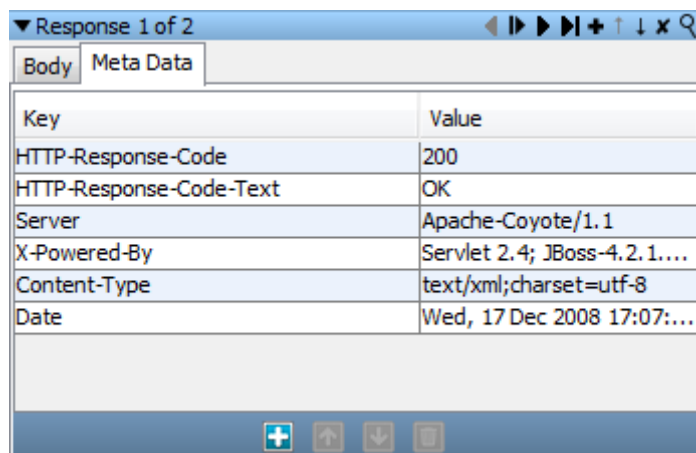
Editeur de données de réponse

Pour afficher et modifier les informations de réponse, utilisez l'éditeur de données de réponse.

- Pour modifier la réponse attendue pour une transaction, utilisez la zone Response Body (Corps de la réponse).
- Pour ajouter, classer, supprimer et relier des réponses en chaîne, utilisez la barre d'outils, comme indiqué dans la section [Barre d'outils de l'éditeur de scripts de correspondance](#) (page 312).
- Pour agrandir ce panneau, utilisez l'icône de flèche. 
- Pour personnaliser l'éditeur de réponse, utilisez l'icône en forme de clé à molette , comme indiqué dans la section [Personnalisation de l'éditeur de réponses](#) (page 299).
- Pour examiner les résultats de la validation, afficher la source de schéma XML et consulter le journal d'erreurs, utilisez les boutons situés au bas du panneau.



Modifiez la Think time spec (Spécification du délai de réflexion) si nécessaire.



▼ Response 1 of 2

Body Meta Data

Key	Value
HTTP-Response-Code	200
HTTP-Response-Code-Text	OK
Server	Apache-Coyote/1.1
X-Powered-By	Servlet 2.4; JBoss-4.2.1....
Content-Type	text/xml;charset=utf-8
Date	Wed, 17 Dec 2008 17:07:...

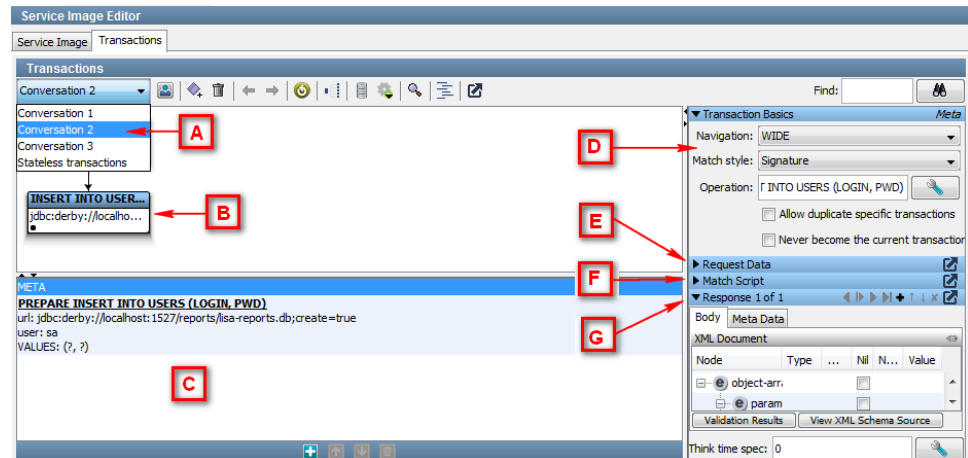
+

↑ ↓

Pour ajouter, modifier, déplacer et supprimer des paires clé-valeur, utilisez l'onglet Meta Data (Métadonnées).

Onglet transactions pour les conversations

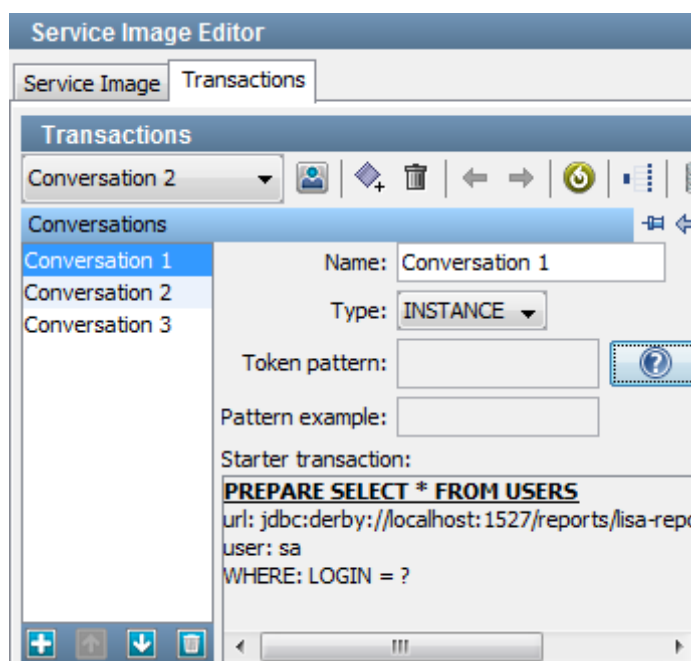
Une transaction avec état (transaction avec des conversations) est composée des éléments suivants.




- **A** : la liste Conversations contient toutes les conversations dans l'image de service. Pour afficher et modifier une conversation, sélectionnez-la. Une conversation comprend plusieurs transactions logiques.
- **B** : le [Conversation Tree editor \(Editeur d'arborescence des conversations\)](#) (page 321) affiche la conversation logique sélectionnée dans la liste Conversations. La conversation est affichée dans une arborescence de noeuds de graphique ou dans une arborescence standard.
- **C** : pour afficher et modifier des transactions spécifiques ou les métadonnées pour des transactions dans une transaction logique sélectionnée, utilisez la liste Transactions. Pour ajouter, déplacer, ou supprimer des transactions sans état, utilisez la barre d'outils située au bas du volet.
- **D** : pour afficher et modifier des demandes de transaction et des données de réponse pour des transactions spécifiques ou des métatransactions, qui sont sélectionnées à partir de la zone Transactions, utilisez la zone [Transaction Basics \(Concepts fondamentaux relatifs aux transactions\)](#) (page 303). Les champs varient selon les protocoles de transport et de données sélectionnés. Pour plus d'informations sur la zone Transaction Basics (Concepts fondamentaux relatifs aux transactions) et ses onglets, reportez-vous à la section [Onglet Transactions de l'éditeur d'images de service pour des transactions sans état](#) (page 302).
- **E** : pour entrer les données pour des demandes conversationnelles pendant la lecture, utilisez le volet [Transaction Request Data \(Données de la demande de transaction\)](#) (page 304).
- **F** : pour entrer et modifier un script pour renvoyer des actions basées sur des conditions de correspondance, utilisez le [Match Script Editor \(Editeur de scripts de correspondance\)](#) (page 309).

- **G** : pour afficher et modifier le contenu d'une réponse, le délai de réflexion et les paires clé-valeur pour les transactions spécifiques ou des métatransactions, utilisez le volet [Transaction Response Data](#) (page 302)(Données de la réponse de transaction).

Volet Toggle Display (Activer/désactiver l'affichage)



Vous pouvez afficher les détails d'une conversation en cliquant sur Display

(Activer/désactiver l'affichage)  dans la barre d'outils Transactions Tab (Onglet Transactions). Dans ce volet, vous pouvez afficher et modifier les valeurs suivantes.

- **Type** : le type est défini sur INSTANCE ou TOKEN.
- **Token Pattern (Modèle de jeton)** : requis pour les conversations basées sur un jeton. Cliquez sur l'icône en forme de point d'interrogation pour obtenir des exemples de modèles de générateur de chaînes.
- **Pattern Example (Exemple de modèle)** : exemple de modèle de jeton spécifié
- **Starter Transaction (Transaction de démarrage)** : transaction de démarrage pour la conversation

Remarque : Si vous ajoutez ou modifiez plusieurs transactions, revenez à l'onglet Basic Info (Informations de base) et cliquez sur Regenerate Magic Strings and Data Variables (Régénérer les variables des dates et des chaînes magiques). Les variables des chaînes et des dates magiques sont créées pour vous. Les chaînes et variables magiques existantes ne sont pas modifiées.

Editeur de conversations

Pour afficher et modifier des transactions enregistrées ou créer des transactions manuellement, utilisez le Conversation Editor (Editeur de conversations). Deux modes d'affichage sont disponibles pour les arborescences de navigation :

- [Graph View \(Vue de graphique\)](#) (page 319)
- [Tree View \(Vue d'arborescence\)](#) (page 321)

Lorsque vous basculez entre les vues, le noeud sélectionné reste sélectionné. Les deux vues permettent d'effectuer les actions suivantes à partir de la barre d'outils de l'éditeur, des menus contextuels, ou dans la vue même.

Barre d'outils de l'éditeur de conversations

La barre d'outils Conversation Tree editor (Editeur d'arborescence des conversations) varie légèrement, en fonction de l'affichage.

Outils de la vue de graphique














Outils de la vue d'arborescence




Remarque : Si un outil est grisé, vous ne pouvez pas l'utiliser avec la transaction sélectionnée.

La barre d'outils Conversation Editor (Editeur de conversations) contient les composants suivants :

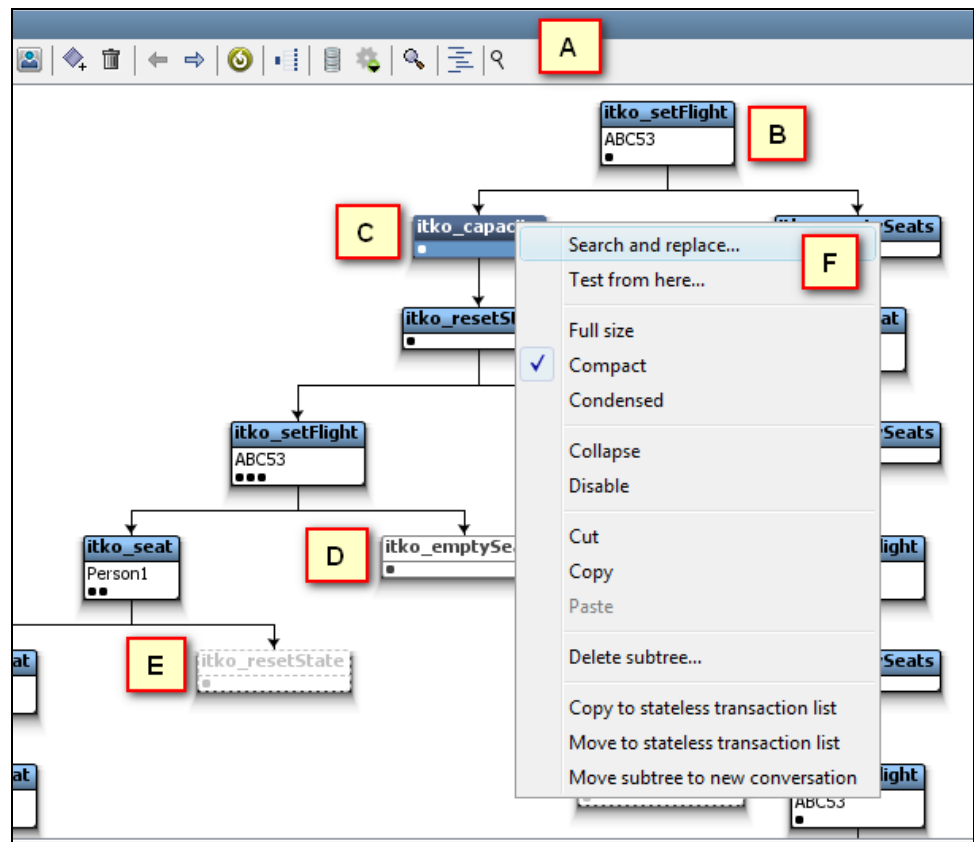
Outil	Icône	Description
Activer/désactiver l'affichage		Permet d'activer/désactiver l'affichage du panneau pour gérer une liste de conversations. Vous pouvez spécifier un nom, un type, un modèle de jeton, un exemple de modèle, ou une transaction de démarrage.
Créer une transaction		Permet d'ajouter une nouvelle transaction.
Supprimer la transaction sélectionnée		Permet de supprimer une transaction sélectionnée.

Flèche vers le haut		Vue d'arborescence : déplace le noeud sélectionné vers le haut de la liste des frères.
Flèche vers le bas		Vue d'arborescence : déplace le noeud sélectionné vers le bas de la liste des frères.
Flèche vers la droite		Vue de graphique : déplace le noeud sélectionné vers le haut de la liste des frères.
Flèche vers la gauche		Vue de graphique : déplace le noeud sélectionné vers le haut de la liste des frères.
Regenerate (Régénérer)		Permet de régénérer des variables de dates et de chaînes magiques pour toutes les transactions.
Affichage de la navigation		Ouvre un menu déroulant pour sélectionner des mises en surbrillance de navigation du menu pour des transactions avec état (conversations). Vous pouvez sélectionner les types d'informations suivants. <ul style="list-style-type: none"> ■ No navigation highlight (Aucune mise en surbrillance de la navigation) ■ Highlight on transaction's tolerance (Mise en surbrillance de la tolérance de la transaction) ■ Highlight as if close (Mise en surbrillance selon le niveau CLOSE) ■ Highlight as if wide (Mise en surbrillance selon le niveau WIDE) ■ Highlight as if loose (Mise en surbrillance selon le niveau LOOSE)
Activer/désactiver		Bascule l'affichage des ID de transaction pour le débogage.
Correspondance		Récupère une description de correspondance à partir du presse-papiers et met en surbrillance les informations pertinentes dans l'image de service.
Zoom		Ouvre un menu déroulant d'options de zoom.
Affichage		Active/désactive l'affichage de conversations dans une vue d'arborescence.
Affichage		Active/désactive l'affichage de conversations dans une vue de graphique.

Panneau de zoom		Agrandit ce panneau à sa taille maximum.
-----------------	---	--

Vue de graphique de l'éditeur de conversations

Dans l'éditeur de conversations, les noeuds sont affichés en fonction de leur statut.



Les composants de l'éditeur graphique des conversations et les thèmes des noeuds sont décrits à l'aide de lettres de légende de l'image dans le diagramme suivant :

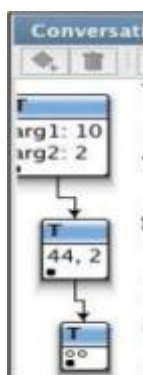
- **A** : barre d'outils. Pour plus d'informations, consultez la section [Barre d'outils de l'éditeur de conversations](#) (page 317).
- **B** : noeud standard
- **C** : noeud sélectionné
- **D** : noeud réduit Aucun noeud enfant n'est affiché. Pour afficher des noeuds enfants, développez le noeud.
- **E** : noeud désactivé. Ce noeud et tous les noeuds enfants (non affichés) sont ignorés pendant l'exécution.
- **F** : menu contextuel

Statut d'affichage des noeuds

Il existe trois styles d'affichage des noeuds :

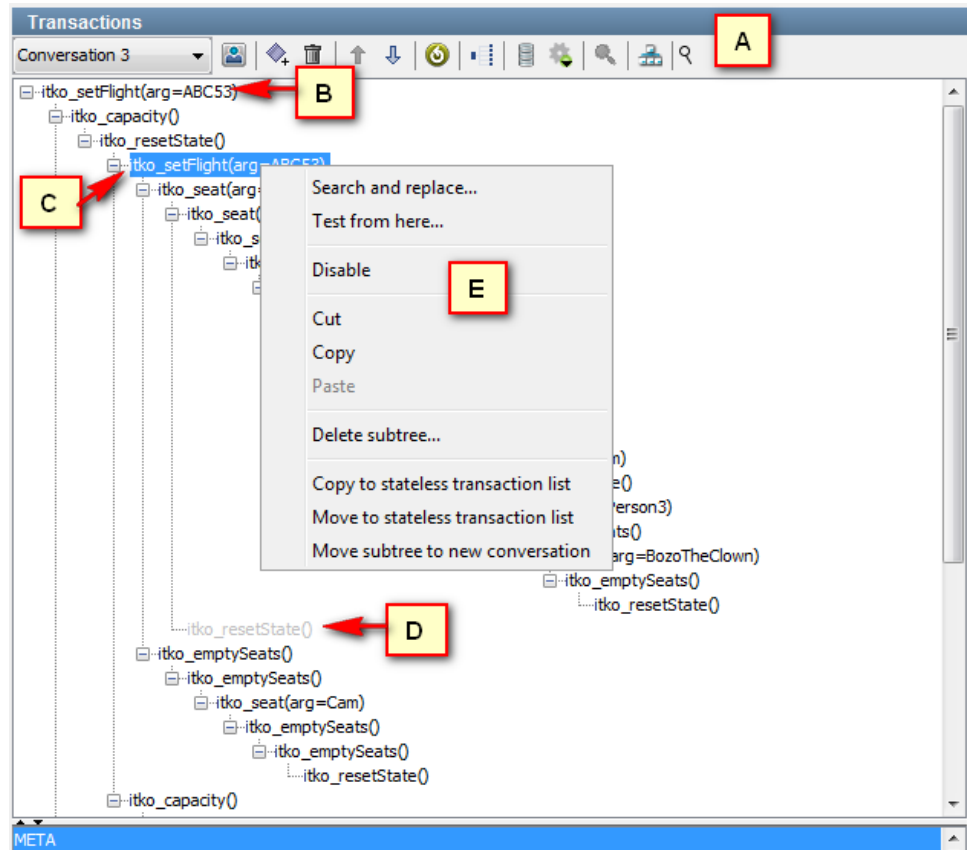
- Taille réelle
- Compact (valeur par défaut)
- Condensé

Dans les trois cas, la ligne de points noirs au bas de chaque noeud indique le nombre de transactions spécifiques appartenant au noeud. Dans le style condensé, les points estompés indiquent le nombre d'arguments pour la demande. Dans le graphique suivant, le premier noeud est affiché en taille réelle, le deuxième en style compact et le troisième en style condensé.



Vue d'arborescence de l'éditeur de conversations

La vue d'arborescence contient les mêmes informations que la vue de graphique, mais sous forme compacte. Elle affiche les noeuds en fonction du statut et comprend les composants suivants :



- **A** : barre d'outils. Pour plus d'informations, reportez-vous à la section [Barre d'outils de l'éditeur de conversations](#) (page 317).
- **B** : noeud standard
- **C** : noeud sélectionné
- **D** : noeud désactivé. Ce noeud et tous les noeuds enfants (non affichés) sont ignorés pendant l'exécution.
- **E** : menu contextuel

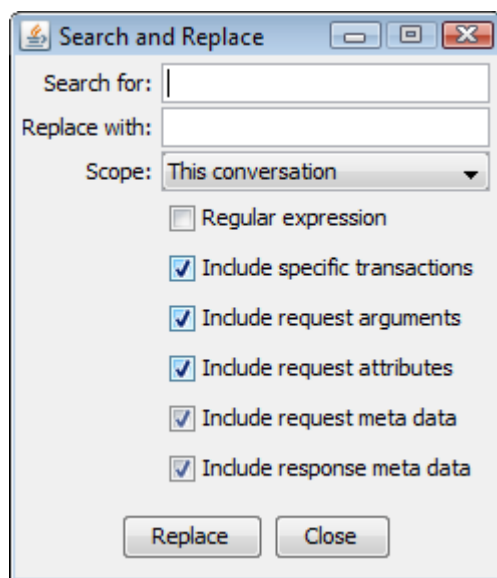
Pour plus d'informations sur les éléments spécifiques de cette fenêtre, reportez-vous aux sections suivantes :

- [Action Search and Replace \(Rechercher et remplacer\)](#) (page 322)
- [Action Test from Here \(Point de départ du test\)](#) (page 323)
- [Possibilités de mise en surbrillance de la navigation](#) (page 325)

- [Restructuration d'une conversation](#) (page 327)

Action Search and Replace (Rechercher et remplacer)

Dans le menu contextuel, vous pouvez sélectionner l'action Search and Replace (Rechercher et remplacer) pour modifier des valeurs dans la conversation ou dans la transaction.



Scope (Etendue) permet de spécifier l'élément auquel s'applique l'action de recherche et de remplacement :

- This conversation (Cette conversation, valeur par défaut)
- This transaction (Cette transaction)
- This transaction and children (Cette transaction et ses enfants)
- The entire service image (L'image de service complète)

Vous pouvez également utiliser les cases à cocher pour spécifier les parties des transactions à inclure.

Action Test from Here (Point de départ du test)

Pour analyser pour une réponse attendue à partir d'une transaction sélectionnée dans la vue de graphique et d'arborescence, utilisez l'éditeur de conversations.

Procédez comme suit:

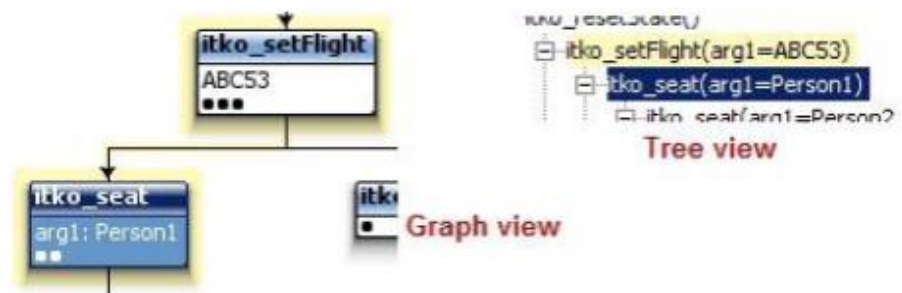
1. Dans le Conversation Tree editor (Editeur d'arborescence des conversations), cliquez sur une transaction avec le bouton droit de la souris.
2. Dans le menu contextuel, sélectionnez Test from here (Point de départ du test).
3. La fenêtre Create Test Request (Créer une demande de test) s'ouvre.

Key	Value
arg1	
arg2	

Key	Value
-----	-------

4. Saisissez le nom d'opération unique et ajoutez des paires clé-valeur d'argument, si nécessaire.
5. Cliquez sur Test.

Comme l'illustre le graphique suivant, le résultat apparaît en bleu avec le chemin d'accès en jaune.



Remarque : Si le test échoue, l'application affichera l'erreur suivante :

No transaction matching the request follows the selected one in this conversation (Aucune transaction correspondant à la demande ne suit celle sélectionnée dans cette conversation.)

6. Cliquez sur OK pour continuer.
7. Cliquez sur Close (Fermer).

Navigation mise en surbrillance

Pour afficher les possibilités de navigation dans une conversation en fonction de la tolérance de navigation sélectionnée, utilisez Conversation Tree (Arborescence des conversations).

Sélectionnez une transaction et cliquez sur le bouton View Navigation (Affichage de la navigation).

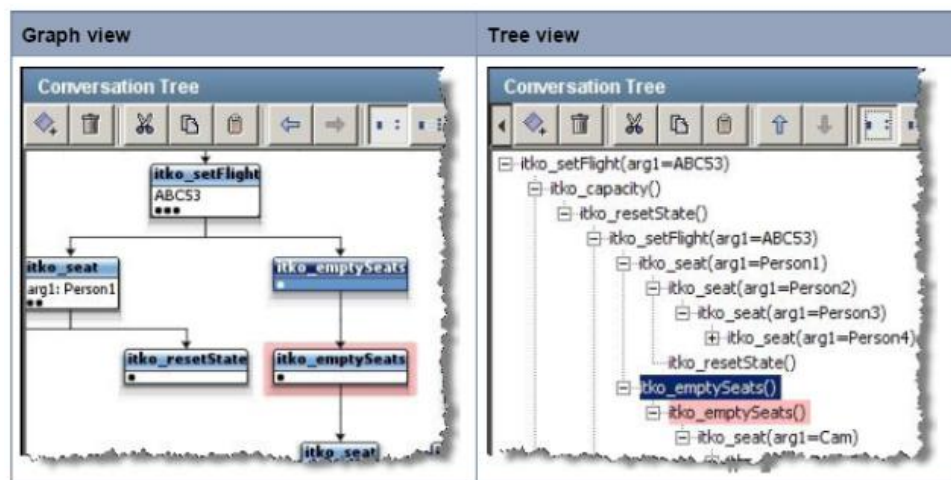
La sélection de menu par défaut est No navigation highlight (Aucune mise en surbrillance de la navigation), c'est-à-dire qu'aucune transaction n'est mise en surbrillance.

Si vous sélectionnez une transaction et que vous cliquez sur l'option Highlight on transaction's tolerance (Mise en surbrillance de la tolérance de la transaction), la transaction et ses transactions associées seront mises en surbrillance en fonction de la tolérance de navigation définie pour la transaction sélectionnée. Les options sont : CLOSE, WIDE, ou LOOSE. La section suivante décrit les résultats attendus pour chaque niveau de tolérance.

Remarque : Si vous utilisez la liste déroulante située dans le coin supérieur droit de l'éditeur pour changer la tolérance de navigation de la transaction actuelle, cette sélection remplacera celle du menu. Pour revenir à l'état correct de la tolérance, résélectionnez cette option de mise en surbrillance de navigation.

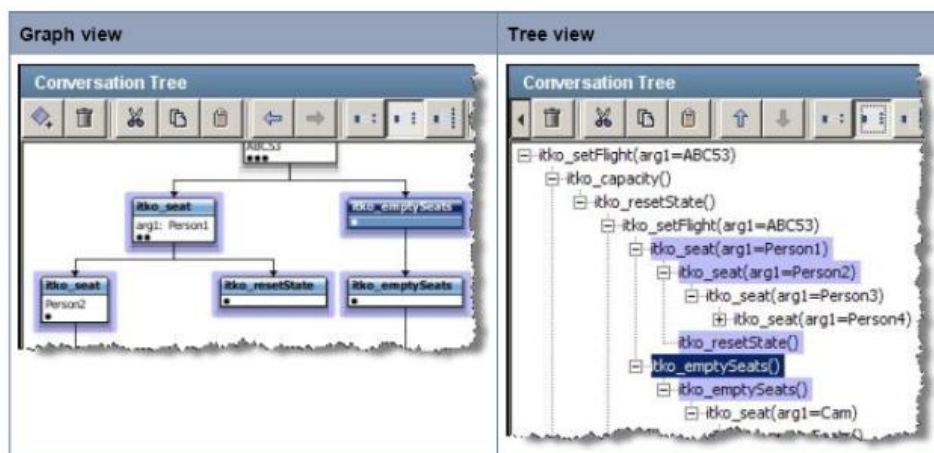
Affichage de la navigation CLOSE

Sélectionnez une transaction et cliquez sur le bouton View Navigation (Affichage de la navigation). Les transactions recherchées dans la sous-arborescence de la transaction sélectionnée sont mises en surbrillance en rouge.



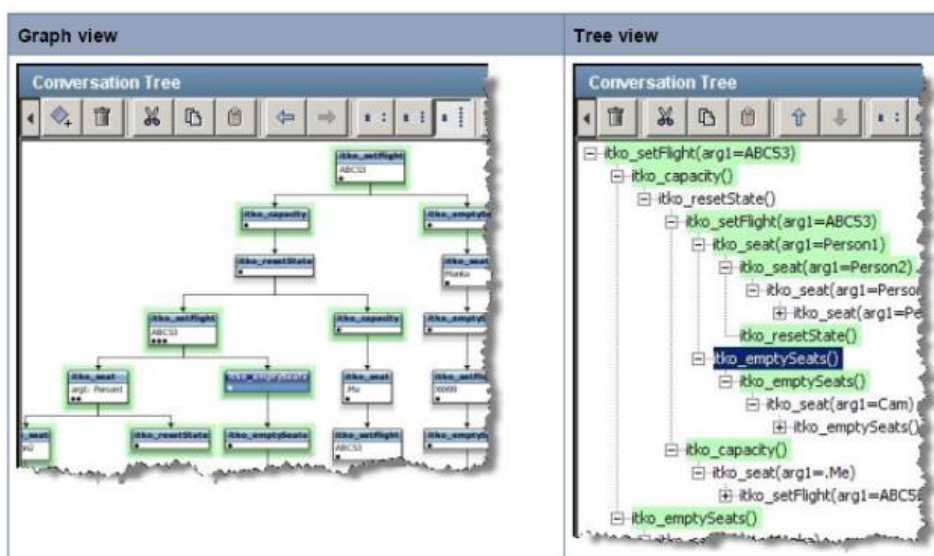
Affichage de la navigation WIDE

Sélectionnez une transaction et cliquez sur le bouton View Navigation (Affichage de la navigation). Les transactions recherchées dans la sous-arborescence de la transaction sélectionnée et les sous-arborescences de frères sont mises en surbrillance en bleu.



Affichage de la navigation LOOSE

Sélectionnez une transaction et cliquez sur le bouton View Navigation (Affichage de la navigation). Les transactions recherchées dans la sous-arborescence de la transaction sélectionnée et les sous-arborescences de frères et parents sont mises en surbrillance en vert. Il est également possible de redémarrer une conversation complète.

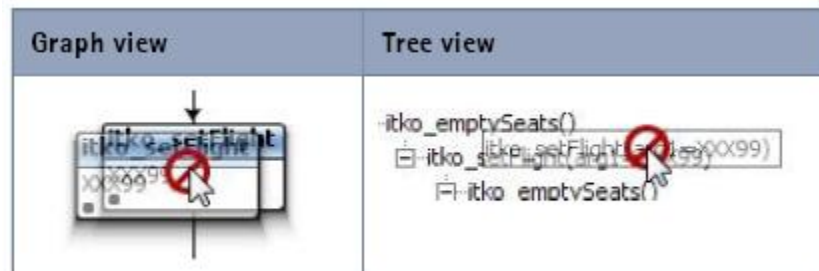


Restructuration d'une conversation

Dans certains cas, vous voudrez restructurer une conversation en déplaçant une transaction et sa sous-arborescence (le cas échéant) à l'aide de la fonction glisser-déposer. Vous pouvez effectuer la restructuration dans la vue de graphique comme dans celle d'arborescence.

Procédez comme suit:

1. Cliquez sur une transaction.



Lorsque vous faites glisser la transaction, le symbole universel indiquant Impossible s'affiche.

2. Faites glisser la transaction vers la transaction que vous voulez définir comme nouvelle transaction parente.

Si vous pouvez déposer la transaction, le symbole Impossible disparaît.



3. Déposez la transaction.

La transaction et toute transaction dans sa sous-arborescence sont déplacées sous le nouveau parent.

Images de service pour le protocole de transport JMS

Cette rubrique décrit les caractéristiques des images de service créées à l'aide du [protocole de transport JMS](#) (page 154).

Les éléments suivants s'appliquent côté demande :

- Par défaut, le nom d'opération pour la transaction est défini sur le nom du canal de demande défini dans l'étape de demande dans le modèle de service virtuel.
- Les propriétés de métadonnées qui ne sont pas groupées à l'aide d'un préfixe contiennent des informations de file d'attente et de connexion pour la file d'attente de demandes de proxy.
- Les propriétés de métadonnées qui commencent par **liveRequest** contiennent les informations de file d'attente et de connexion pour la file d'attente de demandes dynamiques.
- La propriété de métadonnées **channel.name** doit correspondre au nom de canal de demande défini dans la demande et les étapes d'appel dynamique du modèle de service virtuel. Il s'agit de la seule propriété de métadonnées requise côté demande. Vous pouvez modifier les files d'attente de demande du modèle de service virtuel si nécessaire, à condition de conserver le nom du canal de demande.

Les éléments suivants s'appliquent côté réponse :

- Les propriétés de métadonnées qui ne sont pas groupées à l'aide d'un préfixe contiennent informations de file d'attente et de connexion pour la file d'attente de réponses de proxy.
- Les propriétés de métadonnées qui commencent par **msg** sont utilisées pour reconstruire le message de réponse JMS et ses propriétés. La seule propriété requise est **msg.type**, qui indique le type de message à envoyer.
- La propriété de métadonnées **channel.name** doit correspondre à un nom de canal de réponse défini dans les étapes de réponse et d'appel dynamique dans le modèle de service virtuel. La valeur spécifie le nom du canal de réponse à utiliser pour envoyer la réponse au client. En cas de réponses multiples, elles peuvent spécifier des noms de canal différents.
- Les propriétés de métadonnées qui commencent par **liveResponse** contiennent les propriétés de file d'attente et de connexion pour la file d'attente de réponses dynamiques.

Chapitre 10: Modification d'un modèle de service virtuel

Un enregistrement d'image de service virtuel crée un modèle de service virtuel en six ou huit étapes, selon l'option sélectionnée : More Flexible (Flexible) ou More Efficient (Efficace). Vous pouvez parfois vouloir modifier le modèle de service virtuel en modifiant les étapes générées ou en y ajoutant d'autres étapes.

Vous pouvez ignorer cette section sauf si vous devez modifier un modèle de service virtuel ou créer un modèle de service virtuel sans enregistreur.

Un modèle de service virtuel est un scénario de test spécialisé et par conséquent la modification ou la création d'un modèle de service virtuel est similaire à la création ou la modification d'un scénario de test. De nombreux types d'étapes peuvent être ajoutés à un modèle de service virtuel. Vous pouvez ajouter une étape qui ne s'affiche pas dans ce menu ; toutefois, certains autres types d'étape peuvent empêcher le déploiement d'un modèle de service virtuel.

Accédez au menu des types d'étapes à partir de l'éditeur de modèles de service virtuel en sélectionnant Add a new step (Ajouter une nouvelle étape), Virtual Service Environment (Environnement de service virtuel) dans le menu.

Ce chapitre traite des sujets suivants :

[Etape Virtual Service Router \(Routeur de service virtuel\)](#) (page 331)
[Etape Virtual Service Tracker \(Outil de suivi de service virtuel\)](#) (page 332)
[Etape Virtual Conversational/Stateless Response Selector \(Sélecteur virtuel de réponse conversationnelle/sans état\)](#) (page 333)
[Etape Virtual HTTP/S Listener \(Ecouteur virtuel de DRDA\)](#) (page 334)
[Etape Virtual HTTP/S Live Invocation \(Appel dynamique DRDA virtuel\)](#) (page 336)
[Etape Virtual HTTP/S Responder \(Répondeur virtuel HTTP/S\)](#) (page 338)
[Etape Virtual JDBC Listener \(Ecouteur virtuel de JDBC\)](#) (page 339)
[Etape Virtual JDBC Responder \(Répondeur virtuel JDBC\)](#) (page 340)
[Etape Socket Server Emulator \(Emulateur de serveur de socket\)](#) (page 341)
[Etape Messaging Virtualization Marker \(Marqueur de virtualisation de messagerie\)](#) (page 343)
[Etape Compare Strings for Response Lookup \(Comparer les chaînes pour rechercher des réponses\)](#) (page 344)
[Etape Compare Strings for Next Step Lookup \(Comparer les chaînes pour rechercher l'étape suivante\)](#) (page 346)
[Etape Virtual Java Listener \(Ecouteur virtuel Java\)](#) (page 348)
[Etape Virtual Java Live Invocation \(Appel dynamique virtuel via Java\)](#) (page 350)
[Etape Virtual Java Responder \(Répondeur virtuel Java\)](#) (page 350)
[Etape Virtual TCP/IP Listener \(Ecouteur virtuel de DRDA\)](#) (page 351)
[Etape Virtual TCP/IP Live Invocation \(Appel dynamique DRDA virtuel\)](#) (page 353)
[Etape Virtual TCP/IP Responder \(Répondeur virtuel TCP/IP\)](#) (page 354)
[Etape Virtual CICS Listener \(Ecouteur virtuel CICS\)](#) (page 355)
[Etape Virtual CICS Responder \(Répondeur virtuel CICS\)](#) (page 355)
[Etape CICS Transaction Gateway Listener \(Ecouteur de la passerelle de transaction CICS\)](#) (page 356)
[Etape CICS Transaction Gateway Live Invocation \(Appel dynamique de la passerelle de transaction CICS\)](#) (page 358)

[Etape CICS Transaction Gateway Responder \(Répondeur de la passerelle de transaction CICS\)](#) (page 359)

[Etape Virtual DRDA Listener \(Ecouteur virtuel de DRDA\)](#) (page 360)

[Etape Virtual DRDA Response Builder \(Générateur de réponses virtuelles de DRDA\)](#) (page 360)

[Etape Virtual DRDA Live Invocation \(Appel dynamique DRDA virtuel\)](#) (page 361)

[Etape IMS Connect Listener \(Ecouteur d'IMS Connect\)](#) (page 362)

[Etape IMS Connect Live Invocation \(Appel dynamique IMS Connect\)](#) (page 363)

[Etape Virtual IMS Connect Responder \(Répondeur de connexion au service IMS virtuel\)](#) (page 364)

[Etapas JMS du VSE](#) (page 365)

[Etape JCo IDoc Listener \(Ecouteur de RFC via JCo\)](#) (page 370)

[Etape JCo IDoc Live Invocation \(Appel dynamique de fichier IDoc via JCo\)](#) (page 371)

[Etape JCo IDoc Responder \(Répondeur IDoc via JCo\)](#) (page 372)

[Etape JCo RFC Listener \(Ecouteur de RFC via JCo\)](#) (page 372)

[Etape JCo RFC Live Invocation \(Appel dynamique de RFC via JCo\)](#) (page 373)

[Etape JCo RFC Responder \(Répondeur RFC via JCo\)](#) (page 374)

Etape Virtual Service Router (Routeur de service virtuel)

Cette étape permet de router une demande à partir d'une étape d'écoute de service virtuel vers l'étape du sélecteur de réponse et/ou l'étape d'appel dynamique spécifique au protocole. La décision est basée sur le mode d'exécution actuel pour le modèle en cours d'exécution.

Procédez comme suit:

1. Remplissez les champs suivants, tel qu'indiqué :

Live invocation step (Etape d'appel dynamique)

Sélectionnez l'étape d'appel dynamique dans la liste.

If Environment Error (En cas d'erreur d'environnement)

Spécifie l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Valeur par défaut : Abort test (Interrompre le test)

When in dynamic mode, determine real mode using (Mode réel à utiliser en mode dynamique)

Sélectionnez Subprocess (Sous-processus) ou Script.

2. Cliquez sur Test pour tester vos paramètres pour l'étape.

Etape Virtual Service Tracker (Outil de suivi de service virtuel)

Cette étape est utilisée pour suivre les réponses dans un service virtuel en cours d'exécution et d'effectuer une validation au niveau d'un système dynamique (facultatif). Cette validation simplifie le débogage et la réparation de modèles de service.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Image Response (Réponse d'image)

Sélectionnez le fichier de réponse d'image.

Live Response (Réponse dynamique)

Sélectionnez le fichier de réponse dynamique.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Etape Virtual Conversational/Stateless Response Selector (Sélecteur virtuel de réponse conversationnelle/sans état)

Vous pouvez définir l'étape Virtual Conversational/Stateless Response Selector (Sélecteur virtuel de réponse conversationnelle/sans état) comme l'étape principale dans chaque modèle de service virtuel. Cette étape permet d'examiner une image de service spécifique et de sélectionner une réponse virtuelle appropriée pour une demande donnée. Lorsqu'une demande a plusieurs réponses, celles-ci sont toujours affichées sous forme de liste. L'étape est généralement créée via l'enregistrement et la virtualisation d'une forme de trafic de service.

Remplissez les champs suivants, tel qu'indiqué :

Service Image Location (Emplacement de l'image de service)

Dans la liste déroulante des images de service disponibles, sélectionnez-en une à associer à cette étape. Après avoir sélectionné une image de service ici, affichez-la ou modifiez-la en cliquant sur le bouton Open (Ouvrir) ; l'image s'ouvre dans un nouvel onglet.

Request property name (Nom de la propriété de demande)

Pour définir la propriété et rechercher la demande entrante, définissez le nom de la propriété. En règle générale, le nom de la propriété est la réponse de l'étape précédente.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape Virtual HTTP/S Listener (Ecouteur virtuel de DRDA)

Utilisez l'étape Virtual HTTP/S Listener (Ecouteur HTTPS/S virtuel) pour simuler un serveur HTTP, avec prise en charge SSL. Cette étape permet d'écouter les demandes HTTP entrantes et de les convertir au format de demande virtuelle standard.

Le nom par défaut de l'étape Virtual HTTP/S Listener (Ecouteur virtuel HTTPS) est **Virtual HTTPS Listener<numéro port>**. Vous pouvez renommer l'étape à tout moment.

Remplissez les champs suivants, tel qu'indiqué :

Listen port (Port d'écoute)

Saisissez le port sur lequel DevTest écoute le trafic HTTP/S.

Bind address (Adresse de liaison)

Saisissez l'adresse IP locale dans laquelle les connexions peuvent entrer. Si aucune adresse de liaison n'est spécifiée, l'étape de l'ecouteur acceptera des connexions sur le port spécifié quel que soit le NIC (ou l'adresse IP) sur lequel il entre.

Bind only (Liaison uniquement)

Pour acquérir la ressource réseau et passer à l'étape suivante, sélectionnez cette case à cocher. Une deuxième étape d'écoute sans utiliser l'option Bind only (Liaison uniquement) est requise. Cette option permet au modèle d'écouter sur un port (les demandes sont placées dans la file d'attente jusqu'à leur consommation par une étape d'écoute). Le modèle effectue des tâches d'installation avant d'être déplacé dans la boucle attente/traitement/réponse. Par exemple, l'étape 1 du modèle acquiert le port d'écoute (à l'aide de l'option de liaison uniquement) et l'étape 2 déclenche un logiciel externe qui envoie des demandes.

Use SSL to client (Utiliser une connexion SSL au client)

Sélectionnez cette case à cocher pour simuler un site Web HTTP/S sécurisé. Puis, fournissez les informations de référentiel de clés SSL.

SSL keystore file (Fichier de référentiel de clés SSL)

Pour rechercher votre fichier de référentiel de clés SSL, cliquez sur Select.. (Sélectionner). Le même fichier de référentiel de clés doit être disponible pour le serveur du VSE sur lequel le modèle de service virtuel est déployé.

Keystore password (Mot de passe du fichier de référentiel de clés)

Saisissez le mot de passe du référentiel de clés et cliquez sur Verify (Vérifier).

Base path (Chemin de base)

Identifiez les URI de ressource HTTP demandée que l'étape d'écoute doit traiter. Lors de l'entrée de la demande, un nom (chemin de base) qui démarre l'URI dans la demande est recherché dans la liste des noms de file d'attente. Le nom correspondant est celui de la file d'attente dans laquelle la demande est placée. L'étape d'écoute associée à la file d'attente (par chemin de base) traite la demande.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape Virtual HTTP/S Live Invocation (Appel dynamique DRDA virtuel)

Utilisez l'étape Virtual HTTP/S Live Invocation (Appel dynamique HTTP virtuel) pour effectuer un appel HTTP réel vers un serveur réel dans le contexte d'un service HTTP virtualisé. L'étape est généralement créée via l'enregistrement et la virtualisation d'une forme de trafic HTTP. L'étape effectue la demande réelle, qui est basée sur la demande du VSE actuelle en cours d'utilisation.

Le nom par défaut de l'étape Virtual HTTP-S Live Invocation (Appel dynamique HTTPS/S virtuel) est **Virtual HTTPS LiveInvocation**<numéro port >. Vous pouvez renommer l'étape à tout moment.

Remplissez les champs suivants, tel qu'indiqué :

Target server (Serveur cible)

Indique le nom du serveur vers lequel la demande est envoyée.

Target port (Port cible)

Indique le nom du port sur lequel la demande est envoyée.

Remplacement URI (URL de remplacement)

Pour remplacer l'URI complet dans une demande GET/POST, saisissez un nouveau champ de chemin cible. Vous pouvez fournir l'URI comme propriété DevTest. Ce champ peut être vide, auquel cas, l'URI de la demande dynamique sera utilisé.

Do not modify host header parameter received from client (Ne pas modifier le paramètre d'en-tête d'hôte reçu à partir du client)

Cette option indique à l'appel dynamique d'envoyer l'en-tête d'hôte reçu à partir de l'application cliente vers le serveur cible. Si elle n'est pas sélectionnée, l'appel dynamique régénère le paramètre d'en-tête d'hôte selon le format host: <hôte_cible>:<port_cible>.

Use SSL to server (Utiliser une connexion SSL au serveur)

Si cette option est sélectionnée, une demande HTTPS (secured layer) est envoyée au serveur.

SSL keystore file (Fichier de référentiel de clés SSL)

Pour rechercher votre fichier de référentiel de clés SSL, cliquez sur Select.. (Sélectionner). Le même fichier de référentiel de clés doit être disponible pour le serveur du VSE sur lequel le modèle de service virtuel est déployé.

Le premier certificat dans le référentiel de clés est présenté au serveur cible lorsque l'authentification du client est requise. Ce certificat remplace le certificat que les propriétés **ssl.client.*** dans le fichier **local.properties** spécifient pour l'appel dynamique.

Keystore password (Mot de passe du fichier de référentiel de clés)

Saisissez le mot de passe du référentiel de clés et cliquez sur Verify (Vérifier).

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

Bad Response Codes (Codes de réponse incorrecte)

Définit une réponse d'échec du système.

Format : liste séparée par des virgules de codes à trois caractères, chacun d'eux étant numérique ou la lettre x (caractère générique).

Exemple : 4xx,5xx

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Remarque : L'étape Virtual HTTP-S Live Invocation (Appel dynamique HTTP/S virtuel) prend en charge les propriétés **`lisa.http.timeout.socket`** et **`lisa.http.timeout.connection`** pour contrôler les sockets clients utilisés. La propriété **`lisa.vse.http.live.invocation.max.idle.socket`** contrôle la durée d'attente d'un socket client inactif avant de devenir obsolète. Cette durée est définie par défaut sur 2 minutes.

Etape Virtual HTTP/S Responder (Répondeur virtuel HTTP/S)

Utilisez cette étape avec l'étape Virtual HTTP/S Listener (Ecouteur HTTPS/S virtuel) pour transmettre des réponses à des demandes HTTP produites par l'écouteur. L'étape utilise une réponse virtuelle pour répondre à la demande correspondante à l'aide du protocole HTTP/S.

Vous pouvez créer cette étape via l'enregistrement et la virtualisation du trafic HTTP.

Remplissez les champs suivants, tel qu'indiqué :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Etape Virtual JDBC Listener (Ecouteur virtuel de JDBC)

Utilisez l'étape Virtual JDBC Listener (Ecouteur virtuel JDBC) pour contrôler la simulation du trafic de base de données JDBC. L'étape gère la communication avec le pilote de simulation intégré dans le client de base de données.

Le nom par défaut de l'étape Virtual JDBC Listener (Ecouteur virtuel JDBC) est **Virtual JDBC Listener<numéro port>**. Vous pouvez renommer l'étape à tout moment.

Remplissez les champs suivants, tel qu'indiqué :

Endpoint Information (Informations sur le terminal)

Permet de configurer l'hôte de simulation et la plage de ports (valeur par défaut : 2999) de manière adéquate. Le VSE pour JDBC prend en charge plusieurs terminaux pendant l'enregistrement et la lecture. Les informations sur le terminal sont présentées au format de table comprenant l'hôte de pilote, le port de base et le port maximum. Si le port de base et le port maximum sont différents, un terminal unique inclusif sera créé pour chaque port entre les deux.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Connect/Disconnect (Connecter/Déconnecter)

Pour vous connecter au simulateur JDBC, cliquez sur Connect (Connecter). Si vous êtes connecté, cliquez sur Disconnect (Déconnecter) pour mettre fin à la connexion. Ce bouton permet également de valider les informations de connexion.

Installed and Initialized JDBC Drivers (Pilotes JDBC installés et initialisés)

Indique les pilotes JDBC installés et initialisés dans le client de base de données.

Current SQL Activity (Activité SQL actuelle)

Indique l'activité SQL actuelle dans le client de base de données.

Etape Virtual JDBC Responder (Répondeur virtuel JDBC)

Cette étape permet de prendre le résultat d'un appel de données JDBC tel que sélectionné par une étape de sélection de réponse conversationnelle. Puis, l'étape envoie le résultat au pilote de simulation intégré dans le client de base de données.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Etape Socket Server Emulator (Emulateur de serveur de socket)

Utilisez cette étape pour simuler un socket de serveur basé sur du texte (généralement HTTP). L'étape Socket Server Emulator (Emulateur de serveur de socket) prend en charge l'écoute, la réponse et la liaison. L'étape Socket Server Emulator (Emulateur de serveur de socket) est au niveau inférieur. Si vous utilisez cette étape, vous devez vérifier que le bloc de texte envoyé par l'étape de réponse est entièrement compatible avec HTTP.

Remarque : Lorsque vous utilisez l'étape Socket Server Emulator (Emulateur de serveur de socket) en mode réponse, le texte de sortie **doit** prendre la forme d'un message de réponse HTTP valide.

Remplissez les champs suivants, tel qu'indiqué :

Process mode (Mode de traitement)

Dans la liste, sélectionnez le mode de traitement. Les options valides sont les suivantes :

- Traitement complet (valeur par défaut)
- Configuration asynchrone : obtenez la ressource réseau et passez à l'étape suivante.
- Listen Only (Ecoute uniquement)
- Respond Only (Réponse uniquement)

Listen port (Port d'écoute)

Saisissez le port sur lequel DevTest écoute le trafic HTTP/S.

Bind address (Adresse de liaison)

Saisissez l'adresse IP locale dans laquelle les connexions peuvent entrer. Si aucune adresse de liaison n'est spécifiée, l'étape de l'écouteur acceptera des connexions sur le port spécifié quel que soit le NIC (ou l'adresse IP) sur lequel il entre.

Close immediately (Fermer immédiatement)

Sélectionnez cette option pour utiliser le test de cette étape dans l'environnement de conception. Cette option indique à l'étape d'effectuer les tâches configurées, puis de supprimer immédiatement ses ressources réseau.

Use SSL (Utiliser une connexion SSL)

Sélectionnez cette case à cocher pour simuler un site Web HTTPS sécurisé. Puis, fournissez les informations de référentiel de clés SSL.

SSL keystore file (Fichier de référentiel de clés SSL)

Pour rechercher votre fichier de référentiel de clés SSL, cliquez sur Select.. (Sélectionner). Le même fichier de référentiel de clés doit être disponible pour le serveur du VSE sur lequel le modèle de service virtuel est déployé.

Keystore password (Mot de passe du fichier de référentiel de clés)

Saisissez le mot de passe du référentiel de clés et cliquez sur **Verify (Vérifier)**.

Base path (Chemin de base)

Identifiez les URI de ressource HTTP demandée que l'étape d'écoute doit traiter. Lors de l'entrée de la demande, un nom (chemin de base) qui démarre l'URI dans la demande est recherché dans la liste des noms de file d'attente. Le nom correspondant est celui de la file d'attente dans laquelle la demande est placée. L'étape d'écoute associée à la file d'attente (par chemin de base) traite la demande.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Record terminator (Caractère de fin de l'enregistrement)

Dans le cas d'un émulateur de socket simulant un service basé sur l'enregistrement, saisissez le caractère marquant la fin d'un enregistrement. Si vous laissez ce champ vide, les enregistrements basés sur la ligne ou le protocole HTTP seront simulés.

Ensure proper HTTP response format (Vérifier le format de la réponse HTTP)

Cette option est sélectionnée par défaut. Lorsqu'un mode de traitement envoie une réponse qui doit être une réponse HTTP valide, cette option permet de vérifier que le format des en-têtes HTTP dans le texte de la réponse est correct et, si nécessaire, que l'en-tête de réponse HTTP **Content-Length (Longueur du contenu)** est présent et correct. Cette case à cocher permet uniquement de vérifier que les séparateurs de ligne sont conformes à l'HTTP et que l'en-tête Content-Length (Longueur du contenu) est présent et correct. Toutefois, pour fonctionner de manière exhaustive, il doit s'agir d'un message HTTP correct.

Listener status (Statut de l'écouteur)

Champ qui indique si l'écouteur est en cours d'exécution ou non.

Test

Cliquez sur cette option pour tester la configuration de l'écouteur.

Clear Listener (Effacer l'écouteur)

Cliquez sur cette option pour arrêter le test de l'étape.

Onglet Response (Réponse)

L'option Response to Send (Réponse à envoyer) inclut le texte de la réponse.

Read Response From File (Lire la réponse à partir d'un fichier)

Cliquez sur cette option pour rechercher le système de fichiers pour une réponse.

Onglet Request (Demande)

Affiche la dernière demande ou demande d'origine, utilisée uniquement pendant la conception. Cet onglet affiche la dernière demande reçue par l'étape.

Le nom par défaut de l'étape Socket Server Emulator (Emulateur de serveur de socket) est **Socket Server Emulator** <numéro port>. Vous pouvez renommer l'étape à tout moment.

Etape Messaging Virtualization Marker (Marqueur de virtualisation de messagerie)

Utilisez cette étape pour indiquer qu'un scénario de test basé sur un message est conçu pour être utilisé dans l'environnement de service virtuel. Si le scénario de test écoute ou répond via JMS, ajoutez cette étape au modèle de service virtuel pour vérifier que vous pouvez le déployer dans le VSE.

Remarque : Vous ne devez pas utiliser cette étape avec le protocole de transport JMS.

Etape Compare Strings for Response Lookup (Comparer les chaînes pour rechercher des réponses)

Cette étape consulte une demande entrante pour un service virtuel. Puis, l'étape détermine la réponse appropriée en mode sans état, sans consulter aucune image de service. Les parties avec état du VSE ne sont pas prises en charge. Vous pouvez mettre en correspondance des demandes entrantes à l'aide de la correspondance de texte partielle, une expression régulière, etc.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Text to match (Texte à rechercher)

Saisissez le texte à comparer aux critères. Cette valeur est généralement une référence de propriété, comme LASTRESPONSE.

Range to match (Plage à rechercher)

Entrez le début et la fin de la plage.

If no match found (En cas de non-correspondance)

Dans la liste, sélectionnez l'étape suivante en cas de non-correspondance.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Store responses in a compressed form in the test case file (Stocker les réponses au format compressé dans le fichier de scénario de test)

Sélectionnée par défaut, cette option permet de compresser les réponses dans le fichier de scénario de test.

Case Response Entries (Entrées de réponse du scénario)

Permet d'ajouter, de déplacer et de supprimer des entrées.

Enabled (Activé)

Cette option est sélectionnée par défaut lorsque vous ajoutez une entrée. Pour ignorer une entrée, désélectionnez-la.

Name (Nom)

Entrez un nom unique pour l'entrée de réponse du scénario.

Delay Spec (Spécification du délai)

Entrez la plage de spécification du délai. La valeur par défaut est **1000-10000**, qui indique d'utiliser un délai aléatoirement sélectionné entre 1 000 et 10 000 millisecondes. Le format de la syntaxe est identique à celui des spécifications de délai de réflexion.

Criteria (Critères)

Cette zone fournit la chaîne à comparer au champ Text to match (Texte à rechercher). Pour modifier les critères, dans la zone Case Response Entries (Entrées de réponse du scénario) sélectionnez la ligne appropriée, puis sélectionnez un paramètre différent dans la liste Criteria (Critères).

Compare Type (Comparer le type)

Sélectionnez une option dans la liste :

- Find in string (default) (Rechercher dans la chaîne (valeur par défaut))
- Regular expression (Expression régulière)
- Starts with (Commence par)
- Ends with (Finit par)
- Exactly equals (Correspond exactement)

Response (Réponse)

Cette zone contient la réponse de cette étape si l'entrée correspond au champ Text to match (Texte à rechercher). Pour modifier la réponse, dans la zone Case Response Entries (Entrées de réponse du scénario) sélectionnez la ligne appropriée, puis sélectionnez un paramètre différent dans la liste Response (Réponse).

Criteria (Critères)

Permet de mettre à jour la chaîne de critères d'une entrée.

Response (Réponse)

Vous pouvez mettre à jour la réponse de l'étape pour une entrée.

Etape Compare Strings for Next Step Lookup (Comparer les chaînes pour rechercher l'étape suivante)

Cette étape permet de consulter une demande entrante et de déterminer l'étape suivante appropriée. Vous pouvez mettre en correspondance des demandes entrantes à l'aide de la correspondance de texte partielle ou d'une expression régulière, etc.

Chaque critère correspondant spécifie le nom de l'étape vers laquelle effectuer le transfert en cas de correspondance.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Text to match (Texte à rechercher)

Saisissez le texte à comparer aux critères. Cette valeur est généralement une référence de propriété, comme LASTRESPONSE.

Range to match (Plage à rechercher)

Entrez le début et la fin de la plage.

If no match found (En cas de non-correspondance)

Dans la liste, sélectionnez l'étape suivante en cas de non-correspondance.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Next Step Entries (Entrées de l'étape suivante)

Permet d'ajouter, de déplacer et de supprimer des entrées.

Enabled (Activé)

Cette option est sélectionnée par défaut lorsque vous ajoutez une entrée. Pour ignorer une entrée, désélectionnez-la.

Name (Nom)

Entrez un nom unique pour l'entrée de l'étape suivante.

Delay Spec (Spécification du délai)

Entrez la plage de spécification du délai. La valeur par défaut est **1000-10000**, qui indique d'utiliser un délai aléatoirement sélectionné entre 1 000 et 10 000 millisecondes. Le format de la syntaxe est identique à celui des spécifications de délai de réflexion.

Criteria (Critères)

Cette zone fournit la chaîne à comparer au champ Text to match (Texte à rechercher). Pour modifier les critères, dans la zone Case Response Entries (Entrées de réponse du scénario) sélectionnez la ligne appropriée, puis sélectionnez un paramètre différent dans la liste Criteria (Critères).

Compare Type (Comparer le type)

Sélectionnez une option dans la liste :

- Find in string (default) (Rechercher dans la chaîne (valeur par défaut))
- Regular expression (Expression régulière)
- Starts with (Commence par)
- Ends with (Finit par)
- Exactly equals (Correspond exactement)

Next Step (Etape suivante)

Dans la liste, sélectionnez l'étape suivante en cas de correspondance.

Criteria (Critères)

Vous pouvez mettre à jour la chaîne de critères pour une entrée.

Etape Virtual Java Listener (Ecouteur virtuel Java)

Utilisez cette étape pour gérer des appels de machine virtuelle Java virtualisés, par exemple un appel à un système EJB ou tout autre système distant. L'étape permet d'écouter les appels de méthode interceptés par l'agent Java de DevTest et de les convertir en demande du VSE standard.

Available Online Agents (Agents en ligne disponibles)

Répertorie tous les agents en ligne disponibles auquel vous pouvez vous connecter.

Connected Agents (Agents connectés)

Répertorie tous les agents en ligne ou hors ligne connectés pour le modèle de service virtuel. Les agents hors ligne apparaissent en police italique grise.


Connexion d'agents

Vous pouvez sélectionner les agents dans la liste Available Online Agents (Agents en ligne disponibles). Sélectionnez le ou les agents et cliquez sur le bouton de flèche vers la droite. L'agent se déplace vers la liste d'agents connectés.

Lorsque vous sélectionnez un agent dans la liste Available Online Agents (Agents en ligne disponibles) ou dans la liste Connected Agents (Agents connectés), la barre d'informations sous la liste contient des informations sur l'hôte et la classe principale pour l'agent.

Pour ajouter manuellement un agent à la liste d'agents connectés, utilisez le champ Add Agent (Ajouter un agent) situé au-dessus de la liste Connected Agents (Agents connectés).

Le nom d'agent ne peut pas être vide ni déjà présent dans la liste d'agents connectés. Si le nom d'agent saisi est présent dans la liste des agents en ligne, il sera déplacé de la liste des agents en ligne vers la liste des agents connectés.

Si un agent existant ne figure pas dans la liste Available Online Agents (Agents en ligne disponibles), saisissez-le dans la boîte de dialogue et cliquez sur Add (Ajouter) .


Ce mécanisme est essentiellement fourni pour ajouter des agents hors ligne qui n'étaient pas présents préalablement dans la liste d'agents connectés.

Déconnexion d'agents

Pour déconnecter un agent, sélectionnez-le dans la liste d'agents connectés et cliquez sur la flèche vers la gauche.

Sélection de classes et de protocoles

Pour rechercher des classes, sélectionnez la flèche Search for Classes (Rechercher des classes) et saisissez un nom de classe. Ces classes sont saisies sous forme de noms complets (y compris le package), à l'aide d'expressions régulières. Pour sélectionner des classes, sélectionnez leur nom dans la liste et sélectionnez la flèche vers la droite pour les déplacer dans le volet droit. Certaines classes s'affichent plusieurs fois ; une seule classe à la fois doit être sélectionnée pour la virtualiser.

Pour entrer une classe manuellement, sélectionnez la flèche Manually Enter a Class Name (Entrer manuellement un nom de classe). Saisissez le nom de la classe. Pour déplacer la classe vers le volet droit, sélectionnez-la et cliquez sur la flèche vers la droite. Pour récupérer une liste de classes proposées par l'agent de DevTest pour la virtualisation, sélectionnez la flèche Agent Suggestions (Suggestions de l'agent) et cliquez sur Retrieve (Récupérer). 

Pour ajouter un protocole à l'enregistrement, sélectionnez la flèche Protocols. Dans la liste de protocoles disponibles, sélectionnez ceux que vous voulez enregistrer et cliquez sur la flèche vers la droite pour les déplacer dans le volet droit.

Pour afficher ou changer les informations de configuration d'un protocole, double-cliquez sur une ligne contenant trois points (...) à droite du nom du protocole. La fenêtre Protocol Configuration (Configuration du protocole) s'affiche et vous pouvez mettre à jour les paramètres.

Etape Virtual Java Live Invocation (Appel dynamique virtuel via Java)

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Etape Virtual Java Responder (Répondeur virtuel Java)

Cette étape est utilisée avec l'étape Virtual Java Listener (Ecouteur virtuel Java) pour fournir des réponses à des appels de machine virtuelle Java virtualisés.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Etape Virtual TCP/IP Listener (Ecouteur virtuel de DRDA)

Utilisez cette étape pour simuler des connexions TCP/IP vers une application de serveur. L'étape permet d'écouter le trafic TCP/IP entrant et de le convertir au format de demande VSE standard.

Le nom par défaut de l'étape Virtual TCP/IP Listener (Ecouteur virtuel TCP/IP) est **Virtual TCP/IP Listener<numéro port>**. Vous pouvez renommer l'étape à tout moment.

Remplissez les champs suivants, tel qu'indiqué :

Listen port (Port d'écoute)

Saisissez le port sur lequel DevTest écoute le trafic TCP/IP.

Bind address (Adresse de liaison)

Saisissez l'adresse IP locale dans laquelle les connexions peuvent entrer. Si aucune adresse de liaison n'est spécifiée, l'étape de l'écouteur acceptera des connexions sur le port spécifié quel que soit le NIC (ou l'adresse IP) sur lequel il entre.

Use SSL to server (Utiliser une connexion SSL au serveur)

Si cette option est sélectionnée, une demande SSL (secured layer) est envoyée au serveur. Vous pouvez sélectionner cette case à cocher sans sélectionner Use SSL to client (Utiliser une connexion SSL au client). Dans ce cas, une connexion TCP simple est présentée pour l'enregistrement, mais ces demandes seront envoyées au serveur à l'aide d'une connexion SSL.

Use SSL to client (Utiliser une connexion SSL au client)

Cette option sera activée uniquement si l'option Use SSL to Server (Utiliser une connexion SSL au serveur) a été sélectionnée. Vérifiez si une demande SSL peut être lue à partir d'un client en utilisant un référentiel de clés client personnalisé. Lorsque vous sélectionnez Use SSL to client (Utiliser une connexion SSL au client), vous pouvez spécifier un référentiel de clés personnalisé ainsi qu'une phrase secrète. Si ces valeurs sont saisies, elles seront utilisées, au lieu des valeurs par défaut codées de manière irréversible.

SSL keystore file (Fichier de référentiel de clés SSL)

Pour rechercher votre fichier de référentiel de clés SSL, cliquez sur Select.. (Sélectionner). Le même fichier de référentiel de clés doit être disponible pour le serveur du VSE sur lequel le modèle de service virtuel est déployé.

Keystore password (Mot de passe du fichier de référentiel de clés)

Saisissez le mot de passe du référentiel de clés et cliquez sur Verify (Vérifier).

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse

- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape Virtual TCP/IP Live Invocation (Appel dynamique DRDA virtuel)

Utilisez cette étape pour effectuer un appel TCP/IP réel au serveur réel dans le contexte d'un service TCP/IP virtualisé. Cette étape est généralement créée via l'enregistrement et la virtualisation d'une forme de trafic TCP/IP. L'étape effectue la demande réelle, qui est basée sur la demande du VSE actuelle en cours d'utilisation.

Le nom par défaut de l'étape Virtual TCP-IP Live Invocation (Appel dynamique TCP/IP virtuel) est **TCP Protocole Live Invocation**<numéro port >. Vous pouvez renommer l'étape à tout moment.

Remplissez les champs suivants, tel qu'indiqué :

Target port (Port cible)

Indique le nom du port sur lequel la demande est envoyée.

Target server (Serveur cible)

Indique le nom du serveur vers lequel la demande est envoyée.

Use SSL to server (Utiliser une connexion SSL au serveur)

Si cette option est sélectionnée, une demande SSL (secured layer) est envoyée au serveur. Vous pouvez sélectionner cette case à cocher sans sélectionner Use SSL to client (Utiliser une connexion SSL au client). Dans ce cas, une connexion TCP simple est présentée pour l'enregistrement, mais ces demandes seront envoyées au serveur à l'aide d'une connexion SSL.

Use SSL to client (Utiliser une connexion SSL au client)

Cette option est activée uniquement si l'option Use SSL to server (Utiliser une connexion SSL au serveur) est sélectionnée. L'option Use SSL to client (Utiliser une connexion SSL au client) spécifie si l'application peut lire une demande HTTPS à partir du client à l'aide d'un référentiel de clés client personnalisé.

Valeurs :

- **Option sélectionnée** : vous pouvez spécifier un référentiel de clés personnalisé et une phrase secrète. Si vous saisissez ces valeurs, l'application les utilise au lieu des valeurs par défaut codées de manière irréversible.
- **Option désélectionnée** : l'application ne peut pas lire une demande SSL à partir d'un client à l'aide d'un référentiel de clés client personnalisé.

SSL keystore file (Fichier de référentiel de clés SSL)

Pour rechercher votre fichier de référentiel de clés SSL, cliquez sur Select.. (Sélectionner). Le même fichier de référentiel de clés doit être disponible pour le serveur du VSE sur lequel le modèle de service virtuel est déployé.

Keystore password (Mot de passe du fichier de référentiel de clés)

Saisissez le mot de passe du référentiel de clés et cliquez sur Verify (Vérifier).

Treat response as text (Traiter la réponse comme du texte)

Pour indiquer que la réponse doit être traitée comme du texte, sélectionnez cette option.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape Virtual TCP/IP Responder (Répondeur virtuel TCP/IP)

Remplissez les champs suivants, tel qu'indiqué :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Etape Virtual CICS Listener (Ecouteur virtuel CICS)

L'étape Virtual CICS Listener (Ecouteur virtuel CICS) permet de simuler un serveur CICS LINK.

Pour plus d'informations sur les champs de ce panneau, reportez-vous à la section [Utilisation du panneau de programmes CICS à virtualiser](#) (page 198).

Etape Virtual CICS Responder (Répondeur virtuel CICS)

L'étape Virtual CICS Responder (Répondeur virtuel CICS) est utilisée avec l'étape Virtual CICS Listener (Ecouteur virtuel CICS) pour transmettre des réponses à des demandes produites par l'écouteur. L'étape utilise une réponse virtuelle comme réponse à la demande correspondante.

Procédez comme suit:

1. Saisissez les données pour les champs suivant les instructions :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If environment error (En cas d'erreur d'environnement)

Valeur par défaut : Abort test (Interrompre le test)

Dans la liste, sélectionnez l'action ou l'étape à exécuter en cas d'échec du test en raison d'une erreur d'environnement.

2. Pour remplir la liste Conversational Model Properties (Propriétés du modèle conversationnel), utilisez les boutons Add (Ajouter) et Remove (Supprimer).

Etape CICS Transaction Gateway Listener (Ecouteur de la passerelle de transaction CICS)

L'étape CICS Transaction Gateway Listener (Ecouteur de la passerelle de transaction CICS) permet de simuler un serveur CTG, avec prise en charge SSL. Cette étape permet d'écouter les demandes CTG entrantes et de les convertir au format de demande virtuelle standard.

Remplissez les champs suivants, tel qu'indiqué :

Listen on port (Port d'écoute)

Saisissez le port sur lequel DevTest écoute le trafic CTG.

Bind address (Adresse de liaison)

Saisissez l'adresse IP locale dans laquelle les connexions peuvent entrer. Si aucune adresse de liaison n'est spécifiée, l'étape de l'écouteur acceptera des connexions sur le port spécifié quel que soit le NIC (ou l'adresse IP) sur lequel il entre.

Bind only (Liaison uniquement)

Pour acquérir la ressource réseau et passer à l'étape suivante, sélectionnez cette case à cocher. Une deuxième étape d'écoute sans utiliser l'option Bind only (Liaison uniquement) est requise. Cette option permet au modèle d'écouter sur un port ; l'application place les demandes dans la file d'attente jusqu'à leur consommation par une étape d'écoute. Le modèle effectue des tâches d'installation avant d'être déplacé dans la boucle attente/traitement/réponse. Par exemple, l'étape 1 du modèle acquiert le port d'écoute (à l'aide de l'option de liaison uniquement) et l'étape 2 déclenche un logiciel externe qui envoie des demandes.

Expect SSL From Clients (Attendre une connexion SSL des clients)

Si vous sélectionnez cette case à cocher, l'enregistreur suppose que les clients s'y connecteront à l'aide d'une connexion SSL. Le référentiel de clés et le mot de passe associés (s'ils ont été fournis) sont utilisés pour obtenir des informations de sécurité (par exemple, des certificats).

SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

CTG Protocol version (Version du protocole CTG)

Spécifie la version du protocole CTG à utiliser.

Locale (Paramètres régionaux)

Indique les paramètres régionaux qui représentent la langue et le code de pays indiqués au client CTG pendant l'établissement d'une liaison de protocole initial.

JVM text (Texte de la machine virtuelle Java)

Décrit la machine virtuelle Java sur le mainframe.

Server Class (Classe de serveur)

Définit une chaîne signalée par l'application au client. Ce champ n'est en général pas utilisé.

Client app ID (ID d'application cliente)

Définit une chaîne signalée par l'application au client. Ce champ n'est en général pas utilisé.

Has security (Sécurité disponible)

Indique au client CTG si l'authentification utilisateur est requise.

Enable server ping (Activer la commande ping sur le serveur)

Lorsqu'un client CTG se connecte à un serveur, celui-ci commence à envoyer des messages de commande ping au client pour vérifier la connexion de manière régulière. Cela n'est pas strictement nécessaire dans un environnement de test dans lequel le VSE est utilisé, mais l'activation de cette option entraîne l'émulation de ces messages ping par le serveur CTG du VSE.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape CICS Transaction Gateway Live Invocation (Appel dynamique de la passerelle de transaction CICS)

Utilisez cette étape pour effectuer un appel CTG réel au serveur réel dans le contexte d'un service CTG virtualisé. Cette étape est généralement créée via l'enregistrement et la virtualisation d'une forme de trafic CTG. Elle effectue la demande réelle en fonction de la demande du VSE actuelle en cours d'utilisation.

Remplissez les champs suivants, tel qu'indiqué :

Target server (Serveur cible)

Saisissez le nom ou l'adresse IP de l'hôte cible sur lequel le serveur CTG est exécuté.

Target port (Port cible)

Saisissez le numéro du port sur lequel le serveur CTG écoute.

Initiate SSL to the target server (Initialiser la connexion SSL au serveur cible)

Si cette option est sélectionnée, une demande SSL (Secure Socket Layer) est envoyée au serveur.

SSL keystore file (Fichier de référentiel de clés SSL)

Indique le nom du fichier de référentiel de clés.

Keystore password (Mot de passe du fichier de référentiel de clés)

Indique le mot de passe du fichier de référentiel de clés.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape CICS Transaction Gateway Responder (Répondeur de la passerelle de transaction CICS)

Utilisez cette étape avec l'étape CICS Transaction Gateway Listener (Ecouteur de la passerelle de transaction CICS) pour transmettre des demandes CTG produites par l'écouteur. L'étape utilise une réponse virtuelle pour répondre à la demande correspondante à l'aide du protocole CTG.

Vous pouvez créer cette étape via l'enregistrement et la virtualisation du trafic de la passerelle de transaction CICS.

Remplissez les champs suivants, tel qu'indiqué :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Etape Virtual DRDA Listener (Ecouteur virtuel de DRDA)

Utilisez l'étape Virtual DRDA Listener (Ecouteur virtuel de DRDA) pour virtualiser un trafic DRDA sur TCP/IP. Cette étape permet d'intercepter les données DRDA entrantes et de les convertir au format de demande virtuelle standard.

Remplissez les champs suivants, tel qu'indiqué :

Listen/Record on port (Port d'écoute/enregistrement)

Target host (Hôte cible)

Target port (Port cible)

DB2 IP address (Adresse IP DB2)

LISA IP address (Adresse IP LISA)

Stored proc param delimiter (Délimiteur de paramètres de processus stocké)

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape Virtual DRDA Response Builder (Générateur de réponses virtuelles de DRDA)

Cette étape permet de décomposer des demandes DRDA en commandes d'architecture DRDA spécifiques pour atténuer la complexité de la réception de commandes DRDA par le VSE dans différents groupements. Par exemple, DRDA envoie les mêmes ensembles de commandes par groupe d'une, de deux, trois, ou quatre par demande. La mise en correspondance de chaque commande et la création d'une réponse composite rendent nos modèles aussi flexibles que possible pour la lecture.

Etape Virtual DRDA Live Invocation (Appel dynamique DRDA virtuel)

Utilisez l'étape Virtual DRDA Live Invocation (Appel dynamique DRDA virtuel) pour effectuer un appel DRDA réel vers un serveur réel dans le contexte d'un service DRDA virtualisé. En règle générale, l'enregistrement et la virtualisation du trafic DRDA créent l'étape. L'étape effectue la demande réelle par rapport à la demande du VSE actuelle pendant la lecture.

Remplissez les champs suivants, tel qu'indiqué :

Listen/Record on port (Port d'écoute/enregistrement)

Target host (Hôte cible)

Target port (Port cible)

Indique le nom du port sur lequel la demande est envoyée.

DB2 IP address (Adresse IP DB2)

LISA IP address (Adresse IP LISA)

Stored proc param delimiter (Délimiteur de paramètres de processus stocké)

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

Etape IMS Connect Listener (Ecouteur d'IMS Connect)

Utilisez l'étape IMS Connect Listener (Ecouteur d'IMS Connect) pour répondre aux demandes IMS Connect effectuées par le service virtuel IMS Connect.

Remplissez les champs suivants, tel qu'indiqué :

Listen/Record on port (Port d'écoute/enregistrement)

Permet de définir le port sur lequel le client communique avec DevTest.

Target host (Hôte cible)

Désactivé pour l'étape d'écoute

Target port (Port cible)

Désactivé pour l'étape d'écoute

IMS Format File (Fichier au format IMS)

Désactivé pour l'étape d'écoute

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape IMS Connect Live Invocation (Appel dynamique IMS Connect)

Entrez les paramètres suivants.

Listen/Record on port (Port d'écoute/enregistrement)

Non applicable à l'étape d'appel dynamique

Target host (Hôte cible)

Permet d'indiquer le nom ou l'adresse IP de l'hôte cible sur lequel le serveur est exécuté.

Target port (Port cible)

Spécifie le numéro du port cible écouté par le serveur IMS.

IMS Format File (Fichier au format IMS)

Pour utiliser la prise en charge d'IMS Connect fournie avec DevTest par défaut, laissez ce champ vide.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

Etape Virtual IMS Connect Responder (Répondeur de connexion au service IMS virtuel)

Utilisez cette étape avec l'étape IMS Connect Listener (Ecouteur de connexion au service IMS) pour transmettre des réponses de connexion au service IMS.

Vous pouvez créer cette étape via l'enregistrement et la virtualisation du trafic IMS Connect.

Remplissez les champs suivants, tel qu'indiqué :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

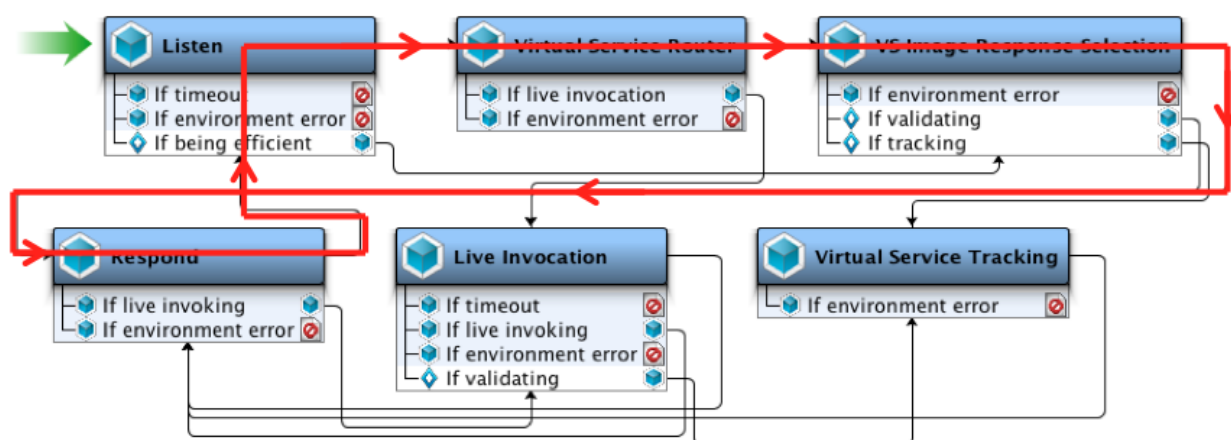
Etapes JMS du VSE

Les étapes suivantes concernent les modèles de service virtuel créés à l'aide du [protocole de transport JMS](#) (page 154) :

- JMS VSE Listen (Ecoute JMS VSE)
- JMS VSE Respond (Réponse JMS VSE)
- JMS VSE Live Invocation (Appel dynamique JMS VSE)

Au cours des opérations standard, le flux d'exécution du modèle de service virtuel ressemble en tout point à un autre service de VSE.

Le graphique suivant utilise une superposition pour illustrer le flux.



L'étape **Listen** (Ecouter) est l'étape JMS VSE Listen (Ecoute JMS VSE). L'étape **Respond** (Réponse) est l'étape JMS VSE Respond (Réponse JMS VSE). L'étape JMS VSE Live Invocation (Appel dynamique JMS VSE) n'est pas utilisée.

1. L'étape **Listen** (Ecouter) reçoit un message de demande et le convertit en une demande de VSE.
2. L'étape **Virtual Service Router** (Routeur de service virtuel) route le flux vers l'étape de sélection de réponse.
3. L'étape **VS Image Response Selection** (Sélection d'une réponse d'image de service virtuel) sélectionne une transaction correspondante à partir de l'image de service et génère une réponse de VSE.
4. L'étape **Respond** (Répondre) envoie un ou plusieurs messages dans la réponse du VSE.
5. Revenez à l'étape 1.

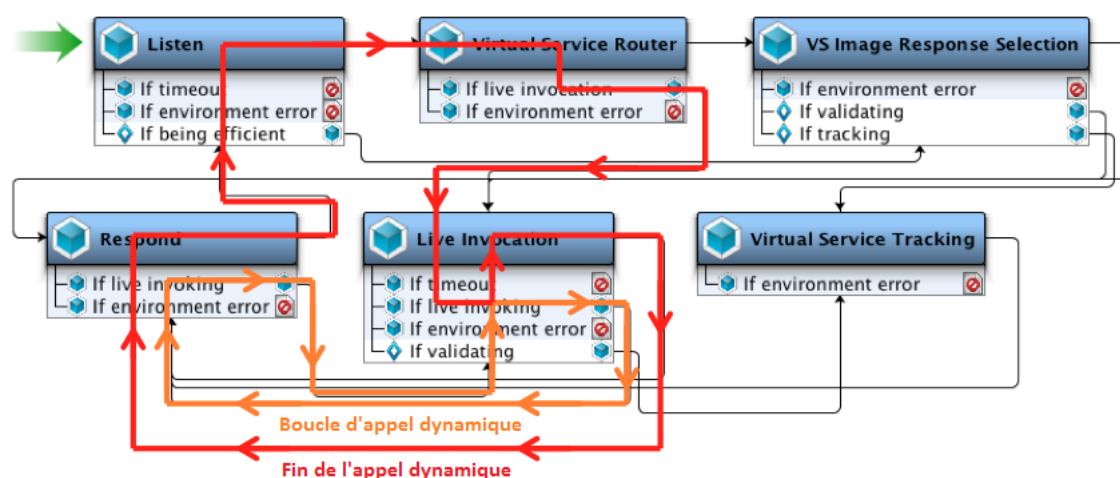
Pendant l'appel dynamique, le flux d'exécution du modèle de service virtuel est plus compliqué.

Les aspects suivants de la messagerie asynchrone rendent l'appel dynamique difficile :

- Une demande unique peut avoir plusieurs réponses.
- Les réponses peuvent être envoyées à différents intervalles.
- En général, il est impossible de déterminer quand toutes les réponses seront reçues.

En conséquence, l'étape **Live Invocation** (Appel dynamique) et l'étape **Respond** (Répondre) sont exécutées dans une boucle.

Le graphique suivant utilise une superposition pour illustrer le flux.



L'étape **Listen** (Ecouter) est l'étape JMS VSE Listen (Ecoute JMS VSE). L'étape **Live Invocation** (Appel dynamique) est l'étape JMS VSE Live Invocation (Appel dynamique JMS VSE). L'étape **Respond** (Réponse) est l'étape JMS VSE Respond (Réponse JMS VSE).

1. L'étape **Listen** (Ecouter) reçoit un message de demande et le convertit en une demande de VSE.
2. L'étape **Virtual Service Router** (Routeur de service virtuel) route le flux vers l'étape d'appel dynamique.
3. L'étape **Live Invocation** (Appel dynamique) envoie la demande au service dynamique.
4. L'étape **Live Invocation** (Appel dynamique) commence ses écoutes sur toutes les files d'attente de réponses dynamiques.
5. L'étape **Live Invocation** (Appel dynamique) reçoit une réponse unique à partir d'une des files d'attente de réponses dynamiques et la convertit en une réponse de VSE.
6. L'étape **Respond** (Réponse) renvoie un message de réponse au client.

7. Revenez à l'étape 5 et répétez le processus jusqu'à ce que l'étape **Live Invocation** (Appel dynamique) détermine que la transaction est terminée. Pour que l'étape **Live Invocation** arrive à cette conclusion, l'une des conditions suivantes est nécessaire :
 - Le délai d'expiration est défini et expire sans recevoir une autre réponse à partir d'une des files d'attente de réponses dynamiques. La valeur par défaut est 30 secondes. Le délai d'expiration est un paramètre avancé que vous pouvez définir dans l'enregistreur de VSE ou dans l'étape **Live Invocation** (Appel dynamique).
 - Le nombre maximum de réponses est défini et ce nombre a été atteint. La valeur par défaut est 1, ce qui indique que la boucle peut s'exécuter une seule fois. Le nombre maximum de réponses est un paramètre avancé que vous pouvez définir dans l'enregistreur de VSE ou dans l'étape **Live Invocation** (Appel dynamique).
 - Le modèle de service virtuel est près de sa capacité maximum si bien qu'il a besoin d'arrêter la transaction en cours du thread d'exécution du modèle actuel pour gérer une nouvelle demande. Si le délai d'expiration et le nombre maximum de réponses n'ont pas été définis, cette méthode est la seule permettant au modèle d'interrompre l'attente des réponses dynamiques et de revenir à l'étape **Listen** (Ecouter).
8. L'étape **Live Invocation** (Appel dynamique) génère une dernière réponse de VSE qui contient tous les messages de réponse transférés dans la boucle. L'étape revient une dernière fois à l'étape **Respond** (Répondre).
9. L'étape **Respond** (Répondre) n'envoie aucun message lors de la dernière boucle. A la place, l'étape **Respond** (Répondre) effectue les tâches habituelles de nettoyage de l'état du VSE.
10. Revenez à l'étape 1.

Etape JMS VSE Listen (Ecoute JMS VSE)

L'étape JMS VSE Listen (Ecoute JMS VSE) écoute les demandes JMS entrantes et les convertit en demandes de VSE standard.

L'éditeur d'étapes comprend des paramètres de base et avancés. Pour afficher les paramètres avancés, cliquez sur PRO en haut de l'éditeur.

L'onglet Request (Demande) contient la liste des réceptions.

Le champ Channel Name (Nom de canal) définit le nom de canal de demande. La valeur doit correspondre au nom d'opération défini dans l'image de service.

Pour désactiver et réactiver des canaux de demande, utilisez la case à cocher Enabled (Activé). Au moins un canal de demande doit être activé.

L'onglet ReplyTo Mappings (Mappages de ReplyTo) s'applique au scénario de messagerie séparé. Ce type de scénario a lieu dans les conditions suivantes :

- Un ensemble de files d'attente est accessible par DevTest.
- Un autre ensemble de files d'attente n'est pas accessible par DevTest.
- L'application est exécutée sur les files d'attente non accessibles.
- Les messages sont automatiquement envoyés au mot près aux files d'attente auxquelles DevTest peut accéder.

Chaque mappage comprend deux éléments : le nom de canal côté réponse et la destination de réponse du client utilisée par le client.

Etape JMS VSE Respond (Réponse JMS VSE)

L'étape JMS VSE Respond (Réponse JMS VSE) envoie un ou plusieurs messages dans la réponse de VSE.

L'éditeur d'étapes comprend des paramètres de base et avancés. Pour afficher les paramètres avancés, cliquez sur PRO en haut de l'éditeur.

Le champ Channel Name (Nom de canal) définit le nom de canal de réponse. La valeur doit correspondre à la propriété de métadonnées **operation.name** définie dans l'image de service.

Pour désactiver et réactiver des canaux de réponse, utilisez la case à cocher Enabled (Activé).

Etape JMS VSE Live Invocation (Appel dynamique JMS VSE)

L'étape JMS VSE Live Invocation (Appel dynamique JMS VSE) envoie des demandes au service dynamique.

L'éditeur d'étapes comprend des paramètres de base et avancés. Pour afficher les paramètres avancés, cliquez sur PRO en haut de l'éditeur.

L'onglet Live Request Send (Envoyer une demande dynamique) contient la liste des opérations à utiliser pour l'envoi d'une demande dynamique.

Le champ Channel Name (Nom de canal) définit le nom de canal de demande. La valeur doit correspondre au nom de canal dans l'étape JMS VSE Listen (Ecoute JMS VSE).

Pour désactiver et réactiver des canaux de demande, utilisez la case à cocher Enabled (Activé). Au moins un canal de demande doit être activé.

L'onglet Live Response Receive (Recevoir une réponse dynamique) contient la liste des opérations à utiliser pour la réception de réponses dynamiques.

Le champ Channel Name (Nom de canal) définit le nom de canal de réponse. La valeur doit correspondre au nom de canal dans l'étape JMS VSE Respond (Réponse JMS VSE).

Pour désactiver et réactiver des canaux de réponse, utilisez la case à cocher Enabled (Activé). Au moins un canal de réponse doit être activé.

Le paramètre Timeout (Délai d'expiration) et le paramètre Maximum Responses (Nombre maximum de réponses) comptent parmi les critères disponibles que l'étape utilise pour déterminer si elle doit quitter la [boucle d'appel dynamique](#) (page 365).

L'onglet ReplyTo Mappings (Mappages de ReplyTo) s'applique au scénario de messagerie séparé. Ce type de scénario a lieu dans les conditions suivantes :

- Un ensemble de files d'attente est accessible par DevTest.
- Un autre ensemble de files d'attente n'est pas accessible par DevTest.
- L'application est exécutée sur les files d'attente non accessibles.
- Les messages sont automatiquement envoyés au mot près aux files d'attente auxquelles DevTest peut accéder.

Chaque mappage consiste en deux éléments : le nom de canal côté réponse et la destination de réponse de service qui doit être envoyée au service afin que la réponse passe par la file d'attente de réponses dynamiques sur le canal de réponse.

Etape JCo IDoc Listener (Ecouteur de RFC via JCo)

Cette étape est utilisée pour répondre aux demandes de IDoc via JCo effectuées par le service virtuel de IDoc via JCo. Pour créer cette étape, enregistrez et virtualisez le trafic de fichiers IDoc via JCo.

Entrez les paramètres suivants.

Client RFC Connection Properties (Propriétés de connexion RFC du client)

Définit le fichier de propriétés de connexion RFC du client qui contient les propriétés de connexion que VSE utilise pour s'enregistrer sous un ID de programme dans une passerelle SAP et recevoir des fichiers IDoc. Les propriétés doivent être identiques à celles spécifiées dans un fichier .jcoServer.

Client RFC Destination Name (Nom de destination RFC sur le client)

Spécifie un nom unique qui identifie la destination RFC.

Client System Connection Properties (Propriétés de connexion au client)

Spécifie le fichier de propriétés de connexion au client qui contient les propriétés de connexion permettant de renvoyer les fichiers IDoc au client SAP. Ces propriétés doivent être identiques à celles spécifiées dans un fichier .jcoDestination que vous pouvez utiliser pour vous connecter au client SAP.

Client System Name (Nom du client)

Spécifie un nom unique pour identifier le client SAP.

Request Identifier XPath Expressions (Expressions XPath d'identificateur de demande)

Spécifie les expressions XPath que le protocole utilise avec la demande XML IDoc pour générer un identificateur. Les expressions XPath d'identificateur de demande peuvent être une expression XPath unique. Cet identificateur est utilisé pour mettre en corrélation une demande IDoc avec une réponse IDoc. Les expressions XPath peuvent également consister en une liste d'expressions XPath séparées par des virgules ; auquel cas, les valeurs résultantes de plusieurs expressions sont concaténées (séparées par des traits d'union) et utilisées comme identificateur.

Response Identifier XPath Expressions (Expressions XPath d'identificateur de réponse)

Définit les expressions XPath que le protocole utilise avec la réponse XML IDoc pour générer un identificateur. Les expressions XPath d'identificateur de réponse peuvent être une expression XPath unique. Cet identificateur est utilisé pour mettre en corrélation une réponse IDoc avec une demande IDoc reçue antérieurement. Les expressions XPath peuvent également consister en une liste d'expressions XPath séparées par des virgules ; auquel cas, les valeurs résultantes de plusieurs expressions sont concaténées (séparées par des traits d'union) et utilisées comme identificateur.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

La structure du VSE attend des étapes de réponse pour accepter l'un des éléments suivants :

- Un objet de réponse
- Une liste d'objets de réponse
- Un document XML représentant l'un ou l'autre

Remarque : Si cette case à cocher n'est pas sélectionnée, l'étape produira une liste d'objets de réponse. L'étape produit la liste même si elle contient une seule réponse.

Par défaut, la réponse de l'étape est au format XML.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape JCo IDoc Live Invocation (Appel dynamique de fichier IDoc via JCo)

Entrez les paramètres suivants.

Client RFC Destination Name (Nom de destination RFC sur le client)

Spécifie un nom unique qui identifie la destination RFC.

Server RFC Connection Properties (Propriétés de connexion RFC du serveur)

Spécifie le fichier de propriétés qui contient les propriétés de connexion que VSE utilise pour s'enregistrer sous un ID de programme dans une passerelle SAP et recevoir des fichiers IDoc. Les propriétés doivent être identiques à celles spécifiées dans un fichier .jcoServer pour démarrer un programme de serveur JCo qui reçoit des fichiers IDoc à partir du serveur SAP.

Server RFC Destination Name (Nom de destination RFC sur le serveur)

Spécifie un nom unique pour identifier la destination RFC sur le serveur.

Server System Connection Properties (Propriétés de connexion au serveur)

Définit le fichier de propriétés de connexion au serveur qui contient les propriétés de connexion permettant de renvoyer les fichiers IDoc au serveur SAP. Ces propriétés doivent être identiques à celles spécifiées dans un fichier .jcoDestination que vous pouvez utiliser pour vous connecter au serveur SAP.

Server System Name (Nom du serveur)

Spécifie un nom unique pour identifier le serveur SAP.

Etape JCo IDoc Responder (Répondeur IDoc via JCo)

Cette étape est utilisée avec l'étape JCo IDoc Listener (Ecouteur de fichiers IDoc via JCo) pour transmettre des réponses IDoc via JCo. Vous pouvez créer cette étape via l'enregistrement et la virtualisation du trafic de fichiers IDoc via JCo.

Il n'existe aucun paramètre pour cette étape.

Etape JCo RFC Listener (Ecouteur de RFC via JCo)

Cette étape est utilisée pour répondre aux demandes de RFC via JCo effectuées par le service virtuel de RFC via JCo. Pour créer cette étape, enregistrez et virtualisez le trafic de fichiers RFC via JCo.

Entrez les paramètres suivants.

Client System Name (Nom du client)

Spécifie un nom unique pour identifier le client SAP.

Client System Connection Properties (Propriétés de connexion au client)

Le fichier de propriétés de connexion au client contient les propriétés de connexion à la destination sur le système client. Il doit s'agir d'un fichier .properties du répertoire Data (Données) de votre projet qui contient les propriétés génériques d'un fichier .jcoServer. Ce fichier ne doit pas spécifier la propriété **jco.server.repository_destination**. Pour plus d'informations sur les propriétés prises en charge, consultez le document JavaDocs **com.sap.conn.jco.ext.ServerDataProvider** dans le dossier **doc** de votre répertoire d'installation.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

Si cette case à cocher est sélectionnée, la réponse d'étape (demande entrante) est mise en série au format XML et peut être traitée comme du texte. La mise en série est annulée lors de l'étape de recherche de réponse. Lorsqu'elle est sélectionnée, cette option ralentit le service virtuel considérablement.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape JCo RFC Live Invocation (Appel dynamique de RFC via JCo)

Entrez les paramètres suivants.

Destination System Name (Nom du système de destination)

Définit un nom unique pour identifier le système SAP sur lequel le RFC est exécuté. Cette valeur est souvent identique au nom du référentiel.

Destination System Connection Properties (Propriétés de connexion au système de destination)

Spécifie le fichier de propriétés de connexion au système de destination qui contient les propriétés de connexion permettant de se connecter système contenant le référentiel. Il doit s'agir d'un fichier `.properties` du répertoire Data (Données) de votre projet et contenir les propriétés génériques d'un fichier `.jcoDestination`. Pour plus d'informations sur les propriétés prises en charge, consultez le document JavaDocs

com.sap.conn.jco.ext.DestinationDataProvider dans le dossier **doc** de votre répertoire d'installation. Il peut s'agir du même fichier que le fichier de propriétés de connexion de référentiel.

Format step response as XML (Transformer la réponse d'étape en fichier XML)

Si cette case à cocher est sélectionnée, la réponse d'étape (la réponse du système dynamique) est mise en série au format XML et peut être traitée comme du texte. La mise en série est annulée lors de l'étape de réponse. Lorsqu'elle est sélectionnée, cette option ralentit le service virtuel considérablement.

If Environment Error (En cas d'erreur d'environnement)

Sélectionnez l'étape à exécuter ou l'action à effectuer si une erreur d'environnement est renvoyée.

Etape VSE Lookup (Recherche de VSE)

Pour la prise en charge du mode d'exécution de basculement par une étape d'appel dynamique, elle doit savoir que l'étape utilisée recherche les réponses VSE de sorte à rediriger le modèle de service virtuel vers l'étape correcte si nécessaire. Ce champ contient une liste d'étapes dans le modèle de service virtuel. Sélectionnez l'étape VSE Lookup (Recherche de VSE) standard. Cela permet à l'étape d'appel dynamique de se déplacer vers l'étape de recherche de réponse VSE si nécessaire.

Etape JCo RFC Responder (Répondeur RFC via JCo)

Cette étape est utilisée avec l'étape JCo RFC Listener (Ecouteur RFC via JCo) pour transmettre des réponses RFC via JCo. Vous pouvez créer cette étape en enregistrant et virtualisant le trafic RFC via JCo.

Entrez les paramètres suivants :

Responses list property name (Nom de la propriété de la liste de réponses)

Indique le nom de la propriété à rechercher pour la réponse à envoyer.

Conversational Model Properties (Propriétés du modèle conversationnel)

Saisissez une propriété et cliquez sur Add (Ajouter). Pour supprimer une propriété, sélectionnez-la dans la liste et cliquez sur Remove (Supprimer). Les propriétés répertoriées ici sont associées à la session de conversation actuelle, qui rend leur valeur disponible pour les demandes conversationnelles en aval.

Chapitre 11: Désensibilisation des données

Dans le VSE, la désensibilisation consiste à tenter de reconnaître des données sensibles et de les substituer par des valeurs aléatoires, mais dont le format est valide pendant l'enregistrement. Utilisez la désensibilisation de données lorsque vous ne voulez pas utiliser les données réelles du client comme données de test.

Vous pouvez activer la désensibilisation de données dans le VSE de trois manières :

- [Désensibilisation dynamique](#) (page 376)
- [Désensibilisation statique](#) (page 376)
- [Protocole de données Data Desensitizer \(Désensibilisation de données\)](#) (page 253)

Désensibilisation dynamique

La désensibilisation dynamique se produit au niveau de la couche de transport. La désensibilisation est appelée via l'activation de la case à cocher Desensitize (transport layer) (Désensibiliser (couche de transport)) dans l'onglet Basics (Paramètres de base) de l'enregistreur de service virtuel.

Ce programme de désensibilisation utilise des filtres volatiles qui veillent à ce que les informations sensibles ne soient jamais écrites sur le disque pendant la phase d'enregistrement. Le fichier **desensitize.xml** dans le répertoire de base de DevTest permet de configurer des désensibilisations des données de sorte à reconnaître des modèles connus, tels que des numéros de carte de crédit. Ce fichier remplace les données actives par des données réalistes, mais inutilisables. Le fichier utilise le modèle d'expression régulière correspondant pour identifier et rechercher des données sensibles. Ce fichier est analysé lors de chaque démarrage de l'enregistreur.

Vous pouvez utiliser les modèles de génération de chaînes de données de test intégrés comme options de données de remplacement. Les données de test contiennent 40 000 lignes disponibles, notamment des données de remplacement pour certains types de données communs : noms, adresses, numéros de téléphone et numéros de carte de crédit.

Vous pouvez personnaliser ces modèles prédéfinis pour créer votre propre modèle. Il est recommandé d'utiliser une boîte à outils d'expressions régulières telle que RegxBuddy. RegxBuddy permet de coller votre charge utile enregistrée et met en surbrillance de manière interactive les correspondances des expressions régulières à mesure que vous les ajustez.

Les correspondances sont traitées dans l'ordre d'apparition dans le fichier ; placez donc vos correspondances les plus spécifiques en premier.

Pour éviter des problèmes d'échappement de texte (notamment avec l'expression régulière), vous **devez** placer entre les caractères < et > l'expression régulière <expression régulière> et le texte enfant de <remplacement> dans un élément CDATA.

Désensibilisation statique

La désensibilisation de données statique implique une recherche et un remplacement manuels des données dans une image de service existante. Pour accéder au menu Search and Replace (Rechercher et remplacer), cliquez avec le bouton droit de la souris sur un noeud dans l'image de service et sélectionnez Search and Replace (Rechercher et remplacer).

Spécifiez une chaîne spécifique que vous voulez remplacer, puis indiquez l'étendue de la modification. Cliquez sur Replace (Remplacer) et la fonction de recherche/remplacement s'applique aux zones sélectionnées.

Gestionnaire de protocoles de données Data Desensitizer (Désensibilisation de données)

Le gestionnaire de protocoles de données Data Desensitizer (Désensibilisation des données) permet l'application de règles de désensibilisation lorsqu'un autre protocole de données doit rendre visible un corps de demande ou de réponse. Par exemple, un corps de message HTTP est converti au format gzip.

Pour plus d'informations sur cette fonctionnalité, reportez-vous à la section [Désensibilisation des données](#) (page 253) de la rubrique *Utilisation de CA Service Virtualization*.

Chapitre 12: Exécution de la virtualisation

La virtualisation consiste dans le processus par lequel le VSE répond au client en l'absence du serveur. La virtualisation utilise le modèle de service virtuel et l'image de service.

Ce chapitre traite des sujets suivants :

[Préparation de la virtualisation](#) (page 380)

[Déploiement et exécution d'un service virtuel](#) (page 382)

[Exécution de demandes dynamiques](#) (page 385)

[Affichage de session et réparation de modèle](#) (page 398)

[Mesures du VSE](#) (page 401)

Préparation de la virtualisation

Démarrage de l'environnement de service virtuel

L'exécution de la virtualisation requiert le démarrage de l'environnement de service virtuel (VSE). Le VSE doit être enregistré dans le registre DevTest.

Pour démarrer le VSE, sélectionnez Programmes, DevTest, Virtual Service Environment (Environnement de service virtuel) dans le menu Démarrer.

Une fenêtre s'ouvre pour la création d'un VSE appelé **lisa.VSEServer** et se connecte à l'instance de registre.

Remarque : Vous pouvez réduire la fenêtre Virtual Service Environment (Environnement de service virtuel), mais ne la fermez pas.

Si vous avez installé des services système Windows pour DevTest, vous pouvez utiliser le service système pour démarrer le VSE.

Vous pouvez également utiliser la commande suivante pour démarrer un VSE nommé à partir d'une invite de commande :

```
[LISA_HOME]\bin\VirtualServiceEnvironment.exe -n VSEName -m RegistryName
```

VSEName (Nom de l'environnement de service virtuel)

Spécifie le nom du VSE.

RegistryName (Nom du registre)

Spécifie le nom d'un registre existant.

Exécution d'environnements de services virtuels multiples

Pour exécuter plusieurs serveurs VSE sur un ordinateur, ajoutez l'option de ligne de commande **-p port** lors de l'exécution du fichier **VirtualServiceEnvironment.exe**, où **port** spécifie le numéro de port. Le numéro de port diffère en fonction des différentes instances du VSE.

La propriété **maxvirtualservices** dans votre licence limite le nombre de services virtuels que vous pouvez exécuter.

Utilisation du gestionnaire du VSE pour la configuration du VSE

Le gestionnaire du VSE permet de changer vos environnements de services virtuels.

Pour plus d'informations sur le gestionnaire du VSE, reportez-vous à la section [Commandes du gestionnaire du VSE](#) (page 418) de la rubrique *Utilisation de CA Service Virtualization*.

Déploiement et exécution d'un service virtuel

Procédez comme suit:

1. Dans la console du VSE, cliquez sur Deploy a new virtual service to the environment

(Déployer un nouveau service virtuel dans l'environnement) .

La fenêtre Deploy Virtual Service (Déployer un service virtuel) s'ouvre.

2. Saisissez le nom d'une archive de modèle (MAR) à charger et cliquez sur Deploy (Déployer).
 - L'archive MAR doit contenir un service virtuel. Lorsque vous cliquez sur Deploy (Déployer), le service charge dans la console du VSE et le service est disponible pour l'exécution.
 - Vous pouvez également accéder au panneau Project (Projet) dans DevTest Workstation et cliquer avec le bouton droit de la souris sur un modèle de service virtuel pour ouvrir la fenêtre Deploy Virtual Service (Déployer un service virtuel).

La fenêtre Deploy Virtual Service (Déployer un service virtuel) s'ouvre.

3. Modifiez les champs à votre convenance :

nom

Nom du service virtuel référencé par le modèle de service virtuel sélectionné.

VS model (Modèle de service virtuel)

Indique le modèle de service virtuel sélectionné.

Configuration

(Facultatif) Permet de sélectionner un autre fichier de configuration.

Balise group

Nom du [groupe de services virtuels](#) (page 433) pour le service virtuel. Si les services virtuels déployés ont des balises group, ces balises sont disponibles dans la liste déroulante. Une balise group doit commencer par un caractère alphanumérique et peut contenir des caractères alphanumériques et les caractères spéciaux suivants :

- Point (.)
- Tiret (-)
- Trait de soulignement (_)
- Symbole du dollar (\$)

Concurrent capacity (Capacité simultanée)

Nombre qui indique la capacité de charge. La *capacité* est le nombre d'utilisateurs virtuels (instances) qui peuvent être exécutés simultanément avec le modèle de service virtuel. Dans cet exemple, la capacité indique le nombre de threads existants pour les demandes de service de ce modèle de service.

VSE alloue un nombre de threads équivalent à la capacité simultanée totale. Chaque thread consomme quelques ressources système, même lorsqu'ils sont inactifs. Par conséquent, pour optimiser les performances système globales, définissez ce paramètre aussi sur une valeur aussi faible que possible. Déterminez les paramètres corrects de manière empirique en les ajustant jusqu'à atteindre les performances souhaitées, ou jusqu'à leur augmentation n'entraîne plus d'amélioration des performances.

Les protocoles prédéfinis utilisent un service d'exécution de tâches de niveau structure pour réduire l'utilisation des threads. Dans le cas de ces protocoles, une capacité simultanée supérieure à 2-3 par noyau est rarement utile, sauf si le modèle de service virtuel a été considérablement personnalisé.

Pour des extensions et tout modèle de service virtuel qui n'utilise pas de protocole prédéfini, la définition d'un délai de réflexion long peut consommer un thread pour la durée de ce délai. Dans ces cas, vous devrez peut-être augmenter la capacité simultanée.

La formule suivante donne un paramètre initial approximatif dans ces cas :

Capacité simultanée = (Transactions souhaitées par seconde / 1000) * délai de réflexion moyen en ms * (pourcentage de délai de réflexion / 100)

Exemple :

Supposons que vous utilisez un protocole personnalisé qui n'utilise pas le service d'exécution de tâches de structure pour gérer des délais de réflexion. Vous voulez un débit global de 100 transactions par seconde. Le délai de réflexion moyen dans l'image de service est de 200 ms et le service virtuel est déployé avec un délai de réflexion de 100 %.

$(100 \text{ transactions par seconde} / 1000) * 200 \text{ ms} * (100 / 100)$
= 20

Dans ce cas, les threads se bloquent pendant environ 200 ms en moyenne avant de répondre et durant ce temps, ils ne parviennent pas à gérer de nouvelles demandes. Nous avons par conséquent besoin d'une capacité de 20 pour contenir 100 transactions par seconde. Un thread deviendra disponible toutes les 10 ms en moyenne, ce qui devrait suffire à obtenir 100 transactions par seconde.

Valeur par défaut : 1

Think time scale (Pourcentage de délai de réflexion)

Indique le pourcentage de délai de réflexion enregistré.

Remarque : Une étape déduit sa propre durée de traitement du délai de réflexion pour obtenir une fréquence d'exécutions des tests cohérente.

Par défaut : 100

Exemples :

- Pour doubler le délai de réflexion, utilisez 200.
- Pour réduire de moitié le délai de réflexion, utilisez 50.

Start the service on deployment (Lancer le service au déploiement)

Indique si le service doit être déployé et démarré immédiatement.

Valeurs :

- **Option sélectionnée :** le service est déployé et démarré immédiatement.
- **Option désélectionnée :** le service est déployé, puis vous le démarrez manuellement à partir de la console du VSE.

If service ends, automatically restart it (Redémarrer automatiquement le service)

Indique si l'exécution du service doit être maintenue, même lorsqu'une session d'émulation se termine.

Valeurs :

- **Option sélectionnée :** l'exécution du service se poursuit après la fin du service d'émulation.
- **Option désélectionnée :** le service s'arrête lorsque le service d'émulation prend fin.

Valeur par défaut : Sélectionné

4. Cliquez sur Deploy (Déployer).

La console de VSE affiche l'état de service virtuel chargé.

Remarque : un service virtuel peut avoir les états suivants :

Deployed (Déployé)

Aucun service portant le nom saisi n'est déjà déployé. Le service est déployé.

Redeployed (Redéployé)

Un service portant le nom saisi est déployé avec le même fichier .vsm. Le service est redéployé.

Overridden (Remplacé)

Un service portant le nom saisi est déployé avec un fichier .vsm différent de celui associé au service indiqué. L'application vous invite à remplacer le service déployé.

Exécution de demandes dynamiques

Exécutez des demandes dynamiques au niveau du VSE après le déploiement du service virtuel. Si possible, notez le service dynamique et configurez les paramètres de passerelle/proxy de sorte que le client communique avec le VSE.

Pour plus d'informations, consultez la section :

- [Accès à la console VSE](#) (page 385)
- [Barre d'outils](#) (page 386)
- [Onglet Services](#) (page 390)
- [Onglet Matching \(Correspondance\)](#) (page 392)
- Onglet Request Events Details (Détails des événements de demande)

Accès à la console VSE

Pour gérer et surveiller des images de service déployées, utilisez la console VSE. A partir de cette console, vous pouvez déployer, démarrer, afficher, arrêter, redéployer et supprimer des services virtuels.

Procédez comme suit:

1. Dans la station de travail DevTest Workstation, cliquez sur Server Console (Console de serveur pour ouvrir la console de serveur de DevTest.
2. Sélectionnez le service VSE et double-cliquez dessus pour ouvrir la console VSE.
3. Dans la console VSE, vérifiez que le service virtuel est déployé ; le statut doit être défini sur Running (En cours d'exécution).
4. Pour vérifier que le service a reçu les demandes, consultez le nombre de transactions (Txn count, Nombre de transactions) dans la console du VSE.


Remarque : Si un modèle de service virtuel déployé n'affiche aucune transaction, le client n'est pas correctement configuré. Reconfigurez le client de sorte à référencer le modèle virtuel au lieu du système réel. Si un autre service utilise ce port, arrêtez-le ou modifiez le paramètre de port afin d'éliminer le conflit.







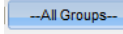
Barre d'outils de la console VSE

La console VSE dispose de sa propre barre d'outils, formée par les boutons suivants. Vous pouvez également accéder à ces fonctions en pointant sur un service et en cliquant dessus avec le bouton droit de la souris.



La barre d'outils de la console VSE contient les commandes suivantes :

		Permet de déployer ou de redéployer un service virtuel dans l'environnement.
--	---	--

		Permet de démarrer un service virtuel sélectionné.
Show (Afficher)		Permet d'afficher la vue d'inspection du service virtuel sélectionné.
View (Vue)		Permet d'afficher les informations de session/suivi du service virtuel sélectionné..
Set (Définir)		Spécifie le comportement du service virtuel sélectionné.
Reset (Réinitialiser)		Permet de réinitialiser le nombre de transactions et d'erreurs pour le service virtuel sélectionné.
Stop (Arrêter)		Permet d'arrêter le service virtuel sélectionné.
Remove (Supprimer)		Permet de supprimer le service virtuel sélectionné de l'environnement.
Configure (Configurer)		Permet de configurer le nettoyage des données de suivi.
Shut down (Fermer)		Permet d'arrêter l'ensemble de l'environnement du service virtuel.
Groups (Groupes)		Permet d'indiquer si les services associés à tous les groupes, à aucun groupe ou à un groupe doivent être affichés.

Deploy a new virtual service to the environment (Déployer un nouveau service virtuel dans l'environnement)

Pour déployer un service virtuel à partir d'une archive de modèle (MAR), sélectionnez cette option.


Saisissez le nom d'une archive de modèle (MAR) à charger. L'archive MAR doit contenir un service virtuel. Lorsque vous cliquez sur Deploy/Redeploy (Déployer/redéployer), le service charge dans la console de VSE et est disponible pour l'exécution.

Start the selected virtual service (Lancer le service virtuel sélectionné)

Permet de démarrer un service virtuel sélectionné.

Une fenêtre de confirmation s'ouvre pour indiquer que le service a démarré. Cliquez sur OK.

Affiche la vue d'inspection du service virtuel sélectionné.

Le bouton Show Inspection View (Afficher la vue d'inspection)  ouvre un onglet contenant un panneau d'inspecteur adapté aux services virtuels. Ce panneau contient deux onglets, [Matching \(Correspondance\)](#) (page 392) et Request Event Details (Request Event Details).

Onglet Service de la console VSE

Onglet Services

L'onglet Services permet de trier les valeurs de colonne, dans l'ordre croissant ou décroissant, en cliquant sur la flèche vers le bas à droite du nom de colonne. Cette flèche permet également de sélectionner les colonnes à afficher dans cette fenêtre.

Cet onglet contient les champs suivants.

Name (Nom)

Indique le modèle de service virtuel actuellement déployé.

Resource / Type (Ressource/Type)

Indique le port et le type, ou le protocole du service.

Status (Statut)

Indique l'état actuel du service virtuel.

Up-Time (Temps de disponibilité)

Indique la durée écoulée depuis le démarrage du service.

Txn Count (Nombre de transactions)

Nombre de transactions enregistrées après le démarrage du service.

Execution Mode (Mode d'exécution)

Mode d'exécution du service virtuel.

Group (Groupe)

[Groupe de services virtuels](#) (page 433) auquel appartient le service virtuel.

Errors (Erreurs) :

Un point rouge indique que des erreurs se sont produites lors de l'exécution du service.

Le panneau Virtual Service Details (Détails du service virtuel) situé au bas de la fenêtre contient des détails sur le service.

Ce panneau contient les champs suivants.

Model Name (Nom du modèle)

Indique le nom du modèle de service virtuel actuellement déployé.

Execution Mode (Mode d'exécution)

Permet d'afficher le mode d'exécution pour ce service virtuel. Reportez-vous à la section Spécification du comportement du modèle sélectionné de la rubrique *Utilisation de CA Service Virtualization*.

Last Start (Dernier démarrage)

Indique la date et l'heure du dernier démarrage de ce service.

Transaction Count (Nombre de transactions)

Nombre de transactions enregistrées après le démarrage du service.

Current txn/s (Nombre actuel de transactions par seconde)

Indique le nombre de transactions en cours d'exécution.

Capacity (Capacité)

Indique le nombre d'utilisateurs virtuels (instances) pouvant être simultanément exécutés avec le modèle de service virtuel. Capacity (Capacité) indique le nombre de threads existants pour les demandes de service de ce modèle de service. Vous pouvez mettre à jour ce champ lors de l'exécution du service.

Balise group

Nom du [groupe de services virtuels](#) (page 433) pour le service virtuel. Si les services virtuels déployés ont des balises group, ces balises sont disponibles dans le champ lorsque vous y entrez un caractère. Une balise group doit commencer par un caractère alphanumérique et peut contenir des caractères alphanumériques et quatre autres caractères : le point (.), le trait d'union (-), le trait de soulignement (_) et le signe dollar (\$).

Vous devez utiliser la touche **Tab** ou **Entrée** après avoir entré la balise group, puis cliquer sur Update (Mise à jour) pour mettre le champ à jour.

Pour supprimer une balise group d'un service virtuel, cliquez sur le **X** à côté du nom de balise group, puis cliquez sur Update (Mettre à jour).

Config Name (Nom de configuration)

Indique le nom du fichier de configuration utilisé par ce service.

Auto-Restart (Redémarrage automatique)

Indique si l'option de redémarrage automatique a été sélectionnée pour ce service. Pour basculer sa valeur de Yes à No, ou de No à Yes lors de l'exécution du service, cliquez sur ce champ.

Last End (Dernière arrêt)

Indique la date et l'heure du dernier arrêt de ce service.

Error Count (Nombre d'erreurs)

Indique le nombre d'erreurs reçues.

Peak txn/s (Nombre maximum de transactions par seconde)

Indique le nombre maximum de transactions exécutées simultanément.

Think Scale (Pourcentage de délai de réflexion)

Indique le pourcentage du délai de réflexion enregistré.

Cliquez sur le nom du service virtuel dans la console VSE pour télécharger l'archive de support pour le service virtuel. Le système vous invite à enregistrer le fichier MAR associé à ce service virtuel.

Onglet Matching (Correspondance) de la console VSE

L'onglet Matching (Correspondance) répertorie les demandes récentes traitées par les services virtuels. Lorsqu'une demande est sélectionnée, une description de la mise en correspondance (ou non-correspondance) de la demande s'affiche. Les mêmes informations sont enregistrées dans le fichier **vse_matches.log**. Si des événements ont été associés à la demande sélectionnée, ils seront affichés dans le côté droit du panneau.

Virtual Service Environment VSE@Default

Services Metrics **kioskV5 Inspector** (X)

Recent Requests

Live Request

2011-08-10 08:07:16,405
deleteToken
token: -8588995a:2557022286c:-5257

2011-08-10 08:07:13,441
getNewToken
username: admin
password: admin

2011-08-10 08:07:01,911
getNewToken

Refresh

What Happened

Service Image: VServices/Images/si-kioskV5.vsi
Session: -8588995a:2557022286c:-5257 (new: [2] Conversation 1)

Match Type: Meta (in conversation)
VSE responded from:
[3] getNewToken
username: demo1
password: pass

Events

Timestamp	Event	Short Info
-----------	-------	------------

Matching Request Event Details

Refresh ☒ Auto Refresh

Onglet Request Event Details (Détails des événements de demande) de la console de VSE

L'onglet Request Events Details (Détails des événements de demande) indique la liste des demandes entrantes qui ont entraîné une erreur du service virtuel. Lorsque vous sélectionnez une demande, la liste des étapes de modèle de service virtuel exécutées s'affiche, avec les étapes contenant des événements d'erreur sélectionnées. Pour consulter les événements qui se sont produits lors du traitement pour une étape (similaire à l'exécution d'un test interactif), sélectionnez cette étape.

Virtual Service Environment VSE@Default

Services Metrics **kioskV5 Inspector**

Errored Requests

Request

2011-08-12 13:41:36,985
deleteToken
token: -4655d680:656078c197c-6627

2011-08-12 13:41:35,225
getNewToken
username: itko

Refresh

Steps Executed

Name
HTTP/S Listen
Prepare Request
VS Image Response Selection
Prepare Response
HTTP/S Respond

Events -- Prepare Request

Timestamp	Event	Short Info
2011-08-12 13:41:36,978	Property set	lisa.vse.request
2011-08-12 13:41:36,978	Step response	Prepare Request
2011-08-12 13:41:36,978	Assertion fired	Prepare Request [if being efficient]

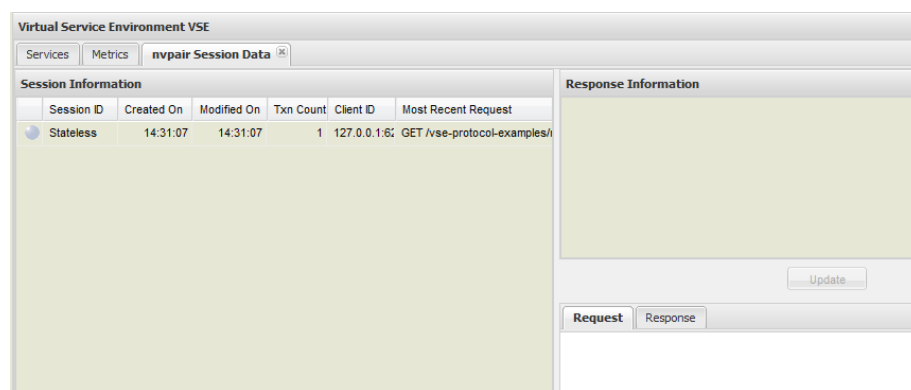
Matching **Request Event Details**

Refresh ☒ Auto Refresh

Remarque : Lorsqu'un service virtuel dépasse 100 transactions par seconde, les événements Property Set (Propriété définie) et Property Removed (Propriété supprimée) sont désactivés pour permettre de meilleures performances globales.

View Session/Tracking Information for the Selected Virtual Service (Afficher les informations de session/suivi pour le service virtuel sélectionné)

Cette option permet de consulter les informations de suivi de session pour le service virtuel. Pour plus d'informations, consultez la section [Affichage de session et réparation de modèle](#) (page 398).

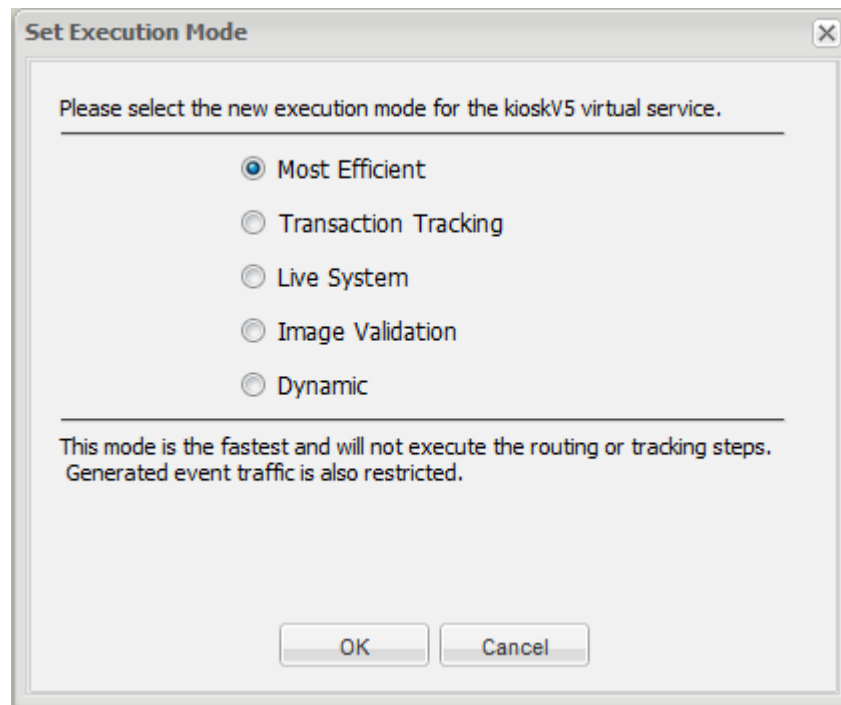


Redeploy the Selected Virtual Service to the Environment (Redéployer le service virtuel sélectionné dans l'environnement)

Si vous modifiez l'image de service ou le modèle de service virtuel, enregistrez les modifications et redéployez l'image de service modifiée dans la console de VSE en cliquant sur Deploy/Redeploy (Déployer/Redéployer).

Spécification du comportement du modèle sélectionné

Cette option vous permet de définir un mode d'exécution pour le service virtuel. Le nombre de modes d'exécution disponibles pour un service virtuel dépend du type d'étape de test. Par exemple, si un modèle n'a pas d'étape d'appel dynamique, il ne prend pas en charge les options Learning (Apprentissage) ou Live Invocation (Appel dynamique).



Les modes d'exécution disponibles sont les suivants :

Most Efficient (Optimisé)

Il s'agit du mode le plus rapide ; il n'exécute pas les étapes de routage ni de suivi. Ce mode limite également le suivi de l'événement généré.

Stand In (Remplacer)

Ce mode route d'abord une demande vers le service virtuel (identique à celui du mode Most Efficient (Optimisé)). Toutefois, si le service virtuel n'a pas de réponse, la demande est alors automatiquement routée vers le système dynamique. Vous pouvez activer le mode Stand in uniquement pour un service virtuel comportant une étape d'appel dynamique. Le mode Stand In n'effectue pas de suivi spécial. Il permet simplement à un service virtuel de recourir au service dynamique.

Transaction Tracking (Suivi des transactions)

Ce mode déclenche davantage d'événements que le mode Most Efficient (Optimisé) et conserve en mémoire le flux de transaction dans les sessions. Ces informations transactionnelles sont utilisées pour définir le choix d'une réponse spécifique à une demande donnée. Ce mode ne fonctionne pas de manière aussi efficace que le mode Most Efficient (Optimisé). Le mode Transaction Tracking (Suivi des transactions) n'affiche pas les réponses du système dynamique ; il indique uniquement la réponse issue de l'image de service.

Live System (Système dynamique)

Ce mode utilise l'étape Live Invocation (Appel dynamique) du modèle pour déterminer une réponse à la demande actuelle. Au lieu d'utiliser la réponse issue du service virtuel, il accède au service dynamique pour obtenir la réponse. Le système cible de l'appel dynamique contrôle les performances. Ce mode est également appelé pass through (Authentification directe).

Bascullement

Le mode Failover (Bascullement) route d'abord une demande vers le système dynamique (le même que le mode Live System (Système dynamique)). Toutefois, si le système dynamique n'a pas de réponse, la demande est alors automatiquement routée vers le service virtuel. Ce mode est opposé au mode Stand In (Remplacer). Vous pouvez activer le mode Failover (Bascullement) uniquement pour un service virtuel comportant une étape d'appel dynamique. Le mode Failover (Bascullement) utilise l'image de service pour déterminer une réponse si l'étape d'appel dynamique échoue ; cela peut être le cas si le système dynamique n'était pas disponible.

Learning (Apprentissage)

Le mode Learning (Apprentissage) est similaire au mode Image Validation (Validation d'image), excepté qu'il répare ou corrige automatiquement le service virtuel pour obtenir la réponse nouvelle ou mise à jour à partir du système dynamique. La demande suivante transmise dans le service virtuel voit automatiquement la nouvelle réponse apprise. Une recherche de formation est effectuée dans les deux systèmes et le système dynamique est actuellement prioritaire.


Lorsqu'un service virtuel est exécuté en mode Learning (Apprentissage) et qu'il a acquis de nouvelles connaissances, une icône  apparaît à gauche du nom du service virtuel dans la console du VSE. L'icône reste visible jusqu'au redéploiement du service virtuel.

Image Validation (Validation de l'image)

Ce mode utilise le VSE et le système dynamique pour dériver une réponse vers la demande actuelle. Les réponses sont journalisées pour être appliquées ultérieurement à l'image de service à l'aide du panneau View Session and Tracking (Afficher la session et le suivi). Ce mode permet d'effectuer une comparaison dynamique entre les réponses fournies par le VSE et un système dynamique correspondant et, en cas de différences, de réparer l'image de service du VSE pour conserver la synchronisation avec le système dynamique. Ce mode est également appelé live healing (réparation dynamique). La validation d'image est le mode le moins efficace.

Dynamic (Dynamique)

Ce mode permet au modèle de déterminer le mode à utiliser pour chaque demande. Les performances sont, par conséquent, imprévisibles. La présence de l'étape VS Routing (Routage du service virtuel) dans le modèle après l'écoute du protocole d'une ou plusieurs étapes est la seule condition requise.

Reset the Transaction and Error Counts for the Selected Virtual Service (Réinitialiser le nombre de transactions et d'erreurs pour le service virtuel sélectionné)

Cette option définit le nombre de transactions et d'erreurs sur zéro pour le service virtuel sélectionné.

Stop the selected virtual service (Arrêter le service virtuel sélectionné)

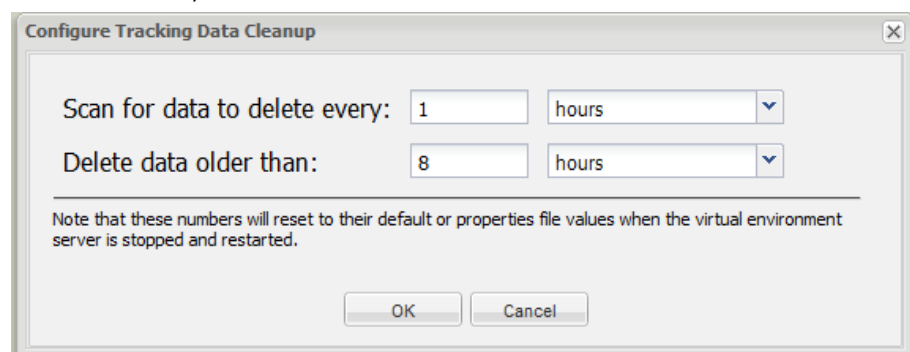
Pour arrêter le service virtuel sélectionné, cliquez sur ce bouton. L'application affiche un message de confirmation avant l'arrêt du service.

Remove the selected virtual service from the environment (Supprimer le service virtuel sélectionné de l'environnement)

Pour supprimer le service virtuel sélectionné de l'affichage de la console, cliquez sur ce bouton. L'application affiche un message de confirmation avant la suppression du service.

Configure tracking data cleanup (Configurer le nettoyage des données de suivi)

Ouvre la fenêtre Configure Tracking Data Cleanup (Configurer le nettoyage des données de suivi).



Saisissez les valeurs de nettoyage des données qui demeureront actives jusqu'à l'arrêt et le redémarrage du service. Lorsque le service redémarre, les valeurs sont réinitialisées sur leur valeur par défaut ou les valeurs définies dans un fichier de propriétés.

Scan for data to delete every (Rechercher des données à supprimer tou(te)s les)

Définit l'intervalle (en millisecondes, secondes, minutes, heures, jours ou semaines) après lequel rechercher des données de suivi à supprimer.

Delete data older than (Supprimer les données antérieures à)

Indique l'ancienneté des données (exprimée en millisecondes, secondes, minutes, heures, jours, semaines) à supprimer.

Shut down the whole virtual service environment (Arrêter l'environnement de service virtuel à tous les niveaux)

Pour arrêter l'environnement complet du VSE, cliquez sur ce bouton. Si vous répondez par l'affirmative au message de confirmation d'arrêt, le VSE s'arrête et la fenêtre correspondante se ferme.

Affichage de session et réparation de modèle

L'*affichage de session* permet à un utilisateur du VSE d'observer le comportement de sessions actuelles (ou récentes) sur un serveur du VSE. L'utilisateur peut déterminer le motif de la réponse à une demande spécifique. Cela permet également d'activer une comparaison dynamique entre les réponses fournies par le VSE et un système dynamique correspondant. En cas de différences, l'application peut utiliser la *réparation de modèle* pour corriger ou réparer l'image de service du VSE de sorte à préserver sa synchronisation avec le système dynamique.

L'affichage de session est uniquement disponible pour les services virtuels exécutés en mode Transaction Tracking (Suivi des transactions) ou Image Validation (Validation de l'image). La réparation de modèle est uniquement disponible pour les services virtuels exécutés en mode Image Validation (Validation de l'image). Pour plus d'informations sur les modes d'exécution, reportez-vous à la section Spécification du comportement du modèle sélectionné.

La réparation de modèle diffère du mode d'apprentissage, car ce dernier modifie immédiatement une image de service lorsque des différences entre les réponses de CA Service Virtualization et les réponses dynamiques sont détectées. La réparation journalise ces différences à des fins d'examen ultérieur et d'application de l'image de service avec le panneau d'informations View Session and Tracking (Affichage de sessions et suivi).

Vous pouvez gérer l'affichage de session et la réparation de modèle à l'aide d'un panneau accessible à partir du tableau de bord du VSE ou lors de la modification d'un modèle de service virtuel.

The screenshot displays the 'Virtual Service Environment VSE@Default' interface. It features a top navigation bar with 'Services', 'Metrics', 'kioskV5 Inspector', and 'kioskV5 Session Data'. The main content area is divided into two primary sections: 'Session Information' and 'Response Information'.

Session Information: This section contains a table with the following data:


Session ID	Created On	Modified On
Stateless	8:25:13	8:28:23
-92ac754b:4289c860a61:-2d54	8:28:21	8:28:21
-10c3833c:4474830fb53:-3878	8:25:10	8:25:10

Response Information: This section shows a single response entry: '8:50:16 getNewToken(username=lisa_simpson, password=golisa)'. Below this is an 'Update' button.

Request/Response Comparison: At the bottom, there are two columns: 'Virtual' and 'Live'. The 'Virtual' column shows a request '[3] getNewToken' with parameters 'username: demo1' and 'password: pass'. The 'Live' column shows the corresponding response 'getNewToken' with parameters 'username: lisa_simpson' and 'password: golisa'.

At the bottom of the interface, there are 'Refresh' and 'Update' buttons, and a checkbox for 'Auto Refresh' which is currently checked.

Pour afficher le suivi de session d'un service virtuel spécifique, dans l'onglet Services de la console du VSE, sélectionnez un service et cliquez sur Session/Tracking Information

(Informations de session/suivi ).

Lorsque vous cliquez sur l'icône, le panneau de session s'ouvre et contient la session et des transactions enregistrées. Si un service virtuel déployé ne contient aucune transaction enregistrée, le panneau de session ne contiendra aucune information sur la session.

Le panneau de sessions est divisé en deux volets : Session Information (Informations sur les sessions) et Response Information (Informations de réponse).

Volet Session Information (Informations sur les sessions)

Le volet Session Information (Informations sur les sessions) répertorie toutes les sessions pour le service virtuel sélectionné sous forme de tableau. Pour réorganiser le tableau, sélectionner ou effacer les colonnes affichées, cliquez sur la flèche à droite de chaque en-tête de colonne.

Session Status (Statut de la session)

Affiche une icône indiquant le statut de la session actuelle.

Une boule grise indique une session enregistrée en mode Transaction Tracking (Suivi des transactions), ou une session réparée à l'aide de la réparation de modèle.

Une boule rouge indique une session enregistrée en mode Image Validation (Validation de l'image) et qui est un candidat pour la réparation.

Une boule verte indique une session enregistrée en mode Image Validation (Validation de l'image) dans laquelle les réponses dynamiques et du VSE correspondent.

Session ID (ID de session)

Indique l'ID unique de chaque session.

Created On (Date de création)

Affiche l'horodatage de la première transaction dans cette session spécifique.

Modified On (Date de modification)

Affiche l'horodatage de la dernière transaction.

Txn Count (Nombre de transactions)

Indique le nombre de transactions enregistrées après le démarrage du service.

Client ID (ID du client)

(Spécifique à chaque protocole) Pour le protocole HTTP, il s'agit du terminal du client qui a soumis la transaction.

Most Recent Request (Demande la plus récente)

Indique la dernière demande passée dans cette session.

Volet Response Information (Informations de réponse)

Le volet Response Information (Informations de réponse) affiche la liste des transactions pour la session sélectionnée. Lorsque vous pointez la souris sur la boule de couleur dans la première colonne de ce volet, une info-bulle identifie la correspondance pour cette transaction. Si vous cliquez sur une transaction spécifique, les onglets de demande et de réponse s'affichent au bas du volet. Ces onglets comparent la demande-réponse entre le système du VSE et le système dynamique.

Dans le volet Response Information (Informations de réponse), les icônes suivantes représentent des transactions :

- Boule verte : indique une correspondance de signature sur une métatransaction.
- Yellow ball (Boule jaune) : indique une correspondance de signature dans une métatransaction. La navigation dans l'image est correcte, mais le corps de la réponse diffère entre le VSE et le système dynamique.
- Red ball (Boule rouge) : indique une transaction conversationnelle qui diffère entre le système dynamique et l'image du VSE.

Les boutons Update (Mettre à jour) permettent de mettre à jour l'image de service avec des transactions de session dynamique/sans état. Ce processus est appelé réparation de modèle. Vous utilisez la réparation de modèle pour supprimer la différence entre l'image du VSE et le système dynamique de sorte que le modèle de service virtuel fonctionne correctement. Lorsque vous cliquez sur Update (Mettre à jour), la boule rouge marquant la session devient grise. Cette session est désormais suivie.

- Le bouton Response Information Update (Mettre à jour les informations de réponse) met à jour l'image de service pour la transaction sélectionnée.
- Le bouton Session Information Update (Mettre à jour les informations sur la session) permet de sélectionner plusieurs sessions affichées dans le volet Service Information (Informations de service) et de les mettre à jour simultanément.

Mesures du VSE

Affichage des mesures du VSE

L'onglet Metrics (Mesures) de la console DevTest permet d'afficher les mesures du VSE.

Procédez comme suit:

1. Pour afficher une liste de services actifs, cliquez sur Select VSE Service (Sélectionner le service du VSE).
2. Sélectionnez le service à afficher.
3. Pour afficher les graphiques disponibles pour le service sélectionné, cliquez sur Select Chart (Sélectionner un graphique).
4. Sélectionnez un graphique et cliquez sur Generate Chart (Générer un graphique).

Pour actualiser l'affichage, cliquez sur Refresh (Actualiser). Utilisez les curseurs Zoom et Scroll (Défiler) pour déplacer le graphique et améliorer l'affichage de vos données.

Remarque : Pour afficher les graphiques de mesures, le module d'extension de Flash Adobe doit être installé sur le navigateur utilisé pour accéder à la console de serveur.

Définition d'une collecte de mesures

Pour définir les paramètres de collecte de mesures du VSE, mettez à jour les propriétés suivantes dans le fichier **lisa.properties** :

- lisa.vse.metrics.collect
- lisa.vse.metrics.txn.counts.level
- lisa.vse.metrics.sample.interval
- lisa.vse.metrics.delete.cycle
- lisa.vse.metrics.delete.age

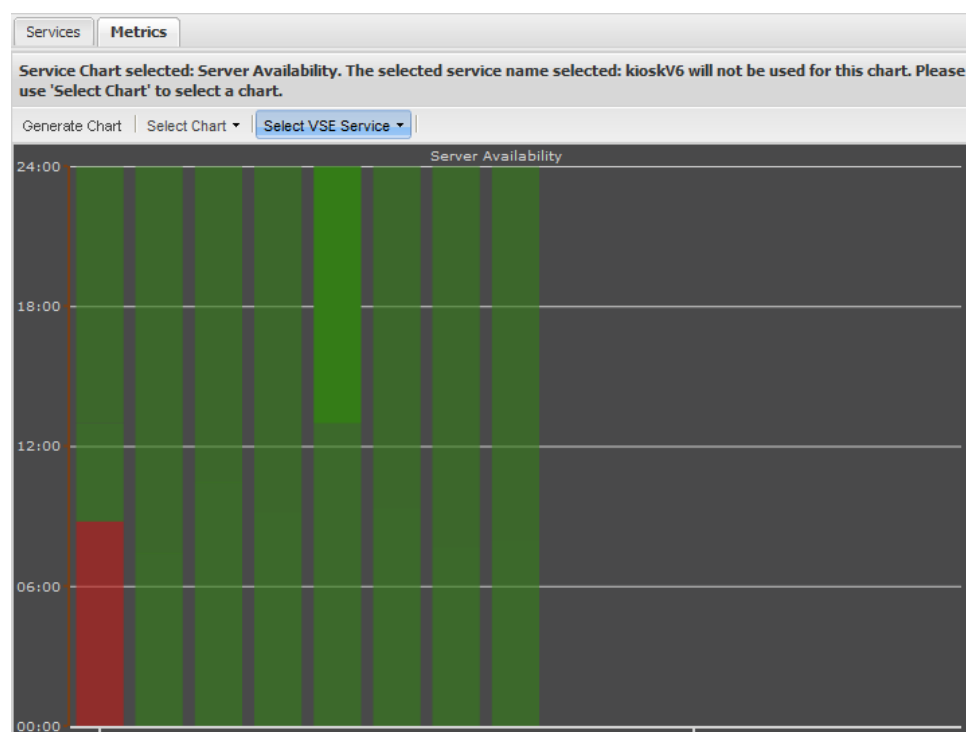
Remarque : Pour plus d'informations, reportez-vous à la section Annexe A - Fichier de propriétés LISA de la rubrique *Utilisation*.

Mesures de graphique de serveur

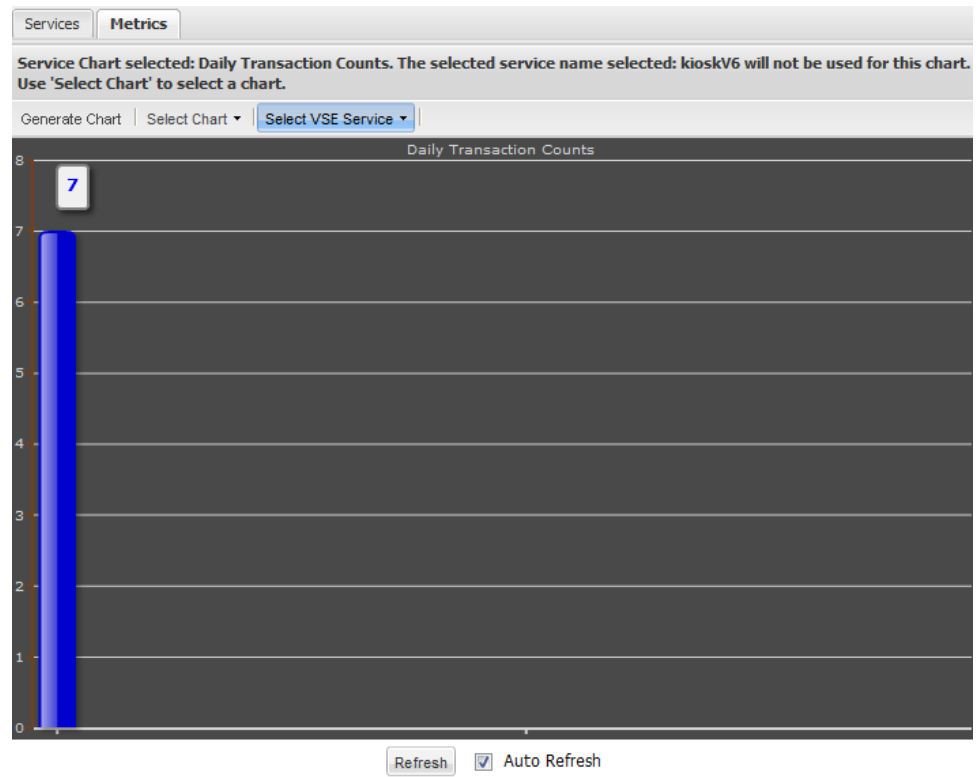
Les mesures de graphique de serveur s'appliquent à toutes les activités sur ce serveur virtuel, ou sur le VSE.

Remarque : Si vous sélectionnez un graphique de serveur, l'en-tête du panneau indiquera que le service virtuel sélectionné n'est pas utilisé pour ce graphique. Ces graphiques s'appliquent à l'ensemble du serveur, non au service sélectionné uniquement.

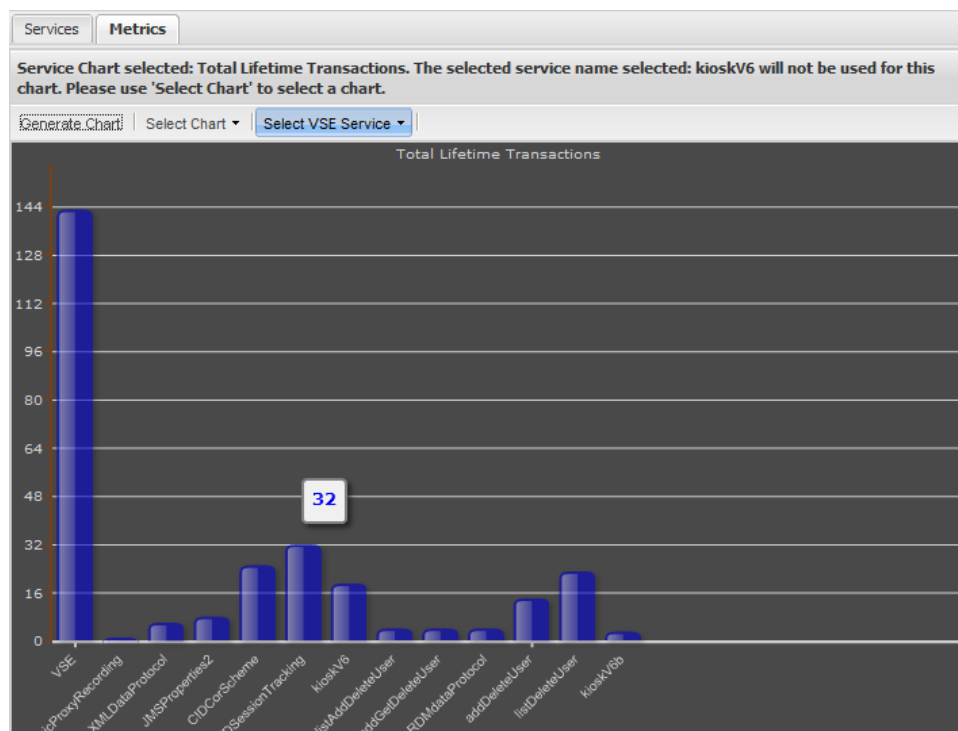
Server Availability (Disponibilité du serveur)



Daily Transaction Counts (Nombres de transactions quotidiennes)



Total Lifetime Transactions (Total des transactions pendant la durée de vie)



Mesures de graphique de service

Les mesures de graphique de service s'appliquent à chaque service virtuel. Le volet droit contient une liste de services virtuels déployés dans cet environnement. Vous pouvez sélectionner l'un d'eux, puis sélectionner un graphique dans la liste.

Response Time (Temps de réponse)



Forced Delay (Retard forcé)



Forced Delay (Retard forcé)

Indique la durée en millisecondes avant l'envoi d'un résultat par VSE.

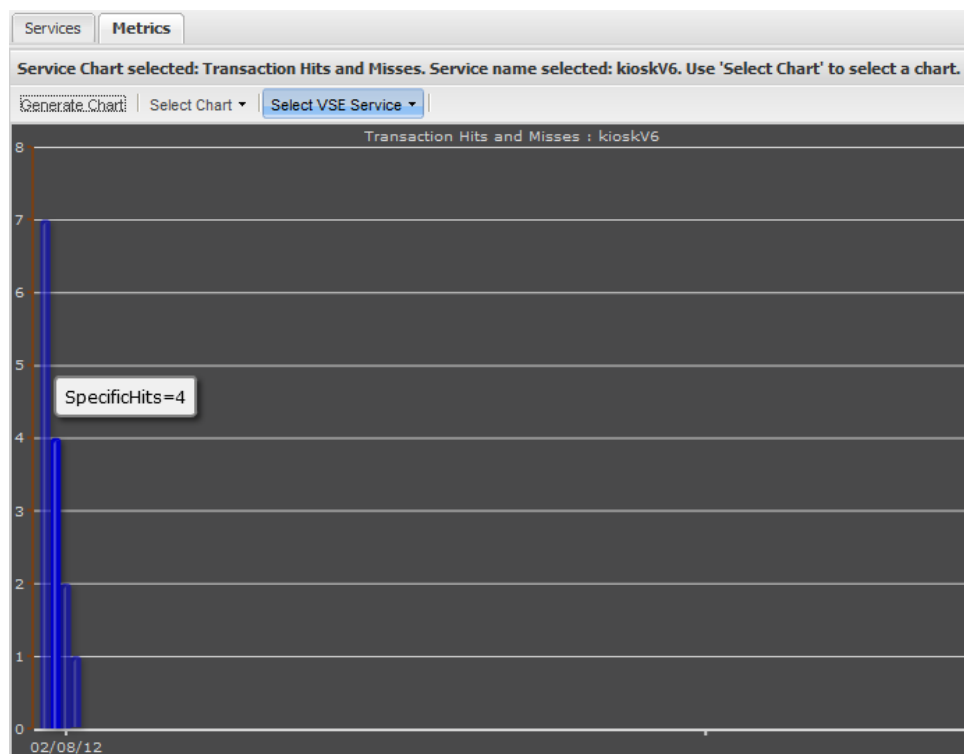
Un nombre positif indique que VSE a patienté délibérément avant d'envoyer un résultat en raison du délai de réflexion spécifié. Par exemple, si le délai de réflexion était de 10 ms et la durée d'écoute et de réponse était de 8 ms, VSE patiente pendant 2 ms.

Un nombre négatif indique que VSE a pris plus de temps pour générer le résultat que le délai de réflexion. Par exemple, le délai de réflexion était de 10 ms, mais la durée d'écoute était de 12 ms et celle de réponse était de 12 ms. Le rapport indique -2.

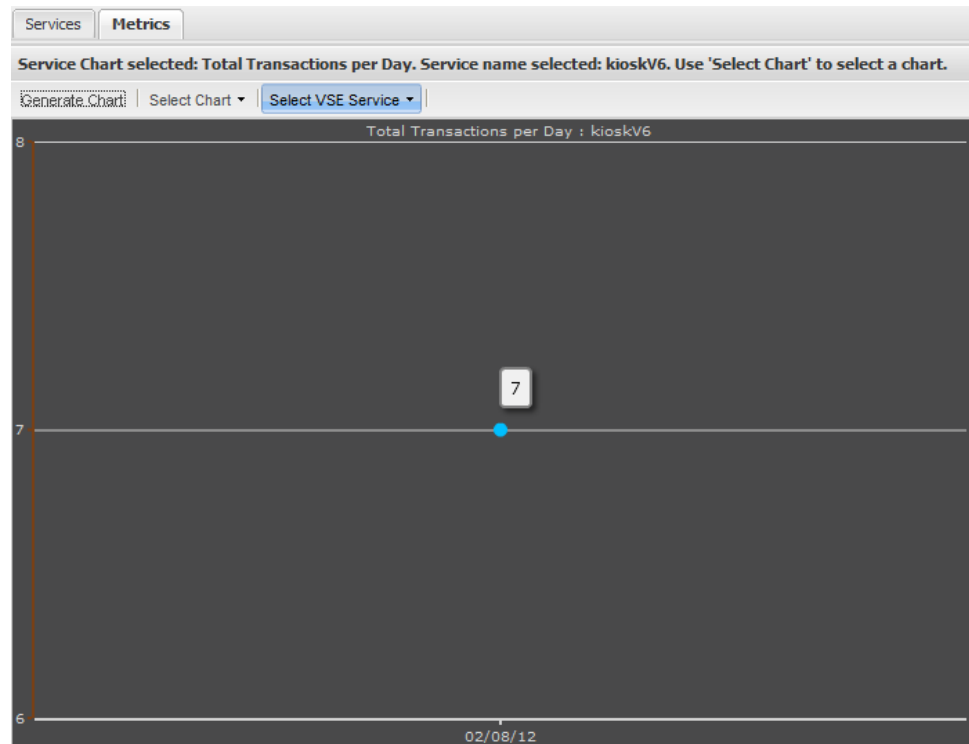
Transactions Per Second (Transactions par seconde)



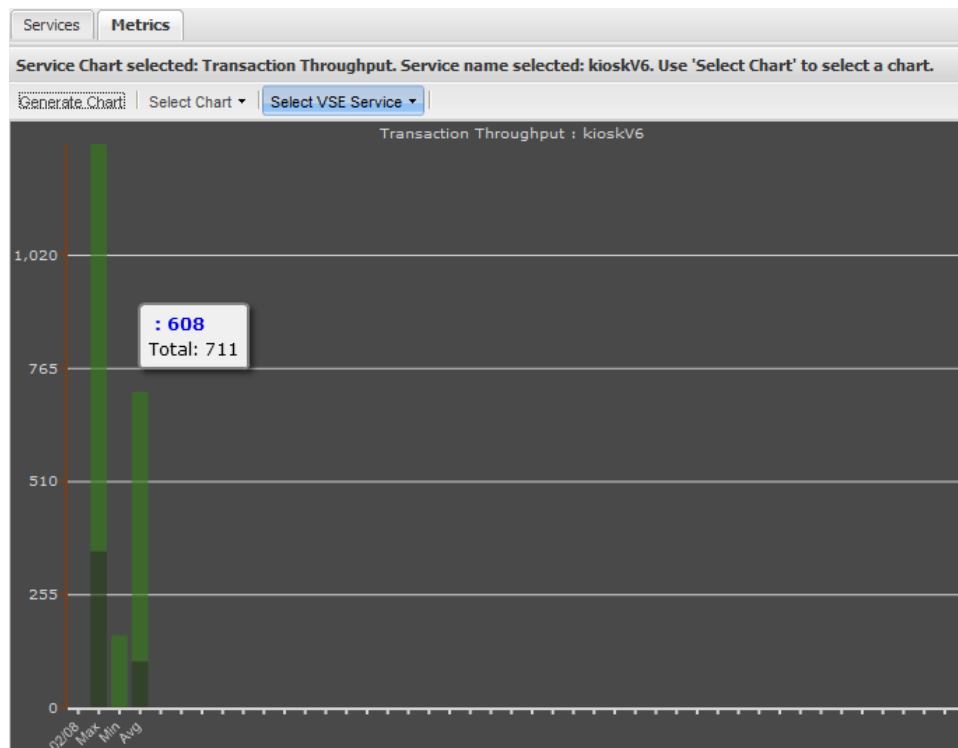
Transaction Hits and Misses (Transactions correspondantes et transactions non correspondantes)



Total Transactions per Day (Total de transactions par jour)



Transaction Throughput (Débit de transactions)



Chapitre 13: Gestionnaire de VSE : gestion et déploiement de services virtuels

Le gestionnaire de VSE est un module d'extension Eclipse qui vous permet de déployer et de gérer les services virtuels de DevTest.

Le gestionnaire de VSE est pris en charge sur Eclipse 4.3 et ses versions ultérieures. Exécutez Eclipse avec une machine virtuelle Java 7.

Ce chapitre traite des sujets suivants :

[Installation du gestionnaire de VSE](#) (page 411)

[Utilisation du gestionnaire de VSE](#) (page 412)

Installation du gestionnaire de VSE

Pour installer le gestionnaire du VSE, suivez la procédure Eclipse habituelle d'ajout de nouveau logiciel.

Procédez comme suit:

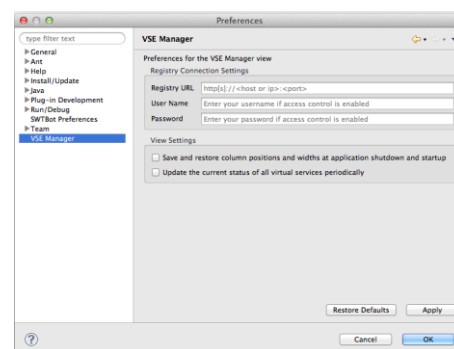
1. Dans Eclipse, sélectionnez Help (Aide), Install New Software (Installer un nouveau logiciel).
La boîte de dialogue Install (Installer) s'ouvre.
2. Sélectionnez Add (Ajouter) pour ajouter l'URL de référentiel de la plate-forme de provisionnement du VSE (p2) :
`http://www.itko.com/downloads/eclipse/8.0/updates/`
3. Sélectionnez la fonctionnalité VSE Manager UI (Interface utilisateur du gestionnaire de VSE), puis cliquez sur Next (Suivant).
La fenêtre Install Details (Informations d'installation) s'affiche.
4. Réviser les informations d'installation, puis cliquez sur Next (Suivant).
La fenêtre Review Licenses (Vérifier les licences) s'ouvre.
5. Consultez et acceptez les licences, puis cliquez sur Finish (Terminer).

Utilisation du gestionnaire de VSE

Configuration du gestionnaire de VSE

Avant d'utiliser le gestionnaire de VSE, vous devez configurer la page VSE Manager (Gestionnaire de VSE) dans la boîte de dialogue Eclipse Preferences (Préférences Eclipse).

Le graphique suivant présente la boîte de dialogue Preferences (Préférences) avec la page VSE Manager (Gestionnaire de VSE) sélectionnée.



Remarques :

- Définissez le champ Registry URL (URL du registre) sur une URL **https**. Ce type d'URL requiert l'activation des communications HTTPS avec la console DevTest, selon la procédure décrite dans la section *Administration*.
- Veillez à saisir des valeurs dans les champs User Name (Nom d'utilisateur) et Password (Mot de passe). Lorsque ces champs sont vides, la boîte de dialogue Eclipse Password Required (Mot de passe Eclipse requis) peut s'afficher. Cette boîte de dialogue ne correspond pas au gestionnaire de VSE. Si elle s'affiche, au lieu d'entrer un nom d'utilisateur et un mot de passe, cliquez sur Cancel (Annuler).

Procédez comme suit:

1. Pour configurer les paramètres de connexion au registre, entrez l'URL du serveur Web du VSE.
2. Le cas échéant, entrez votre nom d'utilisateur et votre mot de passe.
3. (Facultatif) Pour enregistrer l'ordre des colonnes et leurs largeurs associées lorsque vous fermez, puis rouvrez la vue, cochez la case Save and restore... (Enregistrer et restaurer).
4. (Facultatif) Pour que la vue actualise automatiquement le contenu toutes les 10 secondes, cochez la case Update the current status... (Mettre à jour le statut actuel).
5. Pour terminer et fermer la boîte de dialogue, cliquez sur OK.

Importation d'un projet LISA

Procédez comme suit:

1. A partir du menu principal, ouvrez l'assistant d'importation Eclipse.
 2. Dans la catégorie CA LISA, sélectionnez LISA Project (Projet LISA).
 3. Sélectionnez un projet LISA et entrez son nom dans le champ Project Name (Nom du projet).
 4. Cliquez sur Finish (Terminer).
- Le projet est importé.

Utilisation de la vue du gestionnaire de VSE

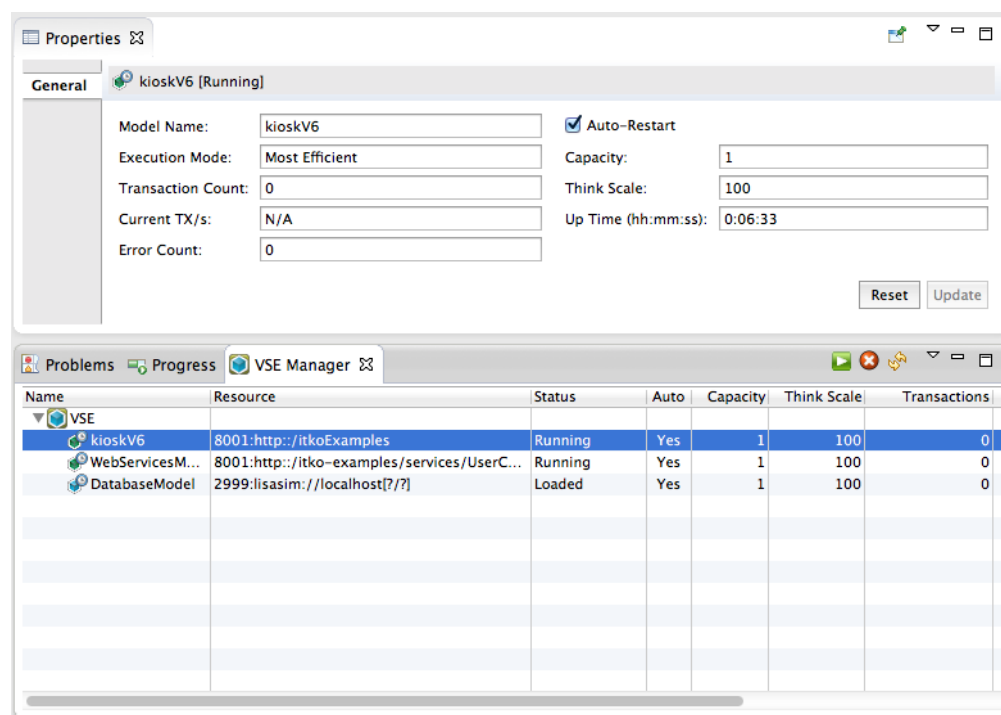
Name	Resource	Status	Auto	Capacity	Think Scale	Transactions
VSE						
kioskV6	8001:http://itkoExamples	Running	Yes	1	100	0
WebServicesM...	8001:http://itko-examples/services/UserC...	Running	Yes	1	100	0
DatabaseModel	2999:lisasim://localhost[?/?]	Loaded	Yes	1	100	0

La vue du gestionnaire de VSE prend en charge les actions suivantes :

- Glisser-déposer pour réorganiser les colonnes de la vue.
- Sélection des colonnes à masquer ou afficher à l'aide de la liste déroulante
- Démarrage, arrêt ou annulation du déploiement du service virtuel sélectionné
- Glisser-déposer à partir d'un produit Eclipse et à partir du système de fichiers natif pour le déploiement de nouveaux services virtuels vers VSE au format de fichier MARS

Propriétés d'intégration de vue

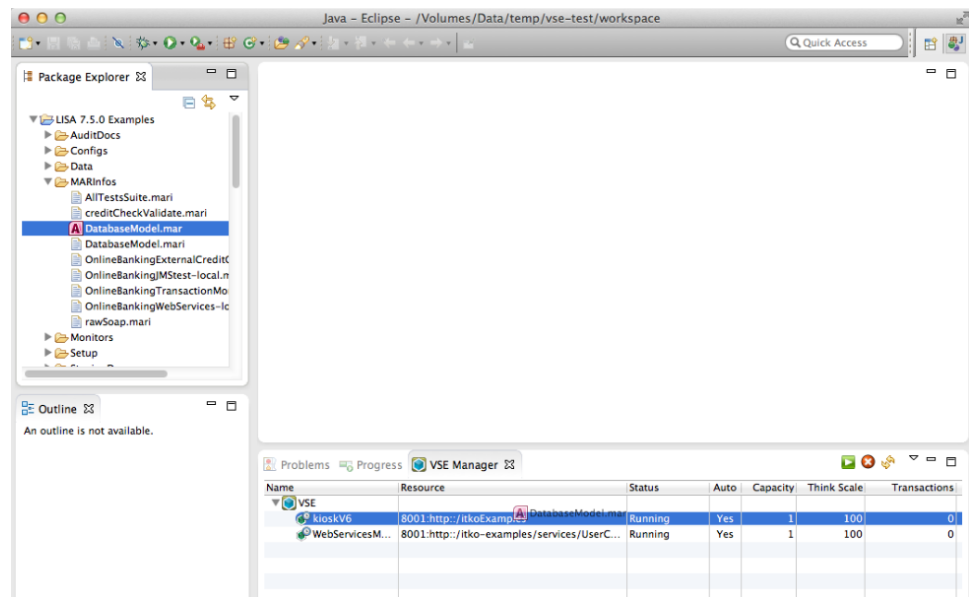
Lorsqu'un service virtuel est sélectionné dans la vue du gestionnaire de VSE et la vue Properties (Propriétés) est ouverte, vous pouvez afficher et mettre à jour des informations supplémentaires sur le service sélectionné.



Dans cette vue, vous pouvez :

- Contrôler automatiquement le démarrage du service lorsque le serveur de VSE est démarré.
- Modifier la capacité d'un service virtuel déployé.
- Modifier le pourcentage de délai de réflexion des demandes au service virtuel.

Vous pouvez faire glisser un fichier MAR vers le gestionnaire de VSE pour le déployer.



Chapitre 14: Commandes du VSE

Ce chapitre traite des sujets suivants :

[Commande du gestionnaire du VSE - gestion des environnements de service virtuel](#)
(page 418)

[Commandes ServiceManager - Gestion des services](#) (page 419)

[Commande ServiceImageManager - Gestion des images de service](#) (page 420)

[Commande VirtualServiceEnvironment](#) (page 425)

Commande du gestionnaire du VSE - gestion des environnements de service virtuel

Pour gérer les environnements de service virtuel, utilisez l'outil de ligne de commande du gestionnaire de VSE. Il est recommandé d'utiliser l'outil de ligne de commande pour LISA 6.0 et les dernières implémentations. Pour plus d'informations, consultez la section [Commande du gestionnaire de services](#) (page 419).

Le gestionnaire du VSE est fourni comme outil dans le répertoire bin de <lsrv>.

Les commandes du gestionnaire de VSE ont le format suivant :

```
VSEManager [--registry nom | --vse nom | --username=nom  
| --password=nom ]  
  
[--status [nom_service virtuel | ALL ] | --deploy fichier_archive |  
--redploy fichier_archive | --update nom_service virtuel [capacity  
capacité] [thinkscale délai] [auto-restart [ON|OFF] [grouptag  
balise_group] | --remove nom_service virtuel | --start nom_service  
virtuel | --set-exec-mode nom_service virtuel --mode mode_exécution  
| --stop nom_service virtuel | --help | --version
```

--registry *name*

Permet de définir le nom du registre de DevTest à utiliser.

Remarque : Le paramètre --registry doit être défini avant l'exécution d'une commande. Sinon, la commande sera par défaut exécutée sur le registre local. Si aucun registre local n'existe, la commande ne sera pas exécutée.

--vse *name*

Définit le nom de l'environnement de service virtuel à utiliser.

--username=*nom*

Permet de définir l'ID permettant l'authentification pour ACL.

--password=*nom*

Définit le mot de passe à utiliser pour l'authentification lorsque la liste de contrôle d'accès est utilisée.

--status [*vs-name* | ALL]

Permet d'imprimer le statut de l'un ou de la totalité des services virtuels dans l'environnement actuel.

--deploy *archive-file*

Permet de déployer un nouveau service virtuel dans l'environnement actuel. L'actif principal du fichier d'archive doit être un modèle de service virtuel.

--redploy *archive-file*

Permet de redéployer un service virtuel existant dans l'environnement actuel.

**--update *nom_service_virtuel* [capacity *capacité*] [--thinkscale *pourcentage*]
[--auto-restart [ON|OFF] [--grouptag *balise_groupe*]**

Permet de mettre à jour le paramètre de capacité, de pourcentage de délai de réflexion, de balise de groupe, de redémarrage automatique ou une combinaison de ces paramètres pour le service virtuel nommé.

--remove *vs-name*

Permet de supprimer le service virtuel nommé de l'environnement actuel.

--start *vs-name*

Permet de démarrer le service virtuel nommé dans l'environnement actuel.

**--set-exec-mode *vs-name* --mode [DYNAMIC | VALIDATION | LIVE | TRACK | EFFICIENT
| LEARNING | STAND_IN | FAILOVER]**

Permet de définir le mode d'exécution du service virtuel nommé dans l'environnement actuel.

--stop *vs-name*

Permet d'arrêter le service virtuel nommé dans l'environnement actuel.

--help

Affiche le texte d'aide.

version

Permet d'imprimer le numéro de version.

Commandes ServiceManager - Gestion des services

Utilisez l'outil de ligne de commande du gestionnaire de services pour surveiller, réinitialiser et arrêter les services DevTest. Les commandes s'appliquent à un registre DevTest, au simulateur ou au coordinateur et au serveur VSE. Le gestionnaire de services est inclus comme outil dans le répertoire bin/ de DevTest Server.

Pour plus d'informations, consultez la section ServiceManager (Gestionnaire de services) de la rubrique *Administration*.

Commande ServiceImageManager - Gestion des images de service

Pour importer des transactions (brutes ou de session) dans une image de service nouvelle ou existante, utilisez l'outil de ligne de commande ServiceImageManager de VSE. L'outil permet également de combiner deux ou plusieurs images de service. La commande ServiceImageManager se trouve dans le répertoire LISA_HOME\bin :

Vous pouvez contrôler l'enregistrement de trois façons : interactive, chronométrée, ou déconnectée. Dans les trois cas, les arguments **--vrs=** et **--si-file=** sont requis. De même, pour les styles interactif et déconnecté, vous pouvez spécifier **--go** pour ignorer l'attente du signal de démarrage et lancer l'enregistrement immédiatement.

- **Interactive (Interactif)** : le style Interactive est utilisé lorsque seul l'argument **--record** est spécifié. Lorsque l'enregistreur est prêt, il lance le processus d'enregistrement après que vous ayez appuyé sur la touche Enter (Entrée) de la console. Puis, l'enregistreur patiente jusqu'à ce que vous appuyez de nouveau sur Enter (Entrée) pour l'arrêter.
- **Timed (Chronométré)** : le style chronométré est utilisé lorsque les arguments **--record** et **--stop=** (et également, **--start=**) sont spécifiés.

Disconnected (Déconnecté) : le style déconnecté est utilisé lorsque les arguments **--record** et **--port=** sont spécifiés. Un écouteur est configuré pour permettre à l'argument **--signal** de transférer des signaux de contrôle à l'enregistreur. Après l'initialisation d'un enregistreur en mode déconnecté, vous pouvez utiliser cet outil, dans un processus distinct, pour le contrôler en envoyant des signaux de démarrage et d'arrêt sur le port. Ce contrôle requiert uniquement de spécifier les arguments **--signal=** et **--port=**.

Pour effectuer une importation, spécifiez le fichier de transaction de trafic brut ou de session à importer après l'argument **--import**. Pour contrôler le processus d'importation, utilisez les protocoles de transport, de données de demande et de réponse, ainsi que les tolérances de navigation. Pour utiliser l'argument **--vrs=** avec les arguments de protocole ou de tolérance, spécifiez d'abord les autres arguments. L'argument **--vrs=** réinitialise les éléments vers leur état initial. Pour spécifier le nom du fichier contenant l'image de service à importer, utilisez l'argument **--si-file**. Spécifiez au moins les arguments suivants :

--import=, **--si-file=** et soit **--vrs=** soit **--transport=**

Puis, vous pouvez spécifier les arguments supplémentaires suivants :

--request-data=, **--response-data=**, **--non-leaf=**, ou **--leaf=**,

Pour effectuer une combinaison, spécifiez l'image de service cible après l'argument **--combine**. Le nom du fichier défini par **--combine=** est l'image de service virtuel cible et les autres fichiers répertoriés sont les sources. Pour contrôler le processus de combinaison des transactions identiques et répertorier les noms de fichier des images sources à combiner dans l'image cible, utilisez l'argument **--favor**. Spécifiez au moins l'argument suivant :

--combine=

Vous pouvez également spécifier l'argument suivant :

--favor=

Les fichiers sources pour l'opération de combinaison sont répertoriés.

Remarque : Tout argument dont les valeurs contiennent des espaces (tels que les noms de protocoles) doit être placé entre guillemets. Par exemple : "**--import=my file.xml**" et **-i"my file.xml"**

La commande ServiceImageManager a le format suivant :

--help -h

Affiche le texte d'aide.

--record -d

Enregistre les transactions dans un fichier d'image de service.

--vrs=*recording-session-file* -v *recording-session-file*

Spécifie le fichier de session d'enregistrement qui contient toutes les informations sur la configuration de l'enregistrement ou de l'importation.

--go -G

Spécifie que l'attente d'un signal de démarrage sera omise pour les modes interactif ou déconnecté.

--start=*time-spec* -S *time-spec*

Indique que l'enregistreur lancera l'enregistrement après la durée spécifiée.

--stop=*time-spec* -E *time-spec*

Indique que l'enregistreur arrêtera l'enregistrement après la durée spécifiée. Spécifiée en millisecondes, cette durée dépend de l'heure à laquelle l'enregistrement a débuté.

--port=*port-number* -P *port-number*

Spécifie le port à utiliser pour le contrôle de l'enregistreur. Lorsque vous utilisez cette option avec **--record**, elle enregistre le port sur lequel l'enregistreur doit écouter pour le contrôle. Lorsque vous l'utilisez avec **--signal**, elle enregistre le port auquel le signal de démarrage-d'arrêt doit être envoyé.

--signal=start|stop -g start|stop

Spécifie le signal à envoyer à un enregistreur dans un processus différent. Cette commande requiert également l'argument **--port=**.

--import=raw/traffic-file -i raw/traffic-file

Importe le document XML de trafic brut ou de session spécifié dans un fichier d'image de service.

--transport=protocol -t protocol

Spécifie le protocole de transport à utiliser pendant une importation.

Valeurs :

- HTTP/S
- IBM MQ Series
- JMS
- Standard JMS (Déconseillé)
- Java
- TCP
- JDBC (Driver based) (JDBC (basé sur le pilote))
- CICS LINK <DPL>, DTP <MRO, LU6.1, LU6.2>
- CICS Transaction Gateway (passerelle de transaction) <ECI>
- DRDA
- Conn IMS
- RFC SAP via JCo
- Protocole IDoc JCo

Opaque Data Processing (Traitement de données opaques)

--request-data=protocol -r protocol

Spécifie le protocole de données côté demande à utiliser pendant une importation.

Options :

Web Services (SOAP) (Services Web (SOAP))

Web Services (SOAP Headers) (Services Web (en-têtes SOAP))

Web Services Bridge (Pont des services Web)

WS-Security Request (Demande WS-Security)

Request Data Manager (Gestionnaire de données de demande)

Request Data Copier (Copieur de données de la demande)

Auto Hash Transaction Discovery (Détection des transactions de hachage automatique)

Generic XML Payload Parser (Analyseur de charge utile XML générique)

Data Desensitizer (Désensibilisation des données)

Copybook Data Protocol (Protocole de données de copybook)

Delimited Text Data Protocol (Protocole de données de texte délimité)

Scriptable Data Protocol (Protocole de données prenant en charge les scripts)

DRDA Data Protocol (Protocole de données DRDA)

XML Data Protocol (Protocole de données XML)

CICS Copybook Data Protocol (Protocole de données de copybook CICS)

--response-data=protocol -R protocol

Spécifie le protocole de données côté réponse à utiliser pendant une importation.

Options :

WS-Security Response (Réponse WS-Security)

Data Desensitizer (Désensibilisation des données)

Copybook Data Protocol (Protocole de données de copybook)

CTG Data Protocol (Protocole de données CTG)

Delimited Text Data Protocol (Protocole de données de texte délimité)

Scriptable Data Protocol (Protocole de données prenant en charge les scripts)

DRDA Data Protocol (Protocole de données DRDA)

CICS Copybook Data Protocol (Protocole de données de copybook CICS)

--non-leaf=CLOSE | WIDE | LOOSE -n CLOSE | WIDE | LOOSE -n tolerance

Spécifie la tolérance de navigation par défaut à appliquer à tous les noeuds de conversation non terminaux créés.

Par défaut : WIDE

--leaf=CLOSE | WIDE | LOOSE -l CLOSE | WIDE | LOOSE -l tolerance

Spécifie la tolérance de navigation par défaut à appliquer à tous les noeuds de conversation terminaux créés.

Par défaut : LOOSE

--config=config-file -C config-file

En mode d'enregistrement, spécifie un fichier de configuration à utiliser.

--si-file=vsi-file -s vsi-file

Spécifie le nom du fichier d'image de service dans lequel importer des transactions. Si ce fichier n'existe pas, il sera créé. S'il existe, les transactions importées seront fusionnées avec l'image de service existante.

--vsm_file=vsm-file -m vsm_file

Spécifie le nom du fichier de modèle de service virtuel à créer pendant un enregistrement.

--combine=*target-vsi-file* -c *target-vsi-file*

Combine un ou plusieurs fichiers d'image de service dans le fichier d'image de service nommé. Les images de service sources sont utilisées par défaut, sauf si l'argument **favor** est spécifié.

--favor=source | target -f source | target

Spécifiez la méthode de combinaison des transactions similaires lors de la combinaison des images de service. Pour mettre à jour le côté cible à partir du côté source avec les transactions identiques (et autres données), spécifiez **source**. Pour conserver le côté cible tel quel, spécifiez **target**.

version

Permet d'imprimer le numéro de version.

La commande ServiceImageManager contient deux codes d'erreur :

- 1 - en cas d'exception d'argument (statut de sortie, paramètre incorrect)
- 2 - en cas d'échec général

Commande VirtualServiceEnvironment

La commande VirtualServiceEnvironment est utilisée pour gérer l'application du VSE. Le fichier exécutable est disponible dans le répertoire [LISA_HOME]\bin.

Cette commande possède le format suivant :

```
VirtualServiceEnvironment [-h|--help] [-n=name|-n=name]
[-m=registry-spec|--registry=registry-spec]
[-l=lab-name|--labName=lab-name] [-f|--force]
[-P=port-number|--port=port-number] [--version]
```

--help -h

Affiche le texte d'aide.

--name=name -n name

Définit le nom de service du serveur d'environnement.

Valeur par défaut : propriété système **lisa.vseName**. La valeur par défaut de la propriété système est **VSEServer**.

--registry=registry-spec -m registry-spec

Définit le registre auquel se connecter.

Exemple :

-m=tcp://localhost:2010/registry1

--labName=lab-name -l lab-name

Indique le nom du laboratoire à utiliser.

Valeur par défaut : Default (Par défaut)

--force -f

Force la connexion du serveur au registre DevTest et remplace tout objet déjà enregistré par le nom de service de l'environnement.

--port=port-number -P port-number

Spécifie le port de service vers lequel VSE effectue les publications. Il s'agit du même port que celui spécifié dans l'argument **--name**.

--username=username -u username

Spécifie le nom de sécurité DevTest de l'utilisateur.

--password=password -p password

Spécifie le mot de passe de sécurité DevTest de l'utilisateur.

version

Permet d'imprimer le numéro de version.

Chapitre 15: Propriétés VSE de l'agent Java

Les propriétés de configuration pour l'agent Java de DevTest comprennent plusieurs propriétés qui s'appliquent au VSE.

Vous pouvez configurer ces propriétés dans la fenêtre Agents du portail DevTest. Les propriétés s'affichent dans l'onglet Settings (Paramètres).

Mode de démarrage

Spécifie le mode de virtualisation au démarrage.

Valeurs :

- Passthrough (Authentification directe)
- Enregistrement
- Playback (Lecture)

Shallow recording (Enregistrement superficiel)

Spécifie d'effectuer des rappels uniquement dans les méthodes virtualisées de niveau supérieur (dans les trames de pile).

Maximum graph size (Taille maximum de graphique)

Définit la taille maximum (en octets) d'un graphique d'objet sérialisé au format XML.

Reply timeout (Délai d'expiration de réponse)

Spécifie l'intervalle maximum autorisé pour une réponse du VSE lors d'une lecture. Cette valeur doit dépasser le délai de réflexion lors de vos tests.

Omit by reference (Ignorer par référence)

Spécifie si les arguments modifiés par les méthodes reference et void doivent être enregistrés ou lus.

Valeurs :

- Si cette option est **sélectionnée**, n'enregistrez pas ou ne lisez pas des arguments modifiés par les méthodes reference et void. Si vous ne virtualisez aucune méthode void modifiant l'état des arguments, sélectionnez cette case à cocher. Aucune méthode void n'accède au VSE et revient immédiatement pour améliorer les performances.
- Si cette option est **désélectionnée**, enregistrez ou lisez des arguments modifiés par les méthodes reference et void.

Cache responses (Réponses de cache)

Spécifie de mettre en cache les objets de réponse clés selon leur représentation de chaîne pour ignorer l'annulation de la conversion.

Valeurs :

- Si cette option est **sélectionnée**, mettez en cache les objets de réponse clé selon leur représentation de chaîne. Lorsqu'elle est activée, cette propriété entraîne encore l'envoi d'une demande au VSE. Lorsque la charge utile XML est renvoyée vers l'agent, au lieu d'être désérialisée, l'objet est recherché dans un cache. La clé est en grande partie une représentation XML de la réponse. Ce paramètre est utile pour l'amélioration des performances, lorsque l'état des réponses n'est pas modifiable. La sélection de cette propriété peut augmenter l'utilisation de la mémoire.
- Si cette option est **désélectionnée**, ne mettez pas en cache les objets de réponse clé selon leur représentation de chaîne.

Activation de JIT

La compilation JIT de VSE consiste en la mise cache complète ou partielle de la logique modèle-image dans l'agent.

JIT threshold (Seuil du JIT)

Définit le nombre d'appels de méthode entraînant une compilation JIT du VSE.

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 : `getNewToken`, `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](#) (page 375) 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 e date magique : `{{=doDateDeltaFromCurrent("yyyy-MM-dd","10");/*2012-08-14*/}}`. Pour plus d'informations, consultez la section [Chaînes et dates magiques](#) (page 51) 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](#) (page 51) 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](#) (page 63) 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](#) (page 61) 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](#) (page 46) 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](#) (page 45) de la rubrique *Utilisation de CA Service Virtualization*.