

IBM Tealeaf Structure de journalisation mobile d'Android  
CX  
Version 9.0  
12 juin 2014

*IBM Tealeaf Structure de journalisation  
mobile d'Android CX Guide*



**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 77.

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

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

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

Vous pouvez également consulter les serveurs Internet suivants :

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

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

Cette édition s'applique à la version 9, édition 0, modification 0 de IBM Tealeaf Structure de journalisation mobile d'Android CX et à toutes les éditions et modifications ultérieures jusqu'à mention contraire dans les nouvelles éditions.

© Copyright IBM Corporation 1999, 2014.

# Table des matières

<b>Avis aux lecteurs canadiens . . . . .</b>	<b>v</b>	Paramètres du fichier de cache local . . . . .	36
<b>IBM Tealeaf Structure de journalisation mobile d'Android CX Guide . . . . .</b>	<b>vii</b>	Paramètres de publication . . . . .	37
<b>Chapitre 1. Installation et mise en oeuvre . . . . .</b>	<b>1</b>	Paramètres de masque. . . . .	38
Versions de structure client prises en charge dans cette documentation . . . . .	1	Paramètre de type de message de filtre . . . . .	38
Démarrage rapide de l'instrumentation de votre application Android . . . . .	1	Paramètres de cookie . . . . .	39
Contenus du package . . . . .	1	Paramétrage de la temporisation de session . . . . .	39
Utilisation du modèle d'application. . . . .	2	Paramètres de capture d'écran . . . . .	39
Exigences minimales. . . . .	2	Ne pas modifier les paramètres internes. . . . .	40
Configurations système requises. . . . .	3	<b>Chapitre 3. Echantillon d'applications</b>	<b>41</b>
Incidence sur les ressources de périphérique. . . . .	4	<b>Chapitre 4. Instructions</b>	<b>43</b>
Modifications de projet Android . . . . .	4	<b>Chapitre 5. Référence</b>	<b>45</b>
Préparation à la mise en production . . . . .	5	Classe UICActivity . . . . .	45
Intégration à l'aide des classes Android IBM Tealeaf étendues . . . . .	5	Classe UICApplication. . . . .	48
Intégration à l'aide de classes Android sans les classes IBM Tealeaf étendues . . . . .	7	Classe Tealeaf . . . . .	49
Consignation de la disposition d'écran pour la relecture de sessions d'application mobile . . . . .	10	Classe TLDefaultHttpClient . . . . .	58
Intégration pour les applications Apache Cordova, PhoneGap et IBM Worklight à l'aide des classes Android sans les classes IBM Tealeaf . . . . .	13	Classe TLHttpRequestInterceptor . . . . .	60
Implémentation de screenViews . . . . .	15	Classe TLHttpResponseInterceptor. . . . .	61
Configuration de base . . . . .	16	Classe UICWebView . . . . .	62
Confidentialité des données . . . . .	16	Classe UICWebChromeClient . . . . .	64
Configuration de la mise en sessions pour les applications Android sur le client . . . . .	16	Classe UICWebViewClient . . . . .	65
Configurez les requêtes dans une application Android . . . . .	18	<b>Chapitre 6. Code d'échantillon</b>	<b>67</b>
Utilise un ID de session non IBM Tealeaf . . . . .	21	Comment instrumenter les contrôles TextView. . . . .	67
Applications hybrides . . . . .	22	Comment instrumenter les contrôles	
Démarrage rapide pour la configuration du serveur	25	ExpandableListView . . . . .	67
Page cible pour la capture de trafic . . . . .	25	Comment instrumenter les contrôles SlidingDrawer	68
Gestion du volume de trafic. . . . .	26	Comment masquer les contrôles . . . . .	68
Application de capture passive CX vérification de capture de trafic . . . . .	26	Fonction d'échantillonnage KillSwitch côté serveur	68
Options de contrôle des captures et du traitement	28	Fonction d'échantillonnage pour ASPX . . . . .	69
Configuration de la mise en sessions pour les applications Android dans IBM Tealeaf . . . . .	28	Fonction d'échantillonnage pour JSP . . . . .	69
Configuration d'exécution . . . . .	32	Fonction d'échantillonnage pour PHP. . . . .	71
IBM Tealeaf événements pour Structure de journalisation mobile d'Android CX . . . . .	32	<b>Chapitre 7. Traitement des incidents</b>	<b>73</b>
Mise à niveau de Structure de journalisation mobile d'Android CX . . . . .	32	Conseils relatifs à l'utilisation de résolution. . . . .	73
<b>Chapitre 2. Fichier de configuration</b>	<b>35</b>	Inclusion de requête brute et d'en-têtes de réponse. . . . .	73
Paramètres de niveau de consignation . . . . .	35	Gestion des problèmes client . . . . .	73
Paramètres de commutateur d'arrêt . . . . .	35	Exceptions ou pannes . . . . .	73
Paramètres de fichier de cache local . . . . .	36	Pannes d'alimentation . . . . .	74
Paramètres de publication . . . . .	37	Commutateur d'arrêt . . . . .	74
Paramètres de masque. . . . .	38	Problèmes de réseau . . . . .	74
Paramètre de type de message de filtre . . . . .	38	Mémoire faible ou stockage local . . . . .	74
Paramètres de cookie . . . . .	39	<b>Chapitre 8. Documentation et aide d'IBM Tealeaf</b>	<b>75</b>
Paramétrage de la temporisation de session . . . . .	39	<b>Remarques</b>	<b>77</b>
Paramètres de capture d'écran . . . . .	39	Marques . . . . .	79
Ne pas modifier les paramètres internes. . . . .	40	Remarques sur les règles de confidentialité. . . . .	79



---

## Avis aux lecteurs canadiens

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

### Illustrations

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

### Terminologie

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

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

### Claviers

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

### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

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

### Nomenclature

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

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

### Brevets

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

### Assistance téléphonique

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

---

# IBM Tealeaf Structure de journalisation mobile d'Android CX Guide

Le IBM Tealeaf Structure de journalisation mobile d'Android CX pour les applications mobiles natives requiert la licence IBM Tealeaf CX Mobile pour applications mobiles.

Pour plus d'informations, contactez votre interlocuteur IBM Tealeaf. Les licences doivent être implémentées dans leur code d'application fourni par IBM Tealeaf. Pour plus d'informations sur le téléchargement d'IBM Tealeaf , voir IBM® Passport Advantage en ligne.

Le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX* fournit des instructions sur la manière d'activer la capture de données d'applications mobiles directement sur l'application installée sur le périphérique Android du visiteur.

**Remarque :** Utilisez la dernière version du logiciel IBM Tealeaf Structure de journalisation mobile d'Android CX dès que possible.



---

## Chapitre 1. Installation et mise en oeuvre

Le IBM Tealeaf Structure de journalisation mobile d'Android CX pour les applications mobiles natives requiert la licence IBM Tealeaf CX Mobile pour applications mobiles.

Pour plus d'informations, contactez votre représentant IBM Tealeaf. Les licences doivent être implémentées dans leur code d'application fourni par IBM Tealeaf. Pour plus d'informations sur le téléchargement d'IBM Tealeaf , voir IBM Passport Advantage en ligne.

---

### Versions de structure client prises en charge dans cette documentation

Les instructions d'installation et de mise en oeuvre dans ce guide s'appliquent à la version à étapes de la messagerie JSON à partir de cette structure client.

La version à étapes de la messagerie JSON à partir de cette structure client a été introduite avec la version 8.5.

Pour les versions 8.5 et ultérieures, IBM Tealeaf continue de prendre en charge la méthode existante de soumission des données à partir des structures client, qui ont entraîné la scission de données soumises en occurrences individuelles dans le pipeline Windows.

**Remarque :** Dans une version ultérieure, la méthode de traitement de scission d'occurrences des données soumises à partir de structures client sera certainement obsolète.

Les instructions d'installation et de mise en oeuvre pour la version existante sont similaires mais nécessitent une configuration supplémentaire dans le pipeline Windows. Pour plus d'informations concernant les versions antérieures, notamment les étapes de mise en oeuvre, utilisez les liens qui suivent.

---

### Démarrage rapide de l'instrumentation de votre application Android

Consultez les sections suivantes pour plus d'informations concernant l'instrumentation de votre application pour que Structure de journalisation mobile d'Android CX puisse effectuer une capture de l'interface utilisateur et des événements d'application.

#### Contenus du package

Un seul fichier contient Structure de journalisation mobile d'Android CX et ses composants logiciels.

IBM Tealeaf Structure de journalisation mobile d'Android CX est distribué dans IBM Tealeaf Structure de journalisation mobile d'Android CX 8.8 - iOS Logging Framework for Windows sur IBM Passport Advantage en ligne.

Le package contient les composants logiciels suivants.

- KillSwitch. Code pour implémenter le gestionnaire de trafic du commutateur d'arrêt pour différentes technologies de serveur.
  - ASPX :
    - killswitch.aspx : page avec logique.
    - web.config : fichier de configuration utilisé par la page.
  - JSP :
    - killswitch.jsp : page avec logique.
    - config.properties : fichier de configuration utilisé par la page.
  - PHP
    - killswitch.php : page avec logique.
    - config.ini : fichier de configuration utilisé par la page.
- UICAndroid:
  - uicandroid.jar : fichier JAR de bibliothèque Android qui contient Structure de journalisation mobile d'Android CX.
  - TLFConfigurableItems.properties : fichier de configuration.
- SampleCode : contient les versions suivantes d'un échantillon d'application Android.
  - UICSP\_Clean : une application Android sans IBM Tealeaf Structure de journalisation mobile d'Android CX intégré.
  - UICSP\_ManualLog : une application Android avec IBM Tealeaf Structure de journalisation mobile d'Android CX intégré
  - UICSP\_ManualLog\_ServerSessionID : une application Android avec IBM Tealeaf Structure de journalisation mobile d'Android CX intégré à l'aide d'un ID de session fourni par une application Web.

Voir "Sample Code" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

## Utilisation du modèle d'application

Vous pouvez déployer le modèle d'application fourni par IBM Tealeaf pour tester les capacités et mesurer les effets de Structure de journalisation mobile d'Android CX.

Au lieu d'intégrer Structure de journalisation mobile d'Android CX avec votre application en développement, vous pouvez déployer le modèle d'application et compléter les étapes nécessaires à la configuration sur le reste de cette page pour commencer à capturer les données d'application mobile dans votre instance de IBM Tealeaf.

Voir Chapitre 3, «Echantillon d'applications», à la page 41.

## Exigences minimales

Développer les applications Android avec un niveau 8 API minimum, ce qui correspond à 2.2 (Froyo).

Consultez le Google Android Dev Center pour toute la documentation et les outils techniques Android.

Les structures de client IBM Tealeaf ne prennent pas en charge la transmission des données d'application à des systèmes de parties tiers. Les données d'application doivent être transmises au serveur qui hôte l'application native.

## Manifeste d'application Android

Le fichier manifeste pour l'application Android donne le droit aux articles suivants d'activer le mode écoute sur les actions de l'utilisateur dans l'application.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.SET_DEBUG_APP" />
```

## Configurations système requises

Pour développer les applications Android avec le kit de développement de logiciels Android, suivez ces configurations système et logiciel requises.

### Systèmes d'exploitation pris en charge

Les versions prises en charge des systèmes d'exploitation Windows, Mac et Linux sont les suivantes.

- Windows XP (32 bits), Vista (32 ou 64 bits) ou Windows 7 (32 ou 64 bits)
- Mac OS X 10.5.8 o version supérieure (x86 uniquement)
- Linux (testé sur Ubuntu Linux, Lucid Lynx)
  - GNU C Library (glibc) 2.7 ou version supérieure nécessaire.
  - Sur Ubuntu Linux, version 8.04 ou version supérieur nécessaire.
  - Les distributions 64 bits doivent pouvoir exécuter des applications 32 bits. Pour plus d'informations concernant la prise en charge d'applications de 32 bits, consultez les notes d'installation Ubuntu Linux .

### Environnements de développement suggérés

L'environnement de développement intégré Eclipse peut être utilisé pour déployer Structure de journalisation mobile d'Android CX.

### Eclipse IDE 3.6 (Helios) ou version supérieure

**Remarque :** Eclipse 3.5 (Galileo) n'est pas pris en charge par la dernière version d'Android Development Tools.

- Plug-in d'outils de développement Java Eclipse (inclus dans la plupart des packages Eclipse IDE).
- Pour plus d'informations à propos du téléchargement, de l'installation ou de la mise à jour d'Eclipse, voir <http://www.eclipse.org/downloads/>.

Différents types de packages Eclipse sont disponibles pour chaque plateforme. Pour le développement d'applications Android, installez un de ces packages.

- Eclipse IDE pour les développeurs Java™
  - Java version 1.6.
  - Java version 1.7 peut être utilisés en mode compatibilité.
- Eclipse Classic
- Eclipse IDE pour les développeurs Java EE
  - Kit Java Development 5 ou kit Java Development 6 (l'environnement d'exécution Java seul ne suffit pas).
  - Plug-in Android Development Tools
  - Non compatible avec le compilateur GNU pour Java (gcj)

## Autres environnements de développement

Structure de journalisation mobile d'Android CX peut être déployé par d'autres environnements de développement. Un développeur qui possède de solides connaissances peut appliquer les exemples répertoriés ci-dessous depuis Eclipse vers d'autres environnements, comme ant.

## Incidence sur les ressources de périphérique

Dans les test de performances Structure de journalisation mobile d'Android CX a les effets suivants sur les ressources du périphérique du visiteur.

- 2-3% de consommation de mémoire supplémentaire
- Effet minimum sur la durée de vie de la batterie

## Modifications de projet Android

Une fois que vous avez acquis IBM Tealeaf Structure de journalisation mobile d'Android CX, complétez les étapes suivantes pour installer les bibliothèques Structure de journalisation mobile d'Android CX dans un projet d'application Android.

Votre projet Eclipse doit inclure les structures suivantes. Test de Structure de journalisation mobile d'Android CX associé à Android 2.2 à 4.3.3.

### Installation de UICAndroid.jar

Utilisation d'Eclipse pour installer uicandroid.jar :

Suivez ces instructions pour utiliser l'environnement de développement Eclipse intégré pour installer uicandroid.jar dans une application Android.

1. Dans Eclipse, ouvrez l'application Android à instrumenter.
2. Placez uicandroid.jar dans le dossier lib.
3. Effectuez un clic droit sur uicandroid.jar et sélectionnez **Build Path**.
4. Cliquez sur **Add on Build Path**.

Installation de uicandroid.jar dans un autre environnement :

Dans des environnements de développement intégrés autre que Eclipse, ajoutez UICAndroid.jar au chemin de génération de l'application que vous souhaitez instrumenter.

### Installation de TLFFConfigurableItems.properties

Pour installer le fichier TLFFConfigurableItems.properties, placez-le dans le dossier assets de l'application Android.

### Auto-instrumentation non prise en charge

Android permet d'utiliser un gestionnaire à la fois pour n'importe quel objet. Par conséquent, Structure de journalisation mobile d'Android CX ne peut pas effectuer d'auto-instrumentation d'objets.

Vous devez appliquer l'instrumentation dans le cadre du développement de votre application.

## Préparation à la mise en production

Pour améliorer les performances de la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX, définissez les configurations ci-après dans la bibliothèque.

Dans le fichier `TLFConfigurableItems.properties`,

- Affectez à `DisplayLogging` la valeur `false` car vous n'avez pas besoin de messages logcat supplémentaires.
- Assurez-vous de disposer d'un commutateur d'arrêt activé et d'une URL opérationnelle.
- Spécifiez un niveau de consignation correct pour la production, si, en phase de développement, sa valeur était plus élevée.
- Assurez-vous que les valeurs de mise en sessions sont ajustées pour la page cible du serveur de production.
- Affectez à `GetImageDataOnScreenLayout` la valeur `false` pour désactiver la capture des données base64 qui augmente le contenu.

## Intégration à l'aide des classes Android IBM Tealeaf étendues

Pour cette méthode, les développeurs étendent à partir de la classe de la bibliothèque de capture IBM Tealeaf qui s'étend à partir de la bibliothèque de structure Android.

### Modifications de code du fichier `android.app.Application`

le fichier d'application gère le cycle de vie d'une application Android. IBM Tealeaf gère la bibliothèque en écoutant `onLowMemory` pour désactiver la bibliothèque si vous obtenez un avertissement, `onTerminate` pour nettoyer la bibliothèque et `onCreate` pour initialiser la bibliothèque. IBM Tealeaf vous recommande cette meilleure pratique.

### Localiser le fichier qui s'étend à partir de `android.app.Application`

Pour localiser le fichier qui s'étend à partir de `android.app.Application`:

1. Ouvrez le fichier Java existant qui s'étend à partir de la classe d'application. Si ce fichier n'existe pas, vous devez le créer et lui demander d'écouter le cycle de vie complet d'une application Android pour contrôler la bibliothèque et consigner les informations nécessaires. Vous devez également modifier le fichier pour qu'il s'étende à partir de `com.tl.uic.app.UICApplication` au lieu de `android.app.Application`.
2. Ajoutez les importations suivantes.
  - a. `import com.tl.uic.Tealeaf;`
  - b. `import com.tl.uic.app.UICApplication;`
3. Dans la méthode `onCreate()`, ajoutez `Tealeaf.enable()` qui initialise la capture d'actions utilisateur dans l'application.
4. Ajustez `AndroidManifest.xml` pour indiquer la classe d'application. Par exemple, si votre classe d'application s'intitule `MyApplication`, vous pouvez ajouter `android:name=".MyApplication"` dans le noeud `<application>`.
5. Ajoutez les droits suivants dans `AndroidManifest.xml`.

```
<uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.SET_DEBUG_APP" />
```

## Exemple

```
import com.tl.uic.Tealeaf;
import com.tl.uic.app.UICApplication;

public class MyApplication extends UICApplication {
    @Override
    public void onCreate() {
        super.onCreate();
        Tealeaf.enable();
    }
}
```

## Modifications de code du fichier android.app.Activity

Le fichier d'activité gère le cycle de vie d'une page dans une application Android native similaire aux tâches d'une page dans une application Web.

IBM Tealeaf écoute les événements suivants `onPause`, qui ont lieu lorsque l'application se met en arrière-plan, `onResume`, ce qui a lieu lorsque l'application se met en premier plan et `onDestroy` lorsque l'activité n'est plus en mémoire et collecte les restes.

## Localiser le fichier qui s'étend à partir de android.app.Activity

Pour chaque fichier d'activité que vous souhaitez consigner, étendez-la à l'aide de `UICActivity`. Cela permet d'étendre l'activité de base de la structure Android. `UICActivity` ajoute des fonctionnalités requises par la bibliothèque de structure de consignation IBM Tealeaf afin d'activer et de désactiver les tâches asynchrones, et d'effectuer des captures d'écran du périphérique après sa création.

Pour localiser le fichier qui s'étend à partir de `android.app.Activity` :

1. Ouvrez le fichier Java existant qui s'étend de la classe `android.app.Activity`, et modifiez-le afin de l'étendre de `com.tl.uic.app.UICActivity` au lieu de `android.app.Activity`.
2. Ajoutez les importations suivantes.
  - a. Importation `com.tl.uic.Tealeaf`;
  - b. Importation `com.tl.uic.app.UICApplication`;
3. Dans la méthode `onCreate()` :
  - a. Ajoutez `this.setTakeSnapshotAfterCreate(true);` //Pour activer les captures d'écran automatiques.

**Remarque :** Pour éviter de capturer des données privées potentielles, Structure de journalisation mobile d'Android CX prend des captures d'écran dès que l'image apparaît sur le périphérique. Par conséquent, aucun champ défini par l'utilisateur n'est rempli dans une image de capture d'écran.

Android ne prend pas en charge la capture des fenêtres en incrustation.

Pour les applications hybrides, les captures d'écran peuvent être manquantes ou dans le désordre en raison de problèmes de délai.

**Remarque :** La méthode mentionnée ci-dessus permet d'effectuer des captures d'écran automatique à partir de l'application client. Si vous n'activez pas cet élément à l'aide de `UICActivity`, vous pouvez effectuer des captures d'écran manuellement à l'aide de la structure de consignation. Voir "Reference" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

- b. Ajoutez `setLogicalPageName("LoginPage")` //Recommandé pour identifier la page.
- c. Ajoutez `setImageBackground(-16777216)` //Pour définir un arrière-plan noir pour la capture d'écran car l'arrière-plan est transparent.  
La valeur ci-dessus pour la couleur de l'arrière-plan peut être remplacée par n'importe quelle constante de couleur pour définir la couleur de l'arrière-plan de vos captures d'écran.

### Exemple

```

Importation com.tl.uic.app.UICActivity;
La classe publique LoginActivity étend UICActivity {
    @Override
    Chaîne publique vide onCreate(Bundle savedInstanceState) {
        this.setTakeSnapshotAfterCreate(true); //Pour activer les captures d'écran automatiques
        setLogicalPageName("LoginPage") //Recommandé pour identifier la page
        setImageBackground(-16777216) //Pour définir l'arrière-plan de
Capture d'écran
        super.onCreate(savedInstanceState);

```

## Intégration à l'aide de classes Android sans les classes IBM Tealeaf étendues

Pour cette méthode, des développeurs ajoutent des fragments de code qui aident la bibliothèque de capture IBM Tealeaf.

### Modifications de code du fichier `android.app.Application`

Le fichier d'application gère le cycle de vie d'une application Android. IBM Tealeaf gère la bibliothèque en écoutant `onLowMemory` pour désactiver la bibliothèque si vous obtenez un avertissement, `onTerminate` pour nettoyer la bibliothèque et `onCreate` pour initialiser la bibliothèque. IBM Tealeaf vous recommande cette meilleure pratique.

### Localiser le fichier qui s'étend à partir de `android.app.Application`

Pour localiser le fichier qui s'étend à partir de `android.app.Application`:

1. Ouvrez le fichier Java existant qui s'étend à partir de la classe `android.app.Application`. Si ce fichier n'existe pas, vous devez le créer et lui demander d'écouter le cycle de vie complet d'une application Android pour contrôler la bibliothèque et consigner les informations nécessaires.
2. Ajoutez les importations suivantes.
  - a. `import com.tl.uic.Tealeaf;`
3. Dans `onCreate()`:
  - a. Ajoutez `Tealeaf tealeaf = new Tealeaf(this);`, qui initialise la bibliothèque IBM Tealeaf avec une référence à l'application instrumentée.
  - b. Ajoutez `Tealeaf.enable();` qui initialise la capture d'actions utilisateur dans l'application.
4. Dans `onLowMemory()`:
  - a. Ajoutez `Tealeaf.onLowMemory();` avant `super` pour qu'il puisse ajuster la bibliothèque en raison d'une mémoire faible.
5. Dans `onTerminate()`:
  - a. Ajoutez `Tealeaf.disable();` avant `super` pour qu'il puisse désactiver la bibliothèque.

6. Ajustez `AndroidManifest.xml` pour indiquer la classe d'application. Par exemple, si votre classe d'application s'intitule `MyApplication`, vous pouvez ajouter `android:name=".MyApplication"` dans le noeud `<application>`.
7. Ajoutez les droits suivants dans `AndroidManifest.xml`.

```
<uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.SET_DEBUG_APP" />
```

### Exemple

```
import android.app.Application;
import com.tl.uic.Tealeaf;

public class MyApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        Tealeaf tealeaf = new Tealeaf(this);
        Tealeaf.enable();
    }

    @Override
    public void onLowMemory() {
        Tealeaf.onLowMemory();
        super.onLowMemory();
    }

    @Override
    public void onTerminate() {
        Tealeaf.disable();
        super.onTerminate();
    }
}
```

### Modifications de code du fichier `android.app.Activity`

Le fichier d'activité gère le cycle de vie d'une page dans une application Android native similaire aux tâches d'une page dans une application Web.

IBM Tealeaf écoute les événements suivants `onPause`, qui a lieu lorsque l'application se met en arrière-plan, `onResume`, qui a lieu lorsque l'application se met en premier plan et `onDestroy` lorsque l'activité n'est plus en mémoire et collecte les restes.

### Localiser le fichier qui s'étend à partir de `android.app.Application`

Pour localiser le fichier qui s'étend à partir de `android.app.Activity` :

1. Ouvrez le fichier Java existant qui s'étend à partir de `android.app.Activity`.
2. Ajoutez les importations suivantes.
  - a. Importation `com.tl.uic.Tealeaf`;
3. Chaque activité a besoin d'un nom de page logique qui permet d'indiquer quelle activité est actuellement affichée. Si aucun nom de page logique est indiqué, IBM Tealeaf recommande d'utiliser un nom de classe qui donne des indications sur l'activité affichée.

Ajoutez le texte suivant à la classe :

```
Chaîne privée logicalPageName;
Chaîne publique finale getLogicalPageName() {
    si ((this.logicalPageName == null) ||
        (this.logicalPageName.equals(""))) {
```

- ```

        this.logicalPageName =
this.getClass().getName().substring(this.getClass()
.getName().lastIndexOf(".") + 1);
    }
    return this.logicalPageName;
}

```
4. Dans la méthode `onPause()` :
    - a. Ajoutez `Tealeaf.onPause(this, getLogicalPageName());`
  5. Dans la méthode `onResume()` :
    - a. Ajoutez `Tealeaf.onResume(this, getLogicalPageName());`
  6. Dans la méthode `onDestroy()` :
    - a. Ajoutez `Tealeaf.onDestroy(this, getLogicalPageName());`

## Exemple

Importation `com.tl.uic.Tealeaf`;

La classe publique `BaseActivity` étend l'activité {  
Chaîne privée `logicalPageName`;

```

/**
 * Nom de page logique de l'activité.
 *
 * @return Nom de page logique de l'activité.
 */
Chaîne publique finale getLogicalPageName() {
    Si ((this.logicalPageName == null) ||
(this.logicalPageName.equals("")) {
        this.logicalPageName =
this.getClass().getName().substring(this.getClass()
.getName().lastIndexOf(".") + 1);
    }
    Renvoyez this.logicalPageName;
}

/**
 * Nom de page logique de l'activité.
 *
 * @param logicalPageName
 *         Nom de page logique de l'activité.
 */
Chaîne publique finale vide setLogicalPageName(chaîne finale logicalPageName) {
    this.logicalPageName = logicalPageName;
}

Chaîne vide protégée onPause() {
    Tealeaf.onPause(this, getLogicalPageName());
    super.onPause();
}

Chaîne vide protégée onResume() {
    Tealeaf.onResume(this, getLogicalPageName());
    super.onResume();
}

Chaîne vide protégée onDestroy() {
    Tealeaf.onDestroy(this, getLogicalPageName());
    super.onDestroy();
}
}

```

## Consignation de la disposition d'écran pour la relecture de sessions d'application mobile

IBM Tealeaf dispose de fonctions permettant de consigner les dispositions d'écran des vues d'écran des sessions d'application mobile natives. Vous pouvez relire une session d'application mobile dans une relecture sur navigateur cxImpact comme pour une session Web HTML au lieu d'afficher la session d'application mobile comme une série de captures d'écran.

Les dispositions d'écran des sessions d'application mobile native sont capturées au format JSON d'IBM Tealeaf. Les dispositions d'écran sont ensuite renvoyées au serveur de relecture. Ce dernier utilise un moteur de modèle, qui interprète les données JSON au format HTML. Vous pouvez ensuite relire la disposition d'écran à partir de la session d'application mobile native sous forme de pages HTML dans une relecture sur navigateur cxImpact.

Il existe plusieurs avantages à utiliser des données JSON pour relire une session d'application mobile au lieu de captures d'écran.

- Réduction de la bande passante. Les captures d'écran de chaque vue d'écran génèrent une quantité de données d'image relativement importante. Non seulement elle consomment une quantité importante de bande passante sans fil et cellulaire, mais elles utilisent également plus de mémoire à l'intérieur du périphérique. Elles ont également un impacts sur les performances des applications mobiles.
- Masquage des informations sensibles. Vous ne pouvez pas masquer les données sensibles dans une capture d'écran. Si vous utilisez des données JSON pour relire des sessions d'application mobile, vous pouvez masquer EditTexts en ajoutant View IDs à l'attribut MaskIdList dans le fichier `TLFConfigurableItems.properties`.
- Représentation des interactions des utilisateurs (événements UI) dans les pages HTML créées à partir des données JSON.

Pour plus d'informations sur les modèles de relecture de session d'application mobile, reportez-vous à la rubrique sur la personnalisation de la relecture de session d'application native, dans le document *IBM Tealeaf CX - Guide de configuration*.

### Modifications apportées à `TLFConfigurableItems.properties`

Pour que la relecture de session d'application native soit activée, vous devez affecter à `LogViewLayoutOnScreenTransition` la valeur `true`. Si vous ne le faites pas, le comportement de la bibliothèque n'est pas modifié.

```
#Capture de la disposition native
LogViewLayoutOnScreenTransition=true
```

Lors du prédéploiement, vous devez effectuer tous les scénarios de relecture pour collecter toutes les images avec `GetImageDataOnScreenLayout` défini sur `true`. Cela crée un contenu volumineux envoyé au serveur qui contient les images base64 utilisées pour la relecture. Lorsque l'application est prête à être déployée sur le Play Store, la valeur `false` doit être spécifiée pour `GetImageDataOnScreenLayout`.

```
#Effectué uniquement sur ImageView
GetImageDataOnScreenLayout=true
```

## Comprendre votre activité

Dans Android, une activité peut être considérée comme une page, qui est affichée sur le périphérique mobile. Par défaut, vous devez enregistrer une activité affichée.

Pour plus d'informations, voir <http://developer.android.com/training/basics/activity-lifecycle/starting.html>.

Vous pouvez enregistrer une activité affichée en plaçant les informations ci-après dans la méthode `onCreate`.

```
// Indique le nom de la page logique.
Tealeaf.logScreenview(activity, "Name", ScreenviewType.LOAD);
// Extrait la disposition de la page une fois qu'elle a été créée.
Tealeaf.logScreenLayoutOnCreate(activity, "Name");
```

Si vous devez consigner une disposition, vous pouvez utiliser ce qui suit.

```
Tealeaf.logScreenLayout(activity, "Name", delayInMS);
```

## Relecture des boîtes de dialogue d'alerte

Vous devez savoir quand une boîte de dialogue d'alerte est affichée afin qu'elle puisse être capturée correctement. `onShowListener` est l'emplacement correct à utiliser pour cela.

```
// Capture l'arrière-plan et l'alerte lorsqu'elle est affichée.
Tealeaf.logScreenLayoutSetOnShowListener(activity, dialog);
```

Si `onShowListener` existe déjà, suivez cet exemple.

```
// Ceci est placé dans onShowListener :
Tealeaf.logScreenLayout(activity, dialog);
```

Pour capturer un événement de boîte de dialogue d'alerte, suivez cet exemple.

```
public void onClick(DialogInterface dialog, int id) {
    Tealeaf.logDialogEvent(dialog, id);
}
```

## Relecture des événements de clavier

Android ne fournit pas d'événement pour déterminer quand un clavier logiciel apparaît et disparaît. Suivez l'exemple ci-après pour effectuer les ajustements nécessaires sur les contrôles basés sur `TextView`.

```
public static void addFocusAndRegister(TextView textView, Activity activity) {
    textView.setOnFocusChangeListener(new OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (hasFocus) {
                InputMethodManager imm = (InputMethodManager) v.getContext()
                    .getSystemService(Context.INPUT_METHOD_SERVICE);
                imm.showSoftInput(v, InputMethodManager.SHOW_FORCED);

                KeyboardView keyboardView = new KeyboardView(v.getContext()
                    .getApplicationContext(), null);
                Tealeaf.logEvent(keyboardView, Tealeaf.TLF_UI_KEYBOARD_
                    DID_SHOW_NOTIFICATION);
                Tealeaf.logEvent(v, Tealeaf.TLF_ON_FOCUS_CHANGE_IN);
            } else {
                Tealeaf.logEvent(v, com.tl.uic.Tealeaf.TLF_ON_FOCUS_CHANGE_OUT);
                InputMethodManager imm = (InputMethodManager) v.getContext()
                    .getSystemService(Context.INPUT_METHOD_SERVICE);
                imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
            }
        }
    });
}
```

```

        KeyboardView keyboardView = new KeyboardView(v.getContext()
        .getApplicationContext(), null);
        Tealeaf.logEvent(keyboardView , Tealeaf.TLF_UI_KEYBOARD
        _DID_HIDE_NOTIFICATION);
    }
}
});
Tealeaf.registerFormField(textView, activity);
}
EditText et = (EditText) findViewById(R.id.editText1);
addFocusAndRegister(et, this);

```

Pour plus d'informations, examinez `ControlsActivity3.java` dans l'exemple de projet de code, `UICAndroidControlsAppdarkHolo`.

## Contrôles pris en charge

IBM Tealeaf relit les contrôles étendus à partir des contrôles ci-après. Pour chaque contrôle, IBM Tealeaf remplit la valeur `t1Type` dans l'objet json qui est renvoyé au serveur.

### **ToggleButton et Switch**

Utilise le modèle switch

### **RadioGroup et RadioButton**

Utilise le modèle radioButton

### **CheckBox**

Utilise le modèle checkBox

### **Button**

Utilise le modèle button

### **Scroller, HorizontalScrollView, ScrollView**

Utilise le modèle scroll

### **AbsSeekBar**

Utilise le modèle slider

### **ProgressBar**

Utilise le modèle progressSpinner ou progressBar

### **AbsSpinner**

Utilise le modèle selectList

### **EditText**

Utilise le modèle label

### **TextView**

Utilise le modèle switch

### **ImageView**

Utilise le modèle image

### **FrameLayout, LinearLayout, ViewStub, View**

Utilise le modèle canvas

### **AbsListView**

Utilise le modèle grid

### **AlertDialog**

Utilise le modèle alert

### **TabWidget**

Utilise le modèle tabBar

## TabHost

Utilise le modèle tabContainer

# Intégration pour les applications Apache Cordova, PhoneGap et IBM Worklight à l'aide des classes Android sans les classes IBM Tealeaf

Pour cette méthode, des développeurs ajoutent des fragments de code qui aident la bibliothèque de capture IBM Tealeaf.

## Modifications de code du fichier android.app.Application

le fichier d'application gère le cycle de vie d'une application Android. IBM Tealeaf gère la bibliothèque en écoutant `onLowMemory` pour désactiver la bibliothèque si vous obtenez un avertissement, `onTerminate` pour nettoyer la bibliothèque et `onCreate` pour initialiser la bibliothèque. IBM Tealeaf vous recommande cette meilleure pratique.

## Localiser le fichier qui s'étend à partir de android.app.Application

Ce fichier n'existe probablement pas, vous devez donc le créer et l'ajouter au cycle de vie complet d'écoute d'une application Android pour contrôler la bibliothèque et consigner les informations nécessaires.

1. Créez une classe d'application à partir de `android.app.Application` et ajoutez le texte suivant. Si la classe d'application est trouvée, continuez vers les étapes suivantes.
2. Ouvrez le fichier Java existant qui s'étend à partir de `android.app.Application` class.
3. Ajoutez les importations suivantes.
  - a. `import com.tl.uic.Tealeaf;`
  - b. `import com.tl.uic.app.UICApplication;`
4. Dans `onCreate()`:
  - a. Ajoutez `Tealeaf tealeaf = new Tealeaf(this);`, qui initialise la bibliothèque Tealeaf avec une référence à l'application instrumentée.
  - b. Ajoutez `Tealeaf.enable();` qui initialise la capture d'actions utilisateur dans l'application.
5. Dans `onLowMemory()` :
  - a. Ajoutez `Tealeaf.onLowMemory();` avant `super` pour qu'il puisse ajuster la bibliothèque en raison d'une mémoire faible.
6. Dans `onTerminate():` :
  - a. Ajoutez `Tealeaf.disable();` avant `super` pour qu'il puisse désactiver la bibliothèque.
7. Ajustez `AndroidManifest.xml` pour indiquer la classe d'application, en ajoutant `android:name=".MyApplication"`.

## Exemple de classe d'application

```
import android.app.Application;
import com.tl.uic.Tealeaf;

public class MyApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        Tealeaf tealeaf = new Tealeaf(this);
    }
}
```

```

        Tealeaf.enable();
    }

    @Override
    public void onLowMemory() {
        Tealeaf.onLowMemory();
        super.onLowMemory();
    }

    @Override
    public void onTerminate() {
        Tealeaf.disable();
        super.onTerminate();
    }
}

```

### Exemple dans AndroidManifest.xml

```

<application
    android:label="@string/app_name"
    android:debuggable="true"
    android:icon="@drawable/icon"
    android:name=".TLWorklightTealeafApplication" >

```

### Modifications de code de fichier org.apache.cordova.DroidGap, com.worklight.androidgap.WLDroidGap

Le fichier s'étend à partir de la classe Activity, qui gère le cycle de vie d'une page dans une application Android native semblable à ce qu'une page fait dans une application Web.

IBM Tealeaf écoute les événements suivants onPause, qui a lieu lorsque l'application se met en arrière-plan, onResume, qui a lieu lorsque l'application est au premier plan, et onDestroy lorsque l'activité n'est plus dans la mémoire et qu'elle collecte les restes.

### Localiser le fichier qui s'étend à partir de extends from org.apache.cordova.DroidGap, com.worklight.androidgap.WLDroidGap

1. Ouvrez le fichier Java existant qui s'étend à partir de android.app.Activity.
2. Ajoutez les importations suivantes.
  - a. import com.tl.uic.Tealeaf;
  - b. import com.tl.uic.app.UICApplication;
  - c. import com.tl.uic.model.ScreenviewType;
3. Chaque activité a besoin d'un nom de page logique qui permet d'indiquer quelle activité est actuellement affichée. Si aucun nom de page logique est indiqué, IBM Tealeaf recommande d'utiliser un nom de classe qui donne des indications sur l'activité affichée.  
Ajoutez le texte suivant à la classe :  
Chaîne privée logicalPageName = "MainPage";
4. Dans la méthode onPause() :
  - a. Ajoutez Tealeaf.onPause(this, logicalPageName);
5. Dans la méthode onResume() :
  - a. Ajoutez Tealeaf.onResume(this, logicalPageName);
6. Dans la méthode onDestroy() :
  - a. Ajoutez Tealeaf.onDestroy(this, logicalPageName);
7. Dans onCreate(), ajoutez après super.onCreate()
  - a. Tealeaf.logScreenview(this, logicalPageName, ScreenviewType.LOAD);

```
b. appView.addJavascriptInterface(new
    JavaScriptInterface(this.getContext()), "tlBridge");
```

### Exemple d'IBM Worklight

```
package com.TLWorklightTealeaf;

import android.os.Bundle;

import com.tl.uic.Tealeaf;
import com.tl.uic.javascript.JavaScriptInterface;
import com.tl.uic.model.ScreenviewType;
import com.worklight.androidgap.WLDroidGap;

public class TLWorklightTealeaf extends WLDroidGap {
    private String logicalPageName = "MainPage";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Consignez la vue d'écran pour cette activité
        Tealeaf.logScreenview(this, logicalPageName, ScreenviewType.LOAD);
        //DeviceAuthManager.getInstance().setProvisioningDelegate
        (<Utilisez la classe ProvisioningDelegateImpl par défaut ou remplacez-la par votre
        implémentation IProvisioningDelegate>);

        // Ajoutez un pont pour que les données Tealeaf soient renvoyées
        appView.addJavascriptInterface(nouvelle JavaScriptInterface
        (this.getContext()), "tlBridge");

        super.loadUrl(getWebMainFilePath());
    }

    public void onPause() {
        // Gérez Tealeaf pendant l'événement onPause
        Tealeaf.onPause(this, logicalPageName);
        super.onPause();
    }

    public void onResume() {
        // Gérez Tealeaf pendant l'événement onResume
        Tealeaf.onResume(this, logicalPageName);
        super.onResume();
    }

    public void onDestroy() {
        // Gérez Tealeaf pendant l'événement onResume
        Tealeaf.onDestroy(this, logicalPageName);
        super.onDestroy();
    }
}
```

## Implémentation de screenViews

Pour les pages dans lesquelles l'état ou le contexte peut être modifié sans effectuer de rendu de la page, IBM Tealeaf segmente les données entre les états en utilisant un objet screenView.

Par exemple, si une page contient plusieurs onglets, chacun représentant une étape différente dans un processus de règlement des achats, vous pouvez instrumenter chaque onglet de la page comme screenView distinct.

Pour implémenter un screenView pour une page, complétez les étapes suivantes.

1. Si vous effectuez une extension à partir de `UICActivity`, définissez un `logicalPageName` pour indiquer l'utilisation de l'activité. Sinon, `logicalPageName` est défini sur le nom de la classe de l'activité.
2. Si l'étape précédente n'est pas terminée, appelez `Tealeaf.logScreenview` et transmettez `logicalPageName`. Vous devez également indiquer si la page en train de charger et de décharger est facultative. Par exemple :

```
Tealeaf.logScreenview(activity, logicalPageName, ScreenviewType.LOAD);
Tealeaf.logScreenview(activity, logicalPageName, ScreenviewType.UNLOAD);
```

## Configuration de base

Vous devez configurer une page cible sur votre serveur Web.

Voir «Démarrage rapide pour la configuration du serveur», à la page 25.

Définissez l'adresse de la page cible dans le fichier de configuration `TLFConfigurableItems.properties` sous la clé `PostMessageUrl`.

Voir "Configuration File" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

## Confidentialité des données

IBM Tealeaf fournit des mécanismes pour masquer ou bloquer les informations sensibles des clients, comme les numéros de carte de crédit, d'être transmises et capturées par IBM Tealeaf. Structure de journalisation mobile d'Android CX, permet de spécifier les champs qui doivent être bloqués ou masqués dans votre application Web.

Si elle est appliquée, la confidentialité des données assure que ces éléments de données ne sont jamais transmis à IBM Tealeaf.

**Remarque :** En raison de modifications dans la façon dont les données de structure client sont soumises à IBM Tealeaf afin d'être capturées, la meilleure méthode de masque ou de blocage de données sensibles est d'appliquer le filtre en capturant la structure de capture du client. Même si d'autres fonctions peuvent être déployées pour gérer les données confidentielles, il n'est pas facile de les implémenter au nouveau format de données acquises depuis l'environnement du client. IBM Tealeaf recommande d'utiliser les méthodes référencées ci-dessous.

- Voir "Configuration File" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.
- Pour plus d'informations concernant la gestion de données sensibles en général, voir "Managing Data Privacy in Tealeaf CX" dans le document *IBM Tealeaf CX Installation Manual*.

## Configuration de la mise en sessions pour les applications Android sur le client

Structure de journalisation mobile d'Android CX crée automatiquement un ID de session si aucun ne vous est fourni. Cet ID de session est utilisé pour identifier la session sur le serveur IBM Tealeaf.

IBM Tealeaf injecte des cookies pour créer une session dans le système IBM Tealeaf.

**Remarque :** Lorsqu'une application Android native ou hybride est placée en arrière-plan, la bibliothèque vide les données collectées et s'arrête au lieu de désactiver l'application. Cela se produit à moins que la session ne soit arrivée à expiration en raison de la propriété de délai d'expiration de session. Cette propriété est indiquée par `SessionTimeout` dans le fichier `TLFConfigurableItems.properties`. La valeur par défaut de cette propriété est de 30 minutes. Une fois ce délai écoulé, un nouvel identificateur de session est créé.

Il existe deux moyens de configurer la mise en sessions ; via l'ID TLTSID fourni par IBM Tealeaf ou via l'ID session client, appelé JSESSIONID. Ces deux méthodes fonctionnent comme un identificateur de session unique dans l'environnement Structure de journalisation mobile d'Android CX pour qu'IBM Tealeaf puisse effectuer le suivi des sessions client. `CookieParam` peut être défini de sorte à utiliser l'ID session client ou JSESSIONID.

Le code suivant est un exemple type de configuration dans le fichier `TLFConfigurableItems.properties` qui utilise l'ID session client.

```
#Paramètres de mise en sessions sur les cookies client
CookieUrl = http://www.sample.com
CookieDomain = .sample.com
CookiePath = /
CookieParam = JSESSIONID
CookieExpires = false
SessionTimeout=30
SessoinTimeoutKillSwitch=false
```

Dans cet exemple, le cookie arrive à expiration 30 minutes après l'heure actuelle. Lorsque la session arrive à expiration, Structure de journalisation mobile d'Android CX extrait le nouveau cookie de votre serveur d'applications et envoie le reste de la demande au serveur d'applications à l'aide de ce nouveau cookie acquis dans l'en-tête de demande. PCA regroupe tous les ID JSESSIONID utilisés dans une même session bien que l'ID JSESSIONID changeait en permanence. Si des cookies générés à partir du serveur d'applications du client sont utilisés, `SessoinTimeoutKillSwitch` peut être défini sur `true` ou `false`. Si vous spécifiez `false` pour `SessoinTimeoutKillSwitch`, l'utilisateur de la session arrivée à expiration ne revérifie pas `KillSwitchUrl` pour déterminer s'il répond.

### **Le trafic réseau utilisé dans l'application contient uniquement des requêtes**

Structure de journalisation mobile d'Android CX utilise les cookies pour ajouter des valeurs au fichier `TLFConfigurableItems.properties`.

### **Utilise l'ID de session généré par IBM Tealeaf Structure de journalisation mobile d'Android CX**

Structure de journalisation mobile d'Android CX utilise les cookies pour ajouter les valeurs suivantes dans `TLFConfigurableItems.properties`.

- `CookieUrl` est pour l'url du site publié et qui obtient des cookies pour une mise en session.
- `CookieParam` est le paramètre qui possède un id de session.
- `CookiePath` est le chemin d'accès du cookie.
- `CookieDomain` est le domaine auquel le cookie appartient.
- `CookieSecure` permet d'ajouter un cookie sécurisé qui peut uniquement être posté sur une url https qui possède un certificat valide.

- CookieExpiresFormat peut avoir le format de date ASCTIME, RFC1036 ou RFC1123, qui a une date d'expiration qui correspond à : l'heure actuelle + le délai d'attente de session indiqué dans la variable ci-dessous.
- SessionTimeout est le délai d'attente de session en minutes. Lorsque cette valeur expire, un nouvel id de session est créé.

Un exemple suit.

```
#Paramètres de mise en sessions
CookieUrl=http://m.ibm.com
CookieParam=TLTSID
CookiePath=/
CookieDomain=.ibm.com
#Que vous souhaitez créer un cookie sécurisé qui peut uniquement être envoyé à l'aide de
url https dans PostMessageUrl.
CookieSecure=false
#Formats de date valides : ASCTIME, RFC1036, RFC1123
CookieExpiresFormat=ASCTIME
#Lorsque la publication est envoyée, l'expiration des cookies correspond à l'heure actuelle
+ le délai d'attente de session
#Le délai d'attente de session est en minutes
SessionTimeout=30
```

**Remarque :** Il est important d'appeler votre serveur en premier pour obtenir un premier cookie sur lequel effectuer des mises en sessions, qui est automatiquement obtenu lorsque vous activez l'URL de commutateur d'arrêt sur l'application. Il est utilisé pour regrouper toutes les données de capture Application de capture passive CX.

## Configurez les requêtes dans une application Android

IBM Tealeaf a besoin de toutes les requêtes pour que l'identifiant de session soit placé dans les cookies de la requête. Cela active le système IBM Tealeaf pour collecter toutes les ressources en une seule session.

Si vous utilisez `org.apache.http.impl.client.DefaultHttpClient`, vous pouvez utiliser `com.tl.uic.http.TLDefaultHttpClient`, qui ajoute l'identifiant de session approprié dans le cookie de la requête. Si vous décidez de ne pas utiliser la classe IBM Tealeaf étendue, vous devez ajouter le code suivant aux classes suivantes.

### Etendre `org.apache.http.impl.client.DefaultHTTIClient`

Si vous décidez de ne pas utiliser la classe IBM Tealeaf étendue `TLDefaultHttpClient`, vous devez ajouter le code suivant aux classes suivantes.

```
Importation org.apache.http.conn.ClientConnectionManager;
Importation org.apache.http.impl.client.DefaultHttpClient;
Importation org.apache.http.params.HttpParams;

/**
 * @author ohernandez
 *
 */
La classe publique TLDefaultHttpClient étend DefaultHttpClient {
    /**
     *
     */
    TLDefaultHttpClient() public {
        super();
        this.init(null);
    }

    /**
     * @param params paramètres Http.
```

```

*/
TLDefaultHttpClient public(paramètres HttpParams finaux) {
    super(params);
    this.init(null);
}

/**
 * @param params paramètres Http.
 * @param sessionId identifiant de session Tealeaf.
 */
TLDefaultHttpClient public (paramètres HttpParams finaux,
Chaîne finale sessionId) {
    super(params);
    this.init(sessionId);
}

/**
 * @param conman ClientConnectionManager.
 * @param params paramètres Http.
 */
TLDefaultHttpClient public (final ClientConnectionManager conman,
Paramètres HttpParams finaux) {
    super(conman, params);
    this.init(null);
}

/**
 * @param sessionId identifiant de session Tealeaf.
 */
init vide privé (chaîne finale sessionId) {
    TLHttpRequestInterceptor tlHttpRequestInterceptor final =
nouveau TLHttpRequestInterceptor(sessionId);
    this.addRequestInterceptor(tlHttpRequestInterceptor);
    this.addResponseInterceptor(new TLHttpResponseInterceptor
(tlHttpRequestInterceptor));
}
}

```

## Etendre org.apache.http.HttpRequestInterceptor

Cette classe est utilisée pour injecter un identifiant de session en tant que cookie et des en-têtes supplémentaires que le système IBM Tealeaf utilise.

```

Importation java.io.IOException;
Importation java.util.Map.Entry;

Importation org.apache.http.HttpException;
Importation org.apache.http.HttpRequest;
Importation org.apache.http.HttpRequestInterceptor;
Importation org.apache.http.protocol.HttpContext;

Importation android.webkit.CookieManager;

Importation com.tl.uic.Tealeaf;
Importation com.tl.uic.util.LogInternal;

/**
 * @author ohernandez
 */
La classe publique TLHttpRequestInterceptor met HttpRequestInterceptor en oeuvre {
Chaîne privée url;
Chaîne finale privée sessionId;

/**
 * Constructeur.
 */
TLHttpRequestInterceptor() public {
    super();
}

```

```

        this.sessionId = null;
    }

    /**
     * Constructeur.
     * @param sessionId identifiant de session Tealeaf.
     */
    TLHttpRequestInterceptor public (chaîne finale sessionId) {
        this.sessionId = sessionId;
    }

    /**
     * Obtenir l'URL de la requête.
     * @return l'url de la requête.
     */
    Chaîne publique finale getUrl() {
        return url;
    }

    /**
     * Url de la requête.
     * @param url Url de la requête.
     */
    Chaîne publique finale vide setUrl(chaîne finale url) {
        this.url = url;
    }

    /**
     * {@inheritDoc}
     */
    Processus public final vide(requête HttpRequest finale, final
    Contexte HttpContext) émet HttpException, IOException {
        try {
            this.url = request.getRequestLine().getUri();

            Si (!request.containsHeader(Tealeaf.TLF_HEADER)) {
                request.addHeader(Tealeaf.TLF_HEADER, "device (Android) Lib/"
+ Tealeaf.getLibraryVersion());
            }

            Si (!request.containsHeader(Tealeaf.TLF_PROPERTY_HEADER)) {
                request.addHeader(Tealeaf.TLF_PROPERTY_HEADER,
Tealeaf.getHttpXTealeafProperty());
            }

            Si (Tealeaf.getAdditionalHeaders() != null) {
                pour l'entrée (final EntryString,<String> :
Tealeaf.getAdditionalHeaders().entrySet()) {
                    request.addHeader(entry.getKey(), entry.getValue());
                }
            }

            Cookies StringBuffer finaux = nouveau StringBuffer(Tealeaf.
getTLCookie(this.sessionId));
            Si (this.getUrl() != null) {
                Chaîne finale existingCookies = CookieManager.getInstance().
getCookie(this.getUrl());
                Si (existingCookies != null) {
                    cookies.append(';');
                    cookies.append(existingCookies);
                }
            }

            request.addHeader("Cookie", cookies.toString());
            LogInternal.log("Session cookie:" + cookies.toString());
        } catch (final Exception e) {

```

```

        Tealeaf.logException(e);
    }
}

```

## Etendre org.apache.http.HttpResponseInterceptor

Cette classe est utilisée pour obtenir des informations afin de remplir un objet de connexion IBM Tealeaf.

```

import java.io.IOException;
import java.util.Date;

import org.apache.http.HttpException;
import org.apache.http.HttpResponse;
import org.apache.http.HttpResponseInterceptor;
import org.apache.http.protocol.HttpContext;

import com.tl.uic.TLFCache;
import com.tl.uic.Tealeaf;
import com.tl.uic.util.LogInternal;

/**
 * @author ohernandez
 *
 */
La classe publique TLHttpResponseInterceptor met HttpResponseInterceptor en oeuvre {
    TLHttpRequestInterceptor tlHttpRequestInterceptor; privé final
    Date finale privée startTime;
    initTime; final privé long

/**
 * Constructeur.
 *
 * @param tlHttpRequestInterceptor TLHttpRequestInterceptor utilisé.
 */
TLHttpResponseInterceptor public (TLHttpRequestInterceptor final
tlHttpRequestInterceptor) {
    this.tlHttpRequestInterceptor = tlHttpRequestInterceptor;
    this.startTime = nouvelle date();
    this.initTime = TLFCache.timestampFromSession();
}

/**
 * {@inheritDoc}
 */
Processus public final vide(réponse HttpResponse finale,
Contexte HttpContext final) émet HttpException, IOException {
    try {
        Date finale endTime = nouvelle date();
        Date finale startLoad = nouvelle date();
        loadTime final long = (nouvelle date()).getTime() - startLoad.getTime();
        responseTime final long = endTime.getTime() - this.startTime.getTime();
        Tealeaf.logConnection(this.tlHttpRequestInterceptor.getUrl(), réponse,
this.initTime, loadTime, responseTime);
    } interceptor (exception finale e) {
        LogInternal.logException(e);
    }
}
}

```

## Utilise un ID de session non IBM Tealeaf

Obtenez votre ID de session généré et utilisez-le lorsque vous activez ou démarrez une nouvelle session dans Structure de journalisation mobile d'Android CX.

```
// Si la structure de journalisation Android est activée, utilisez
Tealeaf.enable();
Tealeaf.enable("votre id de session");
// Si la structure de journalisation Android est activée et qu'une nouvelle session doit
être créée, utilisez Tealeaf.startSession();
Tealeaf.startSession("votre id de session");
```

## Applications hybrides

Une application est considérée comme hybride si elle contient WebView dans l'application Android.

Si vous utilisez WebView, vous devez utiliser UICWebView pour consigner une activité de requête dans WebView. UICWebView étend WebView à partir de la structure Android, qui insère l'en-tête IBM Tealeaf avec l'ID de session actuel pour la mise en sessions. Si vous décidez de ne pas utiliser UICWebView, vous devez alors étendre Webview pour ajouter une mise en session.

Voir "UICWebView Class" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

### Etendre android.webkit.WebView

Le modèle de code qui suit étend la base Android WebView avec UICWebView.

```
Importation java.util.Date;
Importation java.util.Map;

Importation org.apache.http.HttpResponse;

Importation android.annotation.SuppressLint;
Importation android.content.Context;
Importation android.util.AttributeSet;
Importation android.webkit.WebView;

Importation com.tl.uic.TLFCache;
Importation com.tl.uic.Tealeaf;
Importation com.tl.uic.app.UICActivity;
Importation com.tl.uic.javascript.JavaScriptInterface;

/**
 * @author ohernandez
 */
La classe publique UICWebView étend WebView {
    Date privée endLoad;
    Date privée startLoad;
    Chaîne privée url;
    HttpResponse privée httpResponse;
    Date privée initTime;
    responseTime; privé long
    connectionInitFromSession; privé long

    /**
     * Contexte @param contexte Android.
     */
    UICWebView(final Context contexte) public {
        super(contexte);
        init();
    }

    /**
     * Contexte @param contexte Android.
     * Jeu d'attributs @param d'un AttributeSet passé au parent.
     */
    UICWebView(contexte final, jeu d'attributs AttributeSet final) public {
        super(contexte, attrs);
```

```

    init();
}

/**
 * Contexte @param contexte Android.
 * Jeu d'attributs @param d'un AttributeSet passé au parent.
 * @param ID de ressource de style defStyle par défaut.
 */
UICWebView(contexte final, jeu d'attributs AttributeSet final,
final int defStyle) {
    super(contexte, jeu d'attributs, defStyle);
    init();
}

/**
 * Lorsque la page a terminé de charger.
 * @return Lorsque la page a terminé de charger.
 */
Date publique finale getEndLoad() {
    return endLoad;
}

/**
 * Lorsque la page a terminé de charger.
 * @param endLoad Lorsque la page a terminé de charger.
 */
Chaîne publique finale vide setEndLoad(date finale endLoad) {
    this.endLoad = endLoad;
}

/**
 * Lorsque la page commence à charger.
 * @return Lorsque la page commence à charger.
 */
Date publique finale getStartLoad() {
    return startLoad;
}

/**
 * Lorsque la page commence à charger.
 * @param startLoad Lorsque la page commence à charger.
 */
Chaîne publique finale vide setStartLoad(date finale startLoad) {
    this.startLoad = startLoad;
}

/**
 * HttpResponse à partir de la connexion.
 * @return HttpResponse à partir de la connexion.
 */
HttpResponse publique finale getHttpResponse() {
    return httpResponse;
}

/**
 * HttpResponse à partir de la connexion.
 * @param httpResponse HttpResponse à partir de la connexion.
 */
Chaîne publique finale vide setHttpResponse(HttpResponse finale httpResponse) {
    this.httpResponse = httpResponse;
}

/**
 * Heure initiale à partir de la connexion.
 * @return Heure initiale à partir de la connexion.
 */
Date publique finale getInitTime() {

```

```

        return initTime;
    }

    /**
     * Heure initiale à partir de la connexion.
     * @param initTime Heure initiale à partir de la connexion.
     */
    Chaîne publique finale vide setInitTime(date finale initTime) {
        this.initTime = initTime;
        this.connectionInitFromSession = TLFCache.timestampFromSession();
    }

    /**
     * Heure de réponse à partir de la connexion.
     * @return Heure de réponse à partir de la connexion.
     */
    getResponseTime publique finale longue() {
        return responseTime;
    }

    /**
     * Heure de réponse à partir de la connexion.
     * @param responseTime Heure de réponse à partir de la connexion.
     */
    Chaîne publique finale vide setResponseTime(responseTime finale longue) {
        this.responseTime = responseTime;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    Chaîne publique vide loadData(données de chaîne finales, chaîne mimeType finale,
    Chaîne finale encoding) {
        this.url = null;
        this.initTime = null;
        this.connectionInitFromSession = 0;
        this.responseTime = 0;
        this.httpResponse = null;
        this.startLoad = new Date();
        super.loadDataWithBaseUrl(this.url, données, mimeType, codage, "");
    }

    /**
     * {@inheritDoc}
     */
    @Override
    Chaîne publique vide loadUrl(chaîne finale url) {
        loadUrl(url, null);
    }

    /**
     * {@inheritDoc}
     */
    Chaîne publique finale vide loadUrl(Chaîne url finale, mappage final
    <Chaîne, chaîne> extraHeaders) {
        this.url = url;
        Tealeaf.setTLCookie(this.url);
        super.loadUrl(url, extraHeaders);
    }

    /**
     * Initialise WebView.
     */
    Chaîne vide privée init() {
        this.setWebViewClient(nouveau UICWebViewClient());
        this.addJavascriptInterface(nouvelle JavaScriptInterface

```

```

(this.getContext()), "tlBridge");
}

/**
 * Consignez l'heure de chargement de WebView.
 */
Chaîne publique finale vide logConnection() {
    loadTime final long = this.getEndLoad().getTime() - this.getStartLoad()
.getTime();
    Tealeaf.logConnection(this.url, this.httpResponse,
this.connectionInitFromSession,
loadTime, this.responseTime);
}
}

```

## Etendre android.webkit.WebViewClient

Le modèle de code qui suit étend la base Android WebViewClient avec UICWebViewClient.

```

Importation android.webkit.WebView;
Importation android.webkit.WebViewClient;

/**
 * @author ohernandez
 *
 */
La classe publique UICWebViewClient étend WebViewClient {

    /**
     * {@inheritDoc}
     */
    @Override
    shouldOverrideUrlLoading public boolean(vue WebView finale, chaîne url finale) {
        view.loadUrl(url);
        return true;
    }
}

```

## Mise en sessions pour les applications PhoneGap

Puisqu'aucune requête ou réponse n'est renvoyée au serveur, IBM Tealeaf ne requiert pas le développement de WebView pour l'ajout d'une mise en sessions.

---

## Démarrage rapide pour la configuration du serveur

Cette section décrit les étapes de base de la configuration de serveurs IBM Tealeaf Application de capture passive CX et Windows pour capturer et traiter les données soumises à partir de Structure de journalisation mobile d'Android CX.

Pour activer le processus de données soumises, vous pouvez compléter les étapes dans les sections suivantes.

### Page cible pour la capture de trafic

IBM Tealeaf est conçu pour capturer le trafic entre un client et un serveur Web. Pour faciliter la capture, vous devez ajouter une page cible à votre environnement de serveur Web auquel Structure de journalisation mobile d'Android CX peut soumettre des publications.

Vous pouvez utiliser la même page cible disponible pour IBM Tealeaf Capture d'interface utilisateur CX pour AJAX.

Voir "UI Capture for Ajax Installation and Implementation" dans le guide *IBM Tealeaf Capture d'interface utilisateur CX pour AJAX*.

Après avoir ajouté la page cible à votre environnement Web et activé les bonnes autorisations d'accès, vous devez configurer l'URL pour la page cible à la page `TLFConfigurableItems.properties`.

**Remarque :** Si nécessaire, vous pouvez configurer la structure du client pour soumettre par HTTPS en ajoutant l'identificateur de protocole à l'URL de publication. Voir Chapitre 2, «Fichier de configuration», à la page 35.

## Gestion du volume de trafic

Vous pouvez ajouter une fonction d'échantillonnage pour travailler avec le commutateur d'arrêt Structure de journalisation mobile d'Android CX. Vous pouvez utiliser cette fonction d'échantillonnage pour réguler le taux d'échantillonnage et ainsi le volume du trafic transmis pour la capture.

Pour plus d'informations à propos des fonctions d'échantillonnage pour différents environnements de serveur, voir "Sample Code" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

## Application de capture passive CX vérification de capture de trafic

Vous devez vérifier que IBM Tealeaf Application de capture passive CX dans votre environnement IBM Tealeaf est configuré pour capturer et traiter les données soumises à partir d'infrastructures de consignment.

Les données sont soumises à l'aide de types de contenu spécifique, que Application de capture passive CX capture généralement par défaut. Vous devez vérifier que ces types de contenu ont été activés pour être capturer par la console Web Application de capture passive CX.

**Remarque :** Après avoir effectué les étapes de cette section, les données sont traitées par IBM Tealeaf.

### Vérification des configurations Application de capture passive CX de type de capture

Vous pouvez utiliser cette procédure pour vérifier que Application de capture passive CX est configuré pour capturer les types de contenu soumis par Structure de journalisation mobile d'Android CX.

**Remarque :** En fonction de la version de Application de capture passive CX installée, les types de contenu nécessaires peuvent déjà être configurés pour la capture.

Structure de journalisation mobile d'Android CX soumet des messages à l'aide du type de contenu `text/json`.

**Remarque :** Chaque structure de journalisation IBM Tealeaf peut utiliser un type de contenu différent pour soumettre des événements à capturer pour IBM Tealeaf. Assurez-vous de vérifier le type de contenu de chaque environnement de capture côté client.

1. Connectez-vous à la console Web Application de capture passive CX :  
`<PCAServer>:8080`

où

<PCAServer> est le nom d'hôte du serveur Application de capture passive CX.

2. Cliquez sur l'onglet **Pipeline**.
3. Cliquez sur **Edit Type Lists**.
4. Dans la boîte de dialogue **Capture All POST Types**, vérifiez que les valeurs suivantes sont incluses :

```
text/json
text/x-json
application/json
application/x-json
```

5. Cliquez sur **Ajouter**.
6. Application de capture passive CX est désormais configuré pour capturer les types de contenu requis. Toutes les occurrences ultérieures de ce type sont capturées.
7. Enregistrez les modifications.

Voir "PCA Web Console - Pipeline Tab" dans le manuel *IBM Tealeaf Application de capture passive CX* .

## Configuration de Application de capture passive CX pour les captures d'écran de la structure de consignation

Vous pouvez également activer Structure de journalisation mobile d'Android CX pour effectuer des captures d'écran pendant le chargement initial de chaque vue ou écran de votre application Web. Ces captures d'écran sont transmises à la page cible IBM Tealeaf au format .PNG pour une capture et utilisation pendant l'affichage de session.

Voir «Classe UICActivity», à la page 45.

Lorsque cette option est activée, vous devez configurer Application de capture passive CX pour effectuer ces captures d'écran. Par défaut, Application de capture passive CX supprime les captures d'écran de contenu binaire ou statique, vous devez donc le configurer pour capturer les images soumises en tant que POST binaires sur la page cible.

1. Connectez-vous à la console Web Application de capture passive CX :  
<PCAServer>:8080  
Où  
<PCAServer> est le nom d'hôte du serveur Application de capture passive CX.
2. Cliquez sur l'onglet **Pipeline**.
3. Cliquez sur **Edit Type Lists**.
4. Dans la liste **Excluded File Extensions**, vérifiez que png est répertorié.
5. Dans la liste **Included File Extensions**, vérifiez que png n'est pas répertorié.

**Remarque :** Si une extension de fichier est incluse dans cette liste, toutes les instances qui sont envoyées en réponse sont capturées, ce qui agrandit considérablement le volume des données capturées par le Application de capture passive CX. La capture de cette manière n'est pas requise.

6. Dans la case des types de POST binaires, saisissez la valeur suivante :  
image/png
7. Cliquez sur **Ajouter**.
8. Le type de POST image/png est ajouté et activé afin d'être capturé. Ce paramètre permet aux publications PNG d'être capturées par Application de capture passive CX.

9. Enregistrez les modifications.

Voir "PCA Web Console - Pipeline Tab" dans le document *IBM Tealeaf Application de capture passive CX*.

### **Activation de l'extraction automatisée de POST compressés**

Structure de journalisation mobile d'Android CX compresse automatiquement les données POST. Vous devez configurer Application de capture passive CX pour les extraire.

**Remarque :** Pour Application de capture passive CX build 3502 ou ultérieur, cette décompression se fait automatiquement. Si vous utilisez une de ces versions Application de capture passive CX, cette étape de la configuration n'est pas nécessaire.

1. Dans la console Web Application de capture passive CX, cliquez sur l'onglet **Pipeline**.
2. Sélectionnez **Inflate compressed requests and responses**.
3. Enregistrez les modifications.

Les POST compressés sont désormais automatiquement extraits par Application de capture passive CX et traités normalement.

## **Options de contrôle des captures et du traitement**

Vous pouvez utiliser différents outils pour tester vos captures de configuration et de contrôle sur une base continue.

### **A la page cible**

Vous pouvez tester la fonctionnalité de base de la page cible en déclenchant les actions GET et POST sur l'URL où la page cible a été installée.

Voir "UI Capture for Ajax Installation and Implementation" dans le guide *IBM Tealeaf Capture d'interface utilisateur CX pour AJAX*.

### **Dans le pipeline Windows**

Vous pouvez contrôler la capture et le traitement des occurrences dans le pipeline Windows en temps réel en passant par le système de gestion IBM Tealeaf. Voir "TMS Pipeline Status Tab" dans le document *IBM Tealeaf cxImpact Administration Manual*.

## **Configuration de la mise en sessions pour les applications Android dans IBM Tealeaf**

IBM Tealeaf fournit plusieurs mécanismes d'identification et de suivi des sessions de visiteurs individuels. Pour Structure de journalisation mobile d'Android CX, une configuration supplémentaire peut être requise.

Passez en revue les étapes et complétez les étapes requises pour mettre votre application mobile en session.

Pour activer la mise en sessions d'occurrences capturées par Structure de journalisation mobile d'Android CX, vous devez déployer l'agent de mise en sessions. Voir "Sessioning Session Agent" dans le document *IBM Tealeaf CX Configuration*

1. Connectez-vous au portail en tant qu'administrateur.
2. Dans le menu **Portal**, sélectionnez **Tealeaf > TMS**. Le système de gestion IBM Tealeaf s'ouvre.  
Voir "Tealeaf Management System" dans le document *IBM Tealeaf cxImpact Administration Manual*.
3. Cliquez sur l'onglet **WorldView**.
4. Pour **View**, sélectionnez **Servers**.  
Voir "TMS WorldView Tab" dans le document *IBM Tealeaf cxImpact Administration Manual*.
5. Cliquez sur le noeud **Transport Service**.
6. Sélectionnez **Transport Service configuration**. Puis cliquez sur **View/Edit**.
7. Pipeline Editor s'ouvre.

**Remarque :** Vérifiez que l'agent de mise en session a été installé.

- S'il ne l'a pas été, faites le glisser de la fenêtre Available SessionAgents vers le pipeline.
  - Pour plus d'informations à propos de son déploiement, voir "TMS Pipeline Editor" dans le document *IBM Tealeaf cxImpact Administration Manual*.
8. Sélectionnez **Sessioning session agent**. Cliquez sur **Edit**.
  9. Dans la configuration de l'agent de mise en sessions, modifiez la valeur **PrimarySessField** de la manière suivante :  
`PrimarySessField=AppEnv:SessionID,env:HTTP_X_TEALEAF_SESSION`
  10. Enregistrez le fichier de configuration.
  11. Effectuez la modification sur tous les serveurs. Un redémarrage est requis pour que les nouvelles clés de mise en sessions fassent effet.

## Nom de la vue Android comme URL pendant la relecture

Les erreurs qui peuvent se produire pendant la relecture de session peuvent être résolues à l'aide du nom de vue Android comme URL pendant la relecture.

Lorsque vous essayez de relire une session capturée par Structure de journalisation mobile d'Android CX, un message d'erreur peut indiquer qu'aucune page n'est visualisable dans la session. Cette erreur est liée à la manière dont les pages sont mappées sur les terminaux Android.

Au lieu d'afficher l'URL pendant la relecture, vous pouvez configurer des clients de relecture IBM Tealeaf pour afficher le nom de vue Android qui élimine ce message d'erreur esthétique.

Pour utiliser le nom de vue Android comme URL lorsque vous reliez la session, complétez les étapes suivantes pour configurer la règle du profil de relecture appropriée.

**Remarque :** La configuration de cette règle doit actuellement être appliquée en passant par le profil de relecture stocké sur le serveur de relecture. Les instructions sont listées ci-dessous.

Lorsque cette modification est appliquée, le nom de vue Android comme URL se trouve dans la liste **Navigable Pages** dans la vue de relecture Browser Based Replay et Visualiseur CX RealiTea, au lieu de l'URL de page cible the generic IBM Tealeaf habituelle.

- Browser Based Replay ou BBR

Ce client Web est accessible grâce au portail IBM Tealeaf et récupère ses sessions via le serveur de relecture dans votre environnement IBM Tealeaf.

- Voir "CX Browser Based Replay" dans le manuel d'utilisation *IBM Tealeaf cxImpact*.
- Voir "Configuring the Replay Server" dans le manuel de configuration *IBM Tealeaf CX*.

- IBM Tealeaf Visualiseur CX RealiTea, ou RTV

Ce client de relecture est une application Windows autonome qui doit être installée séparément sur votre bureau. Via Visualiseur CX RealiTea, vous pouvez rechercher et relire des sessions stockées dans votre environnement IBM Tealeaf.

Voir le manuel d'utilisateur *IBM Tealeaf Visualiseur CX RealiTea*.

Pour modifier le profil de relecture, vous pouvez utiliser ces options.

- «Appliquer la modification du nom de la vue en local»  
Vous pouvez appliquer les modifications en local via IBM Tealeaf Visualiseur CX RealiTea, une application de bureau pour l'affichage et la relecture d'événements. Vous pouvez utiliser cette option pour tester les modifications avant de reconfigurer le serveur de relecture.
- «Appliquer la modification du nom de la vue au serveur de relecture», à la page 31  
Si vous n'avez pas accès à Visualiseur CX RealiTea, vous pouvez modifier les paramètres du serveur.

**Remarque :** Les utilisateurs Visualiseur CX RealiTea doivent synchroniser leurs profils de relecture local sur le profil de serveur pour appliquer les modifications.

### Appliquer la modification du nom de la vue en local :

Complétez les étapes suivantes dans Visualiseur CX RealiTea pour effectuer des modifications en local et tester l'utilisation du nom de la vue Android pendant la relecture.

1. Démarrez l'application Visualiseur CX RealiTea sur votre bureau local.

**Remarque :** Visualiseur CX RealiTea doit être installé en local sur votre bureau Windows. Voir "RealiTea Viewer (RTV) User Manual" dans le guide d'utilisation *IBM Tealeaf Visualiseur CX RealiTea*.

2. Chargez une session capturée à partir de Structure de journalisation mobile d'Android CX.
3. Dans le menu Visualiseur CX RealiTea, sélectionnez **Tools > Options...**
4. Cliquez sur l'onglet **Profiles**.  
Voir "RealiTea Viewer - Profile Options" dans le guide d'utilisation *IBM Tealeaf Visualiseur CX RealiTea*.
5. Si vous ne l'avez pas encore fait, saisissez le nom du serveur de relecture qui contrôle le profil de relecture dans la case du serveur. Cliquez sur **Rechercher les mises à jour maintenant**.  
Si une version de serveur du profil est disponible, votre version locale se synchronise avec elle.

6. Cliquez sur **Edit Raw Profile**.

7. Complétez les étapes suivantes répertoriées dans la section qui suit. Vous modifiez le fichier XML brut stocké sur votre bureau local. Voir «Appliquer la modification du nom de la vue au serveur de relecture», à la page 31.

8. Après avoir modifié le profil, cliquez sur **Save Changes & Exit**.
9. Cliquez sur **OK**.
10. Permet de relire la session. Cliquez sur **Replay** dans la barre d'outils.
11. Dans la liste **Navigable Pages**, les URL répertoriées reflètent le nom de vue Android pour l'écran.

| Page | URL                                                                                                       | Gen Time | Net Time | Round Trip | Bytes | Status | TimeSta |
|------|-----------------------------------------------------------------------------------------------------------|----------|----------|------------|-------|--------|---------|
| 3    | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.005    | 0.009    | 0.015      | 3573  | 200    | 11:34:0 |
| 4    | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.005    | 0.009    | 0.015      | 3573  | 200    | 11:34:0 |
| 5    | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.005    | 0.009    | 0.015      | 3573  | 200    | 11:34:0 |
| 7    | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.004    | 0.009    | 0.013      | 1783  | 200    | 11:34:4 |
| 8    | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.004    | 0.009    | 0.013      | 1783  | 200    | 11:34:4 |
| 9    | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.004    | 0.009    | 0.013      | 1783  | 200    | 11:34:4 |
| 11   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.004    | 0.008    | 0.012      | 1926  | 200    | 11:35:0 |
| 12   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.004    | 0.008    | 0.012      | 1926  | 200    | 11:35:0 |
| 13   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.004    | 0.008    | 0.012      | 1926  | 200    | 11:35:0 |
| 15   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.004    | 0.009    | 0.013      | 2687  | 200    | 11:35:0 |
| 16   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.004    | 0.009    | 0.013      | 2687  | 200    | 11:35:0 |
| 17   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.004    | 0.009    | 0.013      | 2687  | 200    | 11:35:0 |
| 20   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.004    | 0.008    | 0.013      | 2679  | 200    | 11:35:2 |
| 21   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.004    | 0.008    | 0.013      | 2679  | 200    | 11:35:2 |
| 22   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.004    | 0.008    | 0.013      | 2679  | 200    | 11:35:2 |
| 23   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoOrdersViewController     | 0.004    | 0.008    | 0.013      | 2679  | 200    | 11:35:2 |
| 25   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.005    | 0.010    | 0.015      | 2977  | 200    | 11:35:2 |
| 26   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.005    | 0.010    | 0.015      | 2977  | 200    | 11:35:2 |
| 27   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.005    | 0.010    | 0.015      | 2977  | 200    | 11:35:2 |
| 28   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoOrdersViewController     | 0.005    | 0.010    | 0.015      | 2977  | 200    | 11:35:2 |
| 29   | http://www.straussandplessier.com/UITabBarController/INavigationController/UITMoreListController          | 0.005    | 0.010    | 0.015      | 2977  | 200    | 11:35:2 |
| 31   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.005    | 0.008    | 0.013      | 2461  | 200    | 11:35:5 |
| 32   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.005    | 0.008    | 0.013      | 2461  | 200    | 11:35:5 |
| 33   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoCategoriesViewController | 0.005    | 0.008    | 0.013      | 2461  | 200    | 11:35:5 |
| 34   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoOrdersViewController     | 0.005    | 0.008    | 0.013      | 2461  | 200    | 11:35:5 |
| 35   | http://www.straussandplessier.com/UITabBarController/INavigationController/UITMoreListController          | 0.005    | 0.008    | 0.013      | 2461  | 200    | 11:35:5 |
| 38   | http://www.straussandplessier.com/TProtoSplashViewController                                              | 0.029    | 0.038    | 0.067      | 26609 | 200    | 11:36:0 |
| 39   | http://www.straussandplessier.com/UITabBarController/INavigationController/TProtoLogInViewController      | 0.029    | 0.038    | 0.067      | 26609 | 200    | 11:36:0 |

12. Si la liste **Navigable Pages** est correctement remplie, vous pouvez renvoyer vos modifications au serveur de relecture pour qu'elles soient déployées aux autres utilisateurs du serveur.
  - a. Dans le menu **RTV**, sélectionnez **Tools > Options**.
  - b. Cliquez sur l'onglet **Profils**.
  - c. Cliquez sur **Téléchargez les paramètres sur le serveur**.

### Appliquer la modification du nom de la vue au serveur de relecture :

Complétez les étapes suivantes sur le serveur de relecture pour que tous les utilisateurs du serveur de relecture sur navigateur voient le nom de vue Android pendant la relecture.

Dans la procédure qui suit, vous pouvez appliquer les modifications au serveur de relecture en accédant au serveur et en modifiant le fichier approprié. Cette modification est ensuite disponible à tous les utilisateurs du profil du serveur de relecture, ce qui inclut tous les utilisateurs sur navigateur.

1. Connectez-vous au serveur hébergeant le serveur de relecture en tant qu'administrateur.
2. Modifiez le fichier suivant :  
`<Tealeaf_install_directory>\system\ReplayServerProfile.xml`
3. Localisez la section **RequestMapping**, qui doit se trouver en haut du fichier.
4. Ajoutez une entrée d'élément d'URL pour le mappage de requêtes. Localisez l'en-tête suivant :  
`<RequestEntry name="URL">`
5. Ajoutez le nom de clé suivant comme nouvelle entrée à la liste des entrées :  
`<Nom de clé ="HTTP_X_TEALEAF_VIEW_CLASS" enabled="1"/>`
6. Enregistrez le fichier.

## Configuration d'exécution

Si nécessaire, vous pouvez modifier les paramètres du serveur depuis l'application client. Tous les éléments de configuration peuvent être configurés de manière dynamique depuis le client.

Vous pouvez planifier la gestion de la configuration du serveur pendant l'initialisation de l'application, puis effectuer une mise à jour sélective et si nécessaire pendant l'exécution.

Voir "TeaLeaf Class" dans le guide *IBM Tealeaf Structure de journalisation mobile d'Android CX*.

---

## IBM Tealeaf événements pour Structure de journalisation mobile d'Android CX

Le format JSON est utilisé pour suivre les données capturées par le Structure de journalisation mobile d'Android CX.

### Type de données Description

#### Données d'environnement de capture côté client (JSON)

Si vous utilisez une création d'événements à étapes, les données de l'environnement de capture côté client sont soumises au format JSON et disponibles grâce à Browser Based Replay pour vérification et création d'événements. Voir "Step-Based Eventing" dans le *IBM Tealeaf Event Manager Manual*.

Pour plus d'explications concernant la capture de ces données en événements IBM Tealeaf, voir "Integrating Client Framework Data into Tealeaf" dans le document *IBM Tealeaf Client Framework Data Integration Guide*.

#### Données d'environnement de capture côté client (scission d'occurrences)

Voir «Versions de structure client prises en charge dans cette documentation», à la page 1.

---

## Mise à niveau de Structure de journalisation mobile d'Android CX

Lors de la mise à niveau de IBM Tealeaf Structure de journalisation mobile d'Android CX, vous complétez les tâches générales suivantes.

**Remarque :** Certaines étapes peuvent varier en fonction de vos environnements de développement et d'application.

1. Passez en revue les exigences actuelles. Voir «Exigences minimales», à la page 2.
2. Passez en revue les contenus du package. Voir «Contenus du package», à la page 1.
3. Au lieu d'intégrer Structure de journalisation mobile d'Android CX à votre application en développement, vous pouvez utiliser l'échantillon d'application fourni par IBM Tealeaf. Voir «Utilisation du modèle d'application», à la page 2.
4. Vérifiez que votre environnement d'application est configuré afin de respecter les exigences du projet.
5. Vérifiez que les modifications de code d'exigences ont été appliquées. Voir «Modifications de projet Android», à la page 4.

6. Appliquez la configuration de base.

**Remarque :** La dernière version de Structure de journalisation mobile d'Android CX inclut de nouvelles configurations requises. Voir «Configuration de base», à la page 16.

7. Vérifiez que les types de contenu approprié sont capturés et transférés par IBM Tealeaf Application de capture passive CX. Voir «Application de capture passive CX vérification de capture de trafic», à la page 26.

**Remarque :** Cette étape met sous tension le commutateur pour démarrer la capture et le traitement des données de l'application mobile dans Tealeaf. En fonction du volume de données, vous pouvez utiliser le commutateur d'arrêt. Voir «Gestion du volume de trafic», à la page 26.

8. Testez votre solution mise à niveau.



---

## Chapitre 2. Fichier de configuration

Ce fichier de configuration pour la bibliothèque Structure de journalisation mobile d'Android CX est placé dans le dossier de documents d'une application Android. Il s'appelle `TLFConfigurableItems.properties`, qui est un fichier de propriétés Java.

---

### Paramètres de niveau de consignation

Les paramètres de niveau de consignation configurent les paramètres de consignation de base.

Tableau 1. Paramètres de niveau de consignation

| ID article                          | Description                                                                                                                                                                                                                                                                                               | Valeurs     |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| LoggingLevel                        | Le niveau actuel de consignation s'applique uniquement lorsque le niveau de consignation n'est pas indiqué dans les instructions de consignation. 0 a la plus haute priorité.<br><br>Pour désactiver la consignation, lancez <code>Tealeaf.Disable()</code> . Voir Chapitre 5, «Référence», à la page 45. | Entier, 0-5 |
| DisplayLogging                      | Lorsque la valeur <code>true</code> est définie, les instructions de consignation de débogage sont affichées dans LogCat. Filtre de la balise <code>UICAndroid</code> .                                                                                                                                   | Booléen     |
| LogViewLayout<br>OnScreenTransition | Si la valeur est <code>true</code> , <code>UICAndroid</code> consigne la disposition de l'écran. Si la valeur est <code>false</code> , la disposition de l'écran n'est pas consignée.                                                                                                                     | Booléen     |

---

### Paramètres de commutateur d'arrêt

Ces paramètres contrôlent le commutateur d'arrêt et l'utilisation (ou non) d'une liste blanche de numéros de téléphone dont il est possible de capturer les événements.

Tableau 2. Paramètres de commutateur d'arrêt

| ID article        | Description                                                                                                                                                                                                                                      | Valeurs                 |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| KillSwitchEnabled | Si la valeur est <code>true</code> , la structure vérifie la page cible du commutateur d'arrêt avant de démarrer. Vous devez spécifier les propriétés suivantes.<br><br>Si <code>KillSwitchEnabled=false</code> , la structure démarre toujours. | <code>true/false</code> |
| KillSwitchUrl     | Définit l'URL à vérifier pour le commutateur d'arrêt. La structure requiert une réponse positive afin d'effectuer une initialisation lorsque <code>KillSwitchEnabled</code> est défini sur <code>true</code> .                                   | Adresse URL             |

Tableau 2. Paramètres de commutateur d'arrêt (suite)

| ID article                 | Description                                                                                                                                                                                                                                                                                              | Valeurs                                          |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| KillSwitchMaxNumberOfTries | Nombre de fois où la structure vérifie l'URL du commutateur d'arrêt avant d'abandonner. Cette valeur doit être définie sur 1.                                                                                                                                                                            | Entier                                           |
| KillSwitchTimeInterval     | Temps d'attente avant de revérifier l'URL du commutateur d'arrêt si elle ne répond pas                                                                                                                                                                                                                   | Secondes                                         |
| UseWhiteList               | Si la valeur est true et KillSwitchEnabled, la structure requiert un identifiant de téléphone à attribuer avant d'appelerEnable pour vérifier la liste blanche d'identifiants de téléphone.<br><br>Si la valeur est false etKillSwitchEnabled, par défaut la structure utilise un échantillon aléatoire. | true/false                                       |
| WhiteListParam             | Paramètre utilisé pour envoyer l'identifiant de liste blanche correspondant à l'identifiant du téléphone.                                                                                                                                                                                                | Le serveur actuel de la liste blanche utilise id |

## Paramètres du fichier de cache local

Vous utiliserez ces paramètres pour configurer l'utilisation du cache local du périphérique.

Tableau 3. Paramètres du fichier de cache local

| ID article             | Description                                                                                                                                                                            | Valeurs     |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| HasToPersistLocalCache | Si la valeur true est définie, les données sont stockées dans un stockage local sur le périphérique, au lieu de la mémoire. Les paramètres suivants doivent également être configurés. | true/false  |
| CachingLevel           | Niveau de cache actuel. Ne s'applique que si HasToPersistLocalCache est true. 0 a la plus haute priorité.                                                                              | Entier, 0-5 |
| CachedFileMaxBytesSize | Nombre maximal d'octets à stocker sur le périphérique.                                                                                                                                 | Octets      |

## Paramètres de publication

Ces paramètres contrôlent le niveau de consignation, l'URL, le volume, et la fréquence des publications sur la page cible.

Tableau 4. Paramètres de publication

| ID article               | Description                                                                                                                                                                                                                                                                                                                                                                                                           | Valeurs                 |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| PostMessageUrl           | URL de publication de données sur votre serveur.<br><b>Remarque :</b> Pour activer le transport sécurisé entre la structure de consignation et la page cible, configurez cette URL afin qu'elle commence par <code>https://</code> . Pour plus d'informations à propos de la page cible, voir Chapitre 1, «Installation et mise en oeuvre», à la page 1.                                                              | Adresse URL             |
| PostMessageLevelWiFi     | Niveau de consignation d'événements à envoyer au serveur par Wi-Fi lorsque la performance réseau est bonne. 0 a la plus haute priorité.                                                                                                                                                                                                                                                                               | 0-5                     |
| PostMessageLevelCellular | Niveau de consignation d'événements à envoyer au serveur par le réseau mobile (3G). 0 a la plus haute priorité.                                                                                                                                                                                                                                                                                                       | 0-5                     |
| MaxStringsLength         | Longueur de chaîne maximale à envoyer à la page cible par valeur dans les instructions de consignation. Empêche les chaînes longues de prendre trop d'espace de stockage et de bande passante.<br><b>Remarque :</b> Cette valeur doit être définie sur 1.                                                                                                                                                             | Entier                  |
| ManualPostEnabled        | Si la valeur <code>true</code> est définie, la structure envoie des données au serveur uniquement lorsque votre application appelle <code>requestManualServerPost</code> .<br><br>Si la valeur <code>false</code> est définie, vous devez configurer les paramètres suivants.<br><b>Remarque :</b> Vous ne pouvez pas activer ce paramètre et <code>DoPostOnIntervals</code> ensemble.                                | <code>true/false</code> |
| DoPostOnIntervals        | Si la valeur <code>true</code> est définie, la structure envoie des données au serveur à intervalles réguliers spécifiés par <code>PostMessageTimeIntervals</code> lorsque l'application est en premier plan. Cette valeur doit être définie sur <code>true</code> si <code>ManualPostEnabled=false</code> .<br><b>Remarque :</b> Vous ne pouvez pas activer ce paramètre et <code>ManualPostEnabled</code> ensemble. | <code>true/false</code> |
| PostMessageTimeIntervals | A quelle fréquence la structure envoie-t-elle des données au serveur lorsque <code>DoPostOnIntervals</code> est défini sur <code>true</code> ?<br><b>Remarque :</b> Cette valeur doit être définie pour être supérieure à <code>+PostMessageTimeout</code> plus <code>PostMessageDelayTimeToSendData</code> .                                                                                                         | Secondes                |

Tableau 4. Paramètres de publication (suite)

| ID article         | Description                                                                                                                                                                             | Valeurs  |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| PostMessageTimeout | Délai d'attente pour les publications de la structure sur le serveur. Si la structure ne reçoit pas de réponse du serveur dans ce délai, elle continue d'essayer d'envoyer des données. | Secondes |

## Paramètres de masque

Ces paramètres contrôlent le masque de confidentialité.

Tableau 5. Paramètres de masque

| ID article                   | Description                                                                                                                    | Valeurs |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------|---------|
| HasMasking                   | Les valeurs peuvent être true ou false pour masquer les valeurs de contrôle. Si HasMasking=true, complétez la valeur suivante. | Booléen |
| MaskIdList                   | Identifiants ou expressions régulières délimités par des virgules pour trouver les identifiants.                               | Chaîne  |
| HasCustomMask                | Cela peut être true ou false d'utiliser les prochaines valeurs ci-dessous si true est la valeur définie.                       | Booléen |
| SensitiveSmallCaseAlphabet   | Caractère à utiliser pour les minuscules.                                                                                      | Chaîne  |
| SensitiveCapitalCaseAlphabet | Caractère à utiliser pour les majuscules.                                                                                      | Chaîne  |
| SensitiveSymbol              | Caractère à utiliser pour les symboles.                                                                                        | Chaîne  |
| SensitiveNumber              | Caractère à utiliser pour les chiffres.                                                                                        | Chaîne  |

## Paramètre de type de message de filtre

Ce paramètre détermine les types de message qui sont renvoyés au serveur.

Tableau 6. Paramètre de type de message de filtre

| ID article         | Description                                                                                                                                                                                             | Valeurs     |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| FilterMessageTypes | S'il a TRUE comme valeur, seuls les MessageTypes inclus dans la liste séparée par des virgules sont renvoyés au serveur. S'il a FALSE comme valeur, tous les types de message sont renvoyés au serveur. | TRUE/ FALSE |

---

## Paramètres de cookie

Ces paramètres contrôlent les cookies.

Tableau 7. Paramètres de cookie

| ID article          | Description                                                                                                                                      | Valeurs                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| CookieSecure        | S'il a pour valeur TRUE, un paramètre de sécurité est ajouté au cookie. Ce paramètre ne peut être utilisé que dans les url de publication https. | TRUE/ FALSE                                            |
| CookieExpiresFormat | Ce paramètre est utilisé pour indiquer le format d'expiration du cookie.                                                                         | Formats de date valides : ASCTIME, RFC1036, ou RFC1123 |

---

## Paramétrage de la temporisation de session

Ce paramètre contrôle les temporisations de session.

Tableau 8. Paramétrage de la temporisation de session

| ID article     | Description                                                                                                                  | Valeurs |
|----------------|------------------------------------------------------------------------------------------------------------------------------|---------|
| SessionTimeout | Lorsque SessionTimeout est défini, l'expiration des cookies est l'heure actuelle plus la valeur de temporisation de session. | Minutes |

---

## Paramètres de capture d'écran

Ces paramètres contrôlent les captures d'écran.

**Remarque :** Vous pouvez stocker les captures d'écran sur la mémoire au lieu d'une mémoire locale sur le périphérique. Pour permettre aux captures d'écran d'être enregistrées sur la mémoire, vous devez définir HasToPersistLocalCache sur false dans les paramètres de fichier de cache local.

Tableau 9. Paramètres de capture d'écran

| ID article               | Description                                                                                                                                                               | Valeurs      |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| ScreenshotFormat         | Format de la capture d'écran.                                                                                                                                             | PNG/ JPG     |
| PercentOfScreenshotsSize | Pourcentage des dimensions pixel originales de la capture d'écran auquel les captures d'écran publiées sont soumises, 1-100.                                              | Entier 1-100 |
| PercentToCompressImage   | Pourcentage de compression des images. Ce paramètre peut uniquement être utilisé pour les images JPG. Les images PNG ignorent ce paramètre et définissent 100 par défaut. | Entier 1-100 |

---

## Ne pas modifier les paramètres internes

Ne modifiez pas ces paramètres sauf si IBM Tealeaf vous y invite.

Tableau 10. Ne pas modifier les paramètres internes

| ID article                   | Description                                                                                                                                                                                                                                                                        | Valeurs     |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| PostMessageSocketTimeout     | Délai de dépassement du socket pour les publications de structure sur le serveur. Si la structure ne reçoit pas de réponse du serveur dans ce délai, elle continue d'essayer d'envoyer des données.                                                                                | Secondes    |
| CompressPostMessage          | S'ils ont true comme valeur, les POST HTTP soumis à partir de la structure sont compressés.<br><b>Remarque :</b> Pour extraire les POST compressés, une configuration supplémentaire du serveur peut être requise. Voir Chapitre 1, «Installation et mise en oeuvre», à la page 1. | true/ false |
| BufferLimit                  | Nombre de messages à stocker dans la mémoire à envoyer au serveur.                                                                                                                                                                                                                 | Entier      |
| BufferPercent                | Pourcentage à supprimer de Bufferlimit avant qu'il ne soit enregistré sur le cache, s'il est activé.                                                                                                                                                                               | Pourcentage |
| TimeIntervalBetweenSnapshots | Intervalle de temps pour prendre des instantanés de données environnementales                                                                                                                                                                                                      | Secondes    |

---

## Chapitre 3. Echantillon d'applications

L'échantillon de code fourni avec la distribution de logiciel contient une application Android qui peut être utilisée pour tester le système IBM Tealeaf.

Une version de code est fournie.

### Version

### Description

#### **UICAndroidControlsAppDarkHolo**

Application Android, qui disposent des contrôles actuellement pris en charge que vous pouvez relire dans BBR avec des exemples d'utilisation des programmes d'écoute d'événements avec l'API Tealeaf.



---

## Chapitre 4. Instructions

Appliquez les conseils suivants au développement et à l'intégration de votre application de IBM Tealeaf Structure de journalisation mobile d'Android CX.

- Utilisez le commutateur d'arrêt pour contrôler la journalisation de l'application Android. Voir Chapitre 6, «Code d'échantillon», à la page 67.
- Ajoutez des identifiants à tous les contrôles d'interface utilisateur que vous souhaitez capturer.
- Appliquez le masque ou le blocage de confidentialité de toutes les données client sensibles en passant par Structure de journalisation mobile d'Android CX.

**Remarque :** L'édition 8.5, IBM Tealeaf a introduit les événements à étapes qui simplifient et unifient la capture d'événements à partir de tous les environnements de capture côté client, tout en améliorant les performances. En raison de modifications dans la façon dont les données sont regroupées, vous pouvez désormais appliquer la confidentialité des données via les environnements client individuels, au lieu d'utiliser les méthodes du serveur IBM Tealeaf pour la confidentialité des données.

- Voir Chapitre 1, «Installation et mise en oeuvre», à la page 1.
- Pour plus d'informations sur la confidentialité d'une manière générale, voir le chapitre "Managing Data Privacy in Tealeaf CX" dans le document *IBM Tealeaf CX Installation Manual*.
- Suivez les instructions pour l'extension de fichier d'application. Voir Chapitre 1, «Installation et mise en oeuvre», à la page 1.
- Suivez les instructions pour l'extension de fichier d'activité. Voir Chapitre 1, «Installation et mise en oeuvre», à la page 1.
- Débogage local dans Eclipse à l'aide de LogCat.  
Pour afficher les messages de débogage dans l'onglet LogCat d'Eclipse, saisissez la chaîne suivante : `tag:UICAndroid`.
- Suivez les instructions d'utilisation de `TLDDefaultHttpClient` concernant les requêtes et les réponses pour obtenir des indicateurs de connexion. Voir «Classe `TLDDefaultHttpClient`», à la page 58.
- Suivez les instructions concernant l'utilisation de zones de texte pour obtenir plus d'indicateurs. Voir «Classe Tealeaf», à la page 49.
- En raison de la manière dont les messages JSON sont capturés et transmis, forcez une soumission de tous les messages en file d'attente avant d'autoriser les utilisateurs de votre application mobile native d'ouvrir une vue Web. Si cette étape n'est pas effectuée, les occurrences peuvent paraître dans le désordre pendant la relecture dans IBM Tealeaf.  
Voir "Search and Replay for Mobile App" dans le document *IBM Tealeaf CX Mobile User Manual*.



---

## Chapitre 5. Référence

Cette section contient des informations de référence de classe à propos de la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.

---

### Classe UICActivity

La classe `com.tl.uic.app.UICActivity` étend `com.tl.uic.app.Activity`. `UICActivity` permet de capturer les actions utilisateur d'une application Android et d'activer la capture d'écran après la création de l'Activity.

**Remarque :** Pour activer la capture d'écran dans IBM Tealeaf , vous devez configurer le PCA pour qu'il capture les POST binaires d'images png. Voir Chapitre 1, «Installation et mise en oeuvre», à la page 1.

#### Détail sur une méthode

##### **getTakeSnapshotAfterCreate**

```
public Boolean getTakeSnapshotAfterCreate()
```

Effectuer une capture d'écran ou non après création.

Indique si la capture d'écran doit être faite après création.

##### **setTakeSnapshotAfterCreate**

```
public void setTakeSnapshotAfterCreate(final Boolean takeSnapshotAfterCreate)
```

Effectuer une capture d'écran ou non après création.

`takeSnapshotAfterCreate` - Effectuer une capture d'écran ou non après création.

##### **getTookImage**

```
public Boolean getTookImage()
```

Si la capture d'écran a été faite ou non.

##### **setTookImage**

```
public void setTookImage(final Boolean tookImage)
```

Si la capture d'écran a été faite ou non.

`tookImage` - Si la capture d'écran a été faite ou non.

##### **getLogicalPageName**

```
public String getLogicalPageName()
```

Nom de page logique de l'Activity.

Renvoie le nom de page logique de l'Activity. Si aucun n'a été attribué, il reçoit un nom de la classe d'activité et un trait de soulignement (  ) avec l'heure actuelle en millisecondes est ajouté.

##### **setLogicalPageName**

```
public void setLogicalPageName(final String logicalPageName)
```

Nom de page logique de l'Activity.

logicalPageName - Nom de page logique de l'Activity.

### **getImageBackground**

```
public int getImageBackground()
```

Couleur d'arrière-plan de l'image de la capture d'écran.

Renvoie la couleur d'arrière-plan de l'image de la capture d'écran.

### **setImageBackground**

```
public void setImageBackground(final int imageBackground)
```

Couleur d'arrière-plan de l'image de la capture d'écran.

imageBackground - Couleur d'arrière-plan de l'image de la capture d'écran.

### **getView**

```
public View getView()
```

Vue à utiliser pour la capture d'écran.

Renvoie la vue à utiliser pour la capture d'écran.

### **setView**

```
public void setView(final View view)
```

Vue à utiliser pour la capture d'écran.

view - Vue à utiliser pour la capture d'écran.

### **getNumOnGlobalLayoutListener**

```
public int getNumOnGlobalLayoutListener()
```

Nombre de OnGlobalLayoutListener définis sur des vues.

Renvoie le délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

### **setNumOnGlobalLayoutListener**

```
public void setNumOnGlobalLayoutListener(final  
int numOnGlobalLayoutListener)
```

Nombre de OnGlobalLayoutListener définis sur des vues.

millisecondSnapshotDelay - Délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

### **getMillisecondSnapshotDelay**

```
public long getMillisecondSnapshotDelay()
```

Délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

Renvoie le délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

### **setMillisecondSnapshotDelay**

```
public void setMillisecondSnapshotDelay(final  
long millisecondSnapshotDelay)
```

Délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

millisecondSnapshotDelay - Délai en millisecondes avant que la capture d'écran soit prise, car plus de temps est nécessaire pour obtenir un bon rendu.

## Référence

Tableau 11. Classe UICActivity

| Package           | Classe                                           | Description                                                                                                                           |
|-------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app    | UICActivity                                      | UICActivity utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                       |
|                   | «Classe UICApplication», à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                      |
| com.tl.uic        | «Classe Tealeaf», à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.              |
| com.tl.uic.http   | «Classe TLDefaultHttpClient», à la page 58       | Etend DefaultHttpClient pour surveiller les requêtes d'URL                                                                            |
|                   | «Classe TLHttpRequestInterceptor», à la page 60  | Etend HttpRequestInterceptor pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                   | «Classe TLHttpResponseInterceptor», à la page 61 | Etend HttpResponseInterceptor pour obtenir des détails pour l'objet de connexion Tealeaf.                                             |
| com.tl.uic.webkit | «Classe UICWebView», à la page 62                | WebView utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                   | «Classe UICWebChromeClient», à la page 64        | Etend WebChromeClient pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |
|                   | «Classe UICWebViewClient», à la page 65          | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf                   |

## Classe UICApplication

La classe `com.tl.uic.app.UICApplication` étend `android.app.Application`. `UICApplication` permet de capturer les actions utilisateur d'une application Android.

### Détail sur une méthode

#### `getTealeaf`

```
public Tealeaf getTealeaf()
```

Obtenez l'instance en cours de IBM Tealeaf.

Renvoie l'instance en cours de IBM Tealeaf.

### Référence

Tableau 12. Classe `UICApplication`

| Package                        | Classe                                                         | Description                                                                                                                                        |
|--------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>com.tl.uic.app</code>    | «Classe <code>UICActivity</code> », à la page 45               | <code>UICActivity</code> utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                       |
|                                | <code>UICApplication</code>                                    | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                                   |
| <code>com.tl.uic</code>        | «Classe <code>Tealeaf</code> », à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.                           |
| <code>com.tl.uic.http</code>   | «Classe <code>TLDefaultHttpClient</code> », à la page 58       | Etend <code>DefaultHttpClient</code> pour surveiller les requêtes d'URL                                                                            |
|                                | «Classe <code>TLHttpRequestInterceptor</code> », à la page 60  | Etend <code>HttpRequestInterceptor</code> pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                                | «Classe <code>TLHttpResponseInterceptor</code> », à la page 61 | Etend <code>HttpResponseInterceptor</code> pour obtenir des détails pour l'objet de connexion IBM Tealeaf                                          |
| <code>com.tl.uic.webkit</code> | «Classe <code>UICWebView</code> », à la page 62                | <code>WebView</code> utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                                | «Classe <code>UICWebChromeClient</code> », à la page 64        | Etend <code>WebChromeClient</code> pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |

Tableau 12. Classe UICApplication (suite)

| Package | Classe                                  | Description                                                                                                         |
|---------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|         | «Classe UICWebViewClient», à la page 65 | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf |

## Classe Tealeaf

La classe `com.tl.uic.TeaLeaf` étend `java.lang.Object`. La bibliothèque TeaLeaf permet de capturer les actions utilisateur d'une application Android.

### Zones

Tableau 13. Classe Tealeaf

| Zone                                    | Récapitulatif      | Description                                                                       |
|-----------------------------------------|--------------------|-----------------------------------------------------------------------------------|
| static<br><code>java.lang.String</code> | TLF_SESSION_HEADER | Clé d'en-tête utilisée pour effectuer des mises en sessions de X-Tealeaf-Session. |
| static<br><code>java.lang.String</code> | BALISE             | UICAndroid utilisé dans LogCat.                                                   |

### Les champs utilisés dans les gestionnaires d'événements pour afficher correctement les actions utilisateur

Tableau 14. Classe Tealeaf

| Zone                                    | Récapitulatif           | Description                                                                              |
|-----------------------------------------|-------------------------|------------------------------------------------------------------------------------------|
| static<br><code>java.lang.String</code> | TLF_ON_FOCUS_CHANGE_IN  | Utilisé dans les contrôles TextView pour indiquer la priorité in.                        |
| static<br><code>java.lang.String</code> | TLF_ON_FOCUS_CHANGE_OUT | Utilisé dans les contrôles TextView pour indiquer la priorité .                          |
| static<br><code>java.lang.String</code> | TLF_ON_GROUP_COLLAPSE   | Utilisé dans les contrôles ExpandableListView pour indiquer quel groupe a été réduit.    |
| static<br><code>java.lang.String</code> | TLF_ON_GROUP_EXPAND     | Utilisé dans les contrôles ExpandableListView pour indiquer quel groupe a été développé. |
| static<br><code>java.lang.String</code> | TLF_ON_DRAWER_OPENED    | Utilisé dans les contrôles SlidingDrawer pour indiquer quel tiroir est ouvert.           |
| static<br><code>java.lang.String</code> | TLF_ON_DRAWER_CLOSED    | Utilisé dans les contrôles SlidingDrawer pour indiquer quel tiroir est fermé.            |

## Champs utilisés pour accéder aux valeurs de fichier de configuration

Tableau 15. Classe Tealeaf

| Zone                       | Récapitulatif                       | Description                                                            |
|----------------------------|-------------------------------------|------------------------------------------------------------------------|
| static<br>java.lang.String | TLF_LOGGING_LEVEL                   | Niveau de journalisation par défaut.                                   |
| static<br>java.lang.String | DISPLAY_LOGGING                     | Afficher ou non les instructions de débogage dans LogCat               |
| static<br>java.lang.String | TLF_KILL_SWITCH_ENABLED             | Activation ou désactivation du commutateur d'arrêt.                    |
| static<br>java.lang.String | TLF_KILL_SWITCH_URL                 | Url pour le commutateur d'arrêt.                                       |
| static<br>java.lang.String | TLF_KILL_SWITCH_MAX_NUMBER_OF_TRIES | Nombre maximum d'essais                                                |
| static<br>java.lang.String | TLF_KILL_SWITCH_TIME_INTERVAL       | Intervalle de temps avant de réessayer d'accès au commutateur d'arrêt. |
| static<br>java.lang.String | TLF_USE_WHITE_LIST                  | Utilisation ou non de la liste blanche.                                |
| static<br>java.lang.String | TLF_WHITE_LIST_PARAM                | Paramètres utilisés par cette liste blanche.                           |
| static<br>java.lang.String | TLF_USE_RANDOM_SAMPLE               | Utilisation ou non de l'échantillon aléatoire.                         |
| static<br>java.lang.String | TLF_RANDOM_SAMPLE_PARAM             | Paramètres utilisés par l'échantillon aléatoire.                       |
| static<br>java.lang.String | TLF_HAS_TO_PERSIST_LOCAL_CACHE      | Possibilité ou non d'enregistrer le cache sur le terminal.             |
| static<br>java.lang.String | TLF_CACHED_LEVEL                    | Niveau de cache à enregistrer sur le terminal.                         |
| static<br>java.lang.String | TLF_CACHED_FILE_MAX_BYTES_SIZE      | Taille maximum du cache en octets à enregistrer sur le terminal.       |
| static<br>java.lang.String | TLF_POST_MESSAGE_URL                | URL de la page cible.                                                  |

Tableau 15. Classe Tealeaf (suite)

| Zone                       | Récapitulatif                   | Description                                                                                                                                                |
|----------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| static<br>java.lang.String | TLF_POST_MESSAGE_LEVEL_WIFI     | Niveau de journalisation si le niveau de connexion est le Wi-Fi.                                                                                           |
| static<br>java.lang.String | TLF_POST_MESSAGE_LEVEL_CELLULAR | Niveau de journalisation si le niveau de connexion est cellulaire.                                                                                         |
| static<br>java.lang.String | TLF_MAX_STRINGS_LENGTH          | Longueur de chaîne maximum.                                                                                                                                |
| static<br>java.lang.String | TLF_MANUAL_POST_ENABLED         | Activation ou désactivation du contrôle d'articles sur la page cible. Le développeur est responsable des articles sur la page cible.                       |
| static<br>java.lang.String | TLF_DO_POSTS_ON_INTERVALS       | Configurer ou non les articles de structures à intervalles définis.                                                                                        |
| static<br>java.lang.String | TLF_POST_MESSAGE_TIME_INTERVALS | Intervalle de temps entre les articles.                                                                                                                    |
| static<br>java.lang.String | TLF_POST_MESSAGE_MAX_BYTES_SIZE | Taille maximum en octets pour la publication d'un message.                                                                                                 |
| static<br>java.lang.String | TLF_HAS_MASKING                 | Masquer ou non les valeurs de contrôles.                                                                                                                   |
| static<br>java.lang.String | TLF_MASK_ID_LIST                | Chaîne délimitée par des virgules qui peut avoir des ID de contrôles ou des expressions régulières pour trouver les ID de contrôles.                       |
| static<br>java.lang.String | TLF_HAS_CUSTOM_MASK             | Utilisation ou non de valeurs de masque personnalisées à remplacer. Si elle est définie sur false, la structure de journalisation renvoie une chaîne vide. |

Tableau 15. Classe Tealeaf (suite)

| Zone                       | Récapitulatif                       | Description                                                     |
|----------------------------|-------------------------------------|-----------------------------------------------------------------|
| static<br>java.lang.String | TLF_SENSITIVE_SMALL_CASE_ALPHABET   | Lettre minuscule à remplacer pendant le masque personnalisé.    |
| static<br>java.lang.String | TLF_SENSITIVE_CAPITAL_CASE_ALPHABET | Lettre en majuscule à remplacer pendant le masque personnalisé. |
| static<br>java.lang.String | TLF_SENSITIVE_SYMBOL                | Symbole à remplacer pendant le masque personnalisé.             |
| static<br>java.lang.String | TLF_SENSITIVE_NUMBER                | Nombre à remplacer pendant le masque personnalisé.              |

## Constructeur

### Public Tealeaf (Application app)

TeaLeaf est une bibliothèque qui permet de capturer des actions utilisateur d'une application Android.

Paramètres :

- application - Référence à l'application Android actuelle.

## Détails des méthodes

### getCurrentSessionId

```
public static java.lang.String getCurrentSessionId()
```

Obtenez l'ID de session actuel.

### getPhoneId

```
public static java.lang.String getPhoneId()
```

Obtenez l'ID utilisé avec la liste blanche sur le serveur du commutateur d'arrêt.

### setPhoneId

```
public static void setPhoneId (java.lang.String phoneId)
```

Définissez l'ID de téléphone utilisé avec la liste blanche sur le serveur de commutateur d'arrêt.

### isEnabled

```
public static boolean isEnabled()
```

Activation d'une bibliothèque.

Indique si la bibliothèque TeaLeaf a été activée.

### getApplication()

```
public static android.app.Application getApplication()
```

Référence à l'application Android actuelle.

Renvoie les références à l'application Android actuelle.

#### **getMessageVersion()**

```
public static String getMessageVersion()
```

Obtenez la version de message JSON actuelle.

Renvoie la version de message JSON actuelle.

#### **getLibraryVersion()**

```
public static java.lang.String getLibraryVersion()
```

Référence à la version de bibliothèque actuelle.

Renvoie les références à la version de bibliothèque actuelle.

#### **activation**

```
public static java.lang.Boolean enable()  
public static java.lang.Boolean enable(sessionId)
```

Activation de la bibliothèque avec un ID de session donné ou généré.

- `sessionId` - ID de session donné à utiliser.

Indique si la bibliothèque TeaLeaf a été activée.

#### **désactivation**

```
public static java.lang.Boolean disable()
```

Désactivation d'une bibliothèque.

Renvoie si la bibliothèque IBM Tealeaf est désactivée.

#### **onPause**

```
public static Boolean onPause(final Activity activity,  
    final String logicalPageName)
```

Si vous n'utilisez pas `UICActivity`, ajoutez cet appel à votre méthode de fichier d'activités `onPause` avant d'appeler `super`.

- `activité` - Activité qui appelle `onPause`.
- `logicalPageName` - Nom descriptif de l'activité qui appelle `onPause`.

Renvoie `True/False` s'il a été capable de correctement mettre en pause.

#### **onResume**

```
public static Boolean onResume(final Activity activity,  
    final String logicalPageName)
```

Si vous n'utilisez pas `UICActivity`, ajoutez cet appel à votre méthode de fichier d'activités `onResume` avant d'appeler `super`.

- `activité` - Activité qui appelle `onResume`.
- `logicalPageName` - Nom descriptif de l'activité qui appelle `onResume`.

Renvoie `True/False` s'il a été capable de reprendre correctement.

#### **onDestroy**

```
public static Boolean onDestroy(final Activity activity,  
    final String logicalPageName)
```

Si vous n'utilisez pas `UICActivity`, ajoutez cet appel à votre méthode de fichier d'activités `onDestroy` avant d'appeler `super`.

- `activité` - Activité qui appelle `onResume`.
- `logicalPageName` - Nom descriptif de l'activité qui appelle `onResume`.

Renvoie True/False s'il a été capable de détruire correctement.

### **OnLowMemory**

```
public static java.lang.Boolean OnLowMemory()
```

Si vous n'utilisez pas UICApplication, ajoutez cet appel à votre méthode de fichier d'activités OnLowMemory avant d'appeler super.

#### **Renvoie :**

True/False s'il a été capable de nettoyer correctement.

### **terminer**

```
public static java.lang.Boolean terminate()
```

Si vous n'utilisez pas UICApplication, ajoutez cet appel à votre fichier d'application de méthode terminate avant d'appeler super.

Renvoie True/False s'il a été capable de terminer correctement.

### **vidage**

```
public static java.lang.Boolean flush()
```

A utiliser pour vider les données.

#### **Renvoie :**

True/False s'il a été capable de vider les données sur le serveur.

### **logEvent**

```
public static java.lang.Boolean  
logEvent(final View view)public static java.lang.  
Boolean logEvent(final View view,  
    final java.lang.String eventType)public  
static java.lang.Boolean logEvent(final View view,  
    final java.lang.String eventType, final  
int logLevel)
```

Consigner un événement à partir d'un gestionnaire d'événements.

- vue - Contrôle à partir du gestionnaire d'événements.
- eventType - Type d'événement de gestionnaire d'événements.
- logLevel - Niveau de journalisation de la bibliothèque TeaLeaf.

Renvoie True/False s'il a été capable de consigner un événement.

### **logCustomEvent**

```
public static java.lang.Boolean  
logCustomEvent(final java.lang.String eventName)  
public static java.lang.Boolean  
logCustomEvent(final java.lang.String eventName,  
final int logLevel)public  
static java.lang.Boolean  
logCustomEvent(final java.lang.String eventName,  
final  
java.util.HashMap<java.lang.String, final java.lang.  
String> data)public static java.lang.  
Boolean  
logCustomEvent(final java.lang.String eventName,  
final  
java.util.HashMap<java.lang.String, java.lang.  
String> data,  
final int logLevel)
```

Consignez un événement personnalisé.

- eventName - Nom d'événement à consigner.

- données - Clé et paire de valeurs à consigner.
- `logLevel` - Niveau de journalisation de la bibliothèque TeaLeaf.

Renvoie True/False s'il a été capable de consigner un événement.

### **logException**

```
public static java.lang.Boolean
    logException(final java.lang.Throwable exception)public
static java.lang.Boolean
    logException(final java.lang.Throwable exception,
        final java.lang.String extraMessage)
```

Consignez une exception.

- `exception` - Exception à consigner.
- `extraMessage` - Message supplémentaire qui a été placé avant d'obtenir les détails d'exception.

Renvoie True/False si une exception a été consignée.

### **logScreenview**

```
public static Boolean logScreenview(final Activity activity,
    final String logicalPageName, final ApplicationContextType
    applicationContextType)
public static Boolean logScreenview(final Activity activity,
    final String logicalPageName,
    final ApplicationContextType applicationContextType,
    final String referrer)
```

Consignez un contexte d'application (screenView).

- `activité` - Activité avec une modification de contexte d'application (screenView).
- `logicalPageName` - Nom d'activité ou nom descriptif créé sur le terminal.
- `applicationContextType` - ApplicationContextType du contexte d'application.
- `référéncieur` - Page de référencieur qu'une page logique utilise.

Renvoie True/False si une exception a été consignée.

### **logScreenLayout**

```
public static Boolean logScreenLayout(final Activity activity)
```

Consignez la disposition de l'activité immédiatement sans nom de disposition.

```
public static Boolean logScreenLayout(final Activity activity,final
String name)
```

Consignez la disposition de l'activité immédiatement avec un nom de disposition.

```
public static Boolean logScreenLayout(final Activity activity,final
String name, final int delayMS)
```

Consignez la disposition d'une activité avec un délai d'exécution.

- `activity` - Activité à consigner.
- `name` - Nom de la disposition.
- `delayMS` - Nombre de millisecondes duquel l'appel doit être retardé.

Indique si la disposition a été consignée.

```
public static Boolean logScreenLayout(final Activity activity,
final AlertDialog alertDialog, final String title, final String message)
```

```
public static Boolean logScreenLayout(final Activity activity,
final AlertDialog alertDialog, final String name, final String title,
final String message)
```

Consignez la disposition de la boîte de dialogue d'alerte.

- activity - Activité dans laquelle AlertDialog doit être consigné.
- alertDialog - AlertDialog à consigner.
- name - Nom de la vue d'écran dans laquelle l'alerte apparaît.
- title - Titre affiché dans la boîte de dialogue d'alerte.
- message - Message affiché dans la boîte de dialogue d'alerte.

Indique si la disposition a pu être consignée.

### **logScreenLayoutSetOnShowListener**

```
public static Boolean logScreenLayoutSetOnShowListener
(final Activity activity, final AlertDialog alertDialog,
final String title, final String message)
```

```
public static Boolean logScreenLayoutSetOnShowListener
(final Activity activity, final AlertDialog alertDialog,
final String name, final String title, final String message)
```

Consignez la disposition de la boîte de dialogue d'alerte.

- activity - Activité dans laquelle AlertDialog doit être consigné.
- alertDialog - AlertDialog à consigner.
- name - Nom de la vue d'écran dans laquelle l'alerte apparaît.
- title - Titre affiché dans la boîte de dialogue d'alerte.
- message - Message affiché dans la boîte de dialogue d'alerte.

Indique si la disposition a été consignée.

### **logScreenLayoutSetOnCreate**

```
public static Boolean logScreenLayoutOnCreate(final Activity activity,
final String name)
```

Consignez la disposition de l'activité avec OnGlobalLayoutListener pour savoir quand la vue est terminée.

- activity - Activité à consigner.
- name - Nom de la disposition.

Indique si la disposition a été consignée.

### **logConnection**

```
public static java.lang.Boolean
logConnection(final java.lang.String url,
final org.apache.http.HttpResponse
httpResponse, final java.util.Date initTime,
final long loadTime, final
long responseTime)
```

Consignez une connexion.

- url - Url de la connexion.
- httpResponse - HttpResponse de la connexion.
- initTime - Heure initiale de la réponse.
- loadTime - Heure de chargement de la réponse.

- `responseTime` - Temps de réponse.

Renvoie `True/False` si une connection a été consignée.

#### **takeScreenShot**

```
public static java.lang.Boolean
takeScreenShot(final View view, final java.lang.
String imageFileName)
```

Effectuez une capture d'écran de la vue donnée.

Remarque : cette méthode nécessite de pouvoir enregistrer sur le terminal pour effectuer une capture d'écran.

- `vue` - Vue pour effectuer une capture d'écran.
- `imageFileName` - Nom de l'image.

Renvoie `True/False` si une capture d'écran a été faite.

#### **startSession**

```
public static void startSession()
public static void startSession(final sessionId)
```

Indiquez si vous souhaitez un démarrage avec un ID de session donné ou généré.

- `sessionId` - ID de session à utiliser.

#### **requestManualServerPost**

```
public static java.lang.Boolean requestManualServerPost()
```

Publiez les données actuelles consignées.

Renvoie `True/False` si des données ont été publiées.

#### **getApplicationContextOffset**

```
public static long getApplicationContextOffset()
```

Décalage actuel du contexte de l'application.

Renvoie `long` : décalage actuel du contexte de l'application.

#### **registerFormField**

```
public static Boolean
registerFormField(final View formField, final Activity activity)
public static Boolean
registerFormField(final View formField, final Activity activity,
final int logLevel)
```

Enregistrez le champ de formulaire qui permet d'obtenir des statistiques.

- `formField` - Champ de formulaire à enregistrer.
- `activité` - Activité disposant d'un champ de formulaire.
- `logLevel` - Niveau de journalisation pour la bibliothèque IBM Tealeaf.

Renvoie `True/False` si un champ de formulaire a été enregistré.

#### **isApplicationInBackground**

```
public static Boolean isApplicationInBackground()
```

Renvoie si l'application a été placée en arrière-plan si elle n'avait aucune activité affichée au premier plan.

Renvoie `True/False` si l'application a été placée en arrière-plan.

## Référence

Tableau 16. Classe Tealeaf

| Package           | Classe                                           | Description                                                                                                              |
|-------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app    | «Classe UICActivity», à la page 45               | UICActivity utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.          |
|                   | «Classe UICApplication», à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.         |
| com.tl.uic        | TeaLeaf                                          | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur. |
| com.tl.uic.http   | «Classe TLDefaultHttpClient»                     | Etend DefaultHttpClient pour surveiller les requêtes d'URL                                                               |
|                   | «Classe TLHttpRequestInterceptor», à la page 60  | Etend HttpRequestInterceptor pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                              |
|                   | «Classe TLHttpResponseInterceptor», à la page 61 | Etend HttpResponseInterceptor pour obtenir des détails pour l'objet de connexion IBM Tealeaf                             |
| com.tl.uic.webkit | «Classe UICWebView», à la page 62                | WebView utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                |
|                   | «Classe UICWebChromeClient», à la page 64        | Etend WebChromeClient pour surveiller le navigateur, il est ensuite possible d'effectuer des captures d'écran            |
|                   | «Classe UICWebViewClient», à la page 65          | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf      |

---

## Classe TLDefaultHttpClient

La classe `com.tl.uic.http.TLDefaultHttpClient` étend `org.apache.http.impl.client.DefaultHttpClient`. Vous pouvez utiliser `TLDefaultHttpClient` pour comprendre comment une valeur d'URL a été récupérée pour ajouter l'objet de connexion IBM Tealeaf.

**Remarque :** Des instances des classes `TLHttpRequestInterceptor` et `TLHttpRequestInterceptor` sont ajoutées à `TLDefaultHttpClient` pour obtenir des détails pour l'objet de connexion IBM Tealeaf.

## Référence

Tableau 17. Classe `TLDefaultHttpClient`

| Package           | Classe                                                         | Description                                                                                                                                        |
|-------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app    | «Classe <code>UICActivity</code> », à la page 45               | <code>UICActivity</code> utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                       |
|                   | «Classe <code>UICApplication</code> », à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                                   |
| com.tl.uic        | «Classe <code>Tealeaf</code> », à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.                           |
| com.tl.uic.http   | <code>TLDefaultHttpClient</code>                               | Etend <code>DefaultHttpClient</code> pour surveiller les requêtes d'URL                                                                            |
|                   | «Classe <code>TLHttpRequestInterceptor</code> », à la page 60  | Etend <code>HttpRequestInterceptor</code> pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                   | «Classe <code>TLHttpResponseInterceptor</code> », à la page 61 | Etend <code>HttpResponseInterceptor</code> pour obtenir des détails pour l'objet de connexion IBM Tealeaf                                          |
| com.tl.uic.webkit | «Classe <code>UICWebView</code> », à la page 62                | <code>WebView</code> utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                   | «Classe <code>UICWebChromeClient</code> », à la page 64        | Etend <code>WebChromeClient</code> pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |
|                   | «Classe <code>UICWebViewClient</code> », à la page 65          | Etend <code>WebViewClient</code> pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf                   |

## Classe TLHttpRequestInterceptor

La classe `com.tl.uic.http.TLHttpRequestInterceptor` étend `org.apache.http.HttpRequestInterceptor`. Vous pouvez utiliser `TLHttpRequestInterceptor` pour ajouter des en-têtes Tealeaf pour la mise en sessions.

### Méthodes redéfinies

- `processus`

### Référence

Tableau 18. Classe `TLHttpRequestInterceptor`

| Package                        | Classe                                                         | Description                                                                                                                                        |
|--------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>com.tl.uic.app</code>    | «Classe <code>UICActivity</code> », à la page 45               | <code>UICActivity</code> utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                       |
|                                | «Classe <code>UICApplication</code> », à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                                   |
| <code>com.tl.uic</code>        | «Classe <code>Tealeaf</code> », à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.                           |
| <code>com.tl.uic.http</code>   | «Classe <code>TLDefaultHttpClient</code> », à la page 58       | Etend <code>DefaultHttpClient</code> pour surveiller les requêtes d'URL.                                                                           |
|                                | <code>TLHttpRequestInterceptor</code>                          | Etend <code>HttpRequestInterceptor</code> pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                                | «Classe <code>TLHttpResponseInterceptor</code> », à la page 61 | Etend <code>HttpResponseInterceptor</code> pour obtenir des détails pour l'objet de connexion Tealeaf.                                             |
| <code>com.tl.uic.webkit</code> | «Classe <code>UICWebView</code> », à la page 62                | <code>WebView</code> utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                                | «Classe <code>UICWebChromeClient</code> », à la page 64        | Etend <code>WebChromeClient</code> pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |

Tableau 18. Classe *TLHttpRequestInterceptor* (suite)

| Package | Classe                                          | Description                                                                                                                |
|---------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|         | «Classe <i>UICWebViewClient</i> », à la page 65 | Etend <i>WebViewClient</i> pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf |

## Classe *TLHttpResponseInterceptor*

La classe `com.tl.uic.http.TLHttpResponseInterceptor` étend `org.apache.http.HttpResponseInterceptor`. Vous pouvez utiliser `TLHttpResponseInterceptor` pour obtenir des détails pour l'objet de connexion IBM Tealeaf.

### Méthodes redéfinies

- `Processus`

### Référence

Tableau 19. Classe *TLHttpResponseInterceptor*

| Package                        | Classe                                                  | Description                                                                                                              |
|--------------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>com.tl.uic.app</code>    | «Classe <i>UICActivity</i> », à la page 45              | <i>UICActivity</i> utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.   |
|                                | «Classe <i>UICApplication</i> », à la page 48           | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.         |
| <code>com.tl.uic</code>        | «Classe <i>Tealeaf</i> », à la page 49                  | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur. |
| <code>com.tl.uic.http</code>   | «Classe <i>TLDefaultHttpClient</i> », à la page 58      | Etend <i>DefaultHttpClient</i> pour surveiller les requêtes d'URL                                                        |
|                                | «Classe <i>TLHttpRequestInterceptor</i> », à la page 60 | Etend <i>HttpRequestInterceptor</i> pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                       |
|                                | <i>TLHttpResponseInterceptor</i>                        | Etend <i>HttpResponseInterceptor</i> pour obtenir des détails pour l'objet de connexion IBM Tealeaf                      |
| <code>com.tl.uic.webkit</code> | «Classe <i>UICWebView</i> », à la page 62               | <i>WebView</i> utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                         |

Tableau 19. Classe TLHttpResponseInterceptor (suite)

| Package | Classe                                    | Description                                                                                                                           |
|---------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
|         | «Classe UICWebChromeClient», à la page 64 | Etend WebChromeClient pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |
|         | «Classe UICWebViewClient», à la page 65   | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf                   |

## Classe UICWebView

La classe `com.tl.uic.webkit.UICWebView` étend `android.webkit.WebView`. Vous pouvez utiliser `UICWebView` pour ajouter un ID de session aux requêtes d'en-tête à des fins de mise en sessions.

Cette classe ajoute un objet de connexion pour fournir des informations concernant `WebView`.

### Détail sur une méthode

#### **getEndLoad**

```
public Date getEndLoad()
```

Lorsque la page a terminé de charger.

Renvoie Date : lorsque la page a terminé de charger.

#### **setEndLoad**

```
public void setEndLoad(final Date endLoad)
```

Lorsque la page a terminé de charger.

endLoad - Lorsque la page a terminé de charger.

#### **getStartLoad**

```
public Date getStartLoad()
```

Lorsque la page commence à charger.

Renvoie Date : lorsque la page commence à charger.

#### **setStartLoad**

```
public void setStartLoad(final Date startLoad)
```

Lorsque la page commence à charger.

startLoad - lorsque la page commence à charger.

#### **getHttpResponse**

```
public HttpResponse getHttpResponse()
```

HttpResponse à partir de la connexion.

Renvoie HttpResponse : HttpResponse à partir de la connexion.

### **setHttpResponse**

```
public void setHttpResponse  
(final HttpResponse httpResponse)
```

HttpResponse à partir de la connexion.

httpResponse - HttpResponse à partir de la connexion.

### **getInitTime**

```
public Date getInitTime()
```

Heure initiale à partir de la connexion.

Renvoie Date : heure initiale à partir de la connexion.

### **setInitTime**

```
public void setInitTime  
(final Date initTime)
```

Heure initiale à partir de la connexion.

InitTime - heure initiale à partir de la connexion.

### **getResponseTime**

```
public long getResponseTime()
```

Heure de réponse à partir de la connexion.

long : heure de réponse à partir de la connexion.

### **setResponseTime**

```
public void setResponseTime  
(final long responseTime)
```

Heure de réponse à partir de la connexion.

ResponseTime - heure de réponse à partir de la connexion.

### **logConnection**

```
public void logConnection()
```

Consigne l'heure actuelle de connexion de webview.

## **Référence**

Tableau 20. Classe UICWebView

| Package        | Classe                                | Description                                                                                                              |
|----------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app | «Classe UICActivity», à la page 45    | UICActivity utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.          |
|                | «Classe UICApplication», à la page 48 | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.         |
| com.tl.uic     | «Classe Tealeaf», à la page 49        | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur. |

Tableau 20. Classe UICWebView (suite)

| Package           | Classe                                           | Description                                                                                                                                             |
|-------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.http   | «Classe TLDefaultHttpClient», à la page 58       | Etend DefaultHttpClient pour surveiller les requêtes d'URL                                                                                              |
|                   | «Classe TLHttpRequestInterceptor», à la page 60  | Etend HttpRequestInterceptor pour ajouter des en-têtes Structure de journalisation mobile d'Android CX pour la mise en sessions                         |
|                   | «Classe TLHttpResponseInterceptor», à la page 61 | Etend HttpResponseInterceptor pour obtenir des détails pour l'objet de connexion Structure de journalisation mobile d'Android CX                        |
| com.tl.uic.webkit | UICWebView                                       | WebView utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                                               |
|                   | «Classe UICWebChromeClient»                      | Etend WebChromeClient pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran                   |
|                   | «Classe UICWebViewClient», à la page 65          | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions Structure de journalisation mobile d'Android CX |

## Classe UICWebChromeClient

La classe `com.tl.uic.webkit.UICWebChromeClient` étend `android.webkit.WebChromeClient`. Vous pouvez utiliser `UICWebChromeClient` pour contrôler la progression du navigateur afin d'effectuer une capture d'écran du périphérique.

### Méthodes redéfinies

- `onProgressChanged`

### Référence

Tableau 21. Classe UICWebChromeClient

| Package        | Classe                             | Description                                                                                                     |
|----------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app | «Classe UICActivity», à la page 45 | UICActivity utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX. |

Tableau 21. Classe UICWebChromeClient (suite)

| Package           | Classe                                           | Description                                                                                                                           |
|-------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
|                   | «Classe UICApplication», à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                      |
| com.tl.uic        | «Classe Tealeaf», à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.              |
| com.tl.uic.http   | «Classe TLDefaultHttpClient», à la page 58       | Etend DefaultHttpClient pour surveiller les requêtes d'URL                                                                            |
|                   | «Classe TLHttpRequestInterceptor», à la page 60  | Etend HttpRequestInterceptor pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                   | «Classe TLHttpResponseInterceptor», à la page 61 | Etend HttpResponseInterceptor pour obtenir des détails pour l'objet de connexion IBM Tealeaf                                          |
| com.tl.uic.webkit | «Classe UICWebView», à la page 62                | WebView utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                   | UICWebChromeClient                               | Etend WebChromeClient pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |
|                   | «Classe UICWebViewClient»                        | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf                   |

## Classe UICWebViewClient

La classe `com.tl.uic.webkit.UICWebViewClient` étend `android.webkit.WebViewClient`. Vous pouvez utiliser `UICWebViewClient` pour contrôler le chargement d'une URL afin d'ajouter des en-têtes IBM Tealeaf pour la mise en sessions.

### Méthodes redéfinies

- `shouldOverrideUrlLoading`

## Référence

Tableau 22. Classe UICWebViewClient

| Package           | Classe                                           | Description                                                                                                                           |
|-------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| com.tl.uic.app    | «Classe UICActivity», à la page 45               | UICActivity utilisé pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                       |
|                   | «Classe UICApplication», à la page 48            | Application utilisée pour contrôler la bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX.                      |
| com.tl.uic        | «Classe Tealeaf», à la page 49                   | Bibliothèque IBM Tealeaf Structure de journalisation mobile d'Android CX utilisée pour capturer les actions utilisateur.              |
| com.tl.uic.http   | «Classe TLDefaultHttpClient», à la page 58       | Etend DefaultHttpClient pour surveiller les requêtes d'URL                                                                            |
|                   | «Classe TLHttpRequestInterceptor», à la page 60  | Etend HttpRequestInterceptor pour ajouter des en-têtes IBM Tealeaf pour la mise en sessions                                           |
|                   | «Classe TLHttpResponseInterceptor», à la page 61 | Etend HttpResponseInterceptor pour obtenir des détails pour l'objet de connexion IBM Tealeaf                                          |
| com.tl.uic.webkit | «Classe UICWebView», à la page 62                | WebView utilisé pour ajouter un ID de session ID aux requêtes d'en-têtes.                                                             |
|                   | «Classe UICWebChromeClient», à la page 64        | Etend WebChromeClient pour surveiller si le navigateur a terminé son action, il est ensuite possible d'effectuer des captures d'écran |
|                   | UICWebViewClient                                 | Etend WebViewClient pour surveiller le chargement des URL pour ajouter des en-têtes de mise en sessions IBM Tealeaf                   |

---

## Chapitre 6. Code d'échantillon

Ce chapitre contient un échantillon de code pour IBM Tealeaf Structure de journalisation mobile d'Android CX.

---

### Comment instrumenter les contrôles TextView

Comme les contrôles TextView sont utilisés pour les champs de texte, vous devez instrumenter OnFocusChangeListener pour obtenir du temps de passage et d'autres données afin de savoir à quel moment un utilisateur commence et termine de saisir du texte.

```
// Obtenez un contrôle TextView
final EditText nameEditText = (EditText) findViewById(R.id.nameEditText);
// Créez un OnFocusChangeListener
OnFocusChangeListener focusListen = nouveau OnFocusChangeListener () {
    public void onFocusChange(View view, boolean hasFocus){
        si(hasFocus == false){
            Tealeaf.logEvent(view, Tealeaf.TLF_ON_FOCUS_CHANGE_OUT);
        }
        autre{
            Tealeaf.logEvent(view, Tealeaf.TLF_ON_FOCUS_CHANGE_IN);
        }
    }
};
// Définissez OnFocusChangeListener sur le contrôle TextView
nameEditText.setOnFocusChangeListener(focusListen);
// Enregistrez le contrôle TextView
Tealeaf.registerFormField(nameEditText, this);
```

---

### Comment instrumenter les contrôles ExpandableListView

Pour les contrôles ExpandableListView, afin de savoir à quel moment un utilisateur développe ou réduit un contrôle, vous pouvez instrumenter OnGroupCollapseListener et OnGroupExpandListener.

```
// Obtenez un contrôle ExpandableListView
final ExpandableListView elv = (ExpandableListView) findViewById(R.id.elv);

elv.setOnChildClickListener(nouveau OnChildClickListener() {
    public boolean onChildClick(ExpandableListView parent, View view,
        int groupPosition, int childPosition, long id) {
        Tealeaf.logEvent(view);
        renvoie true;
    }
});

elv.setOnGroupCollapseListener(nouveau OnGroupCollapseListener() {
    public void onGroupCollapse(int groupPosition) {
        Tealeaf.logEvent(elv, Tealeaf.TLF_ON_GROUP_COLLAPSE);
    }
});

elv.setOnGroupExpandListener(nouveau OnGroupExpandListener(){
    public void onGroupExpand(int groupPosition) {
        Tealeaf.logEvent(elv, Tealeaf.TLF_ON_GROUP_EXPAND);
    }
});
```

---

## Comment instrumenter les contrôles SlidingDrawer

Pour les contrôle SlidingDrawer, afin de savoir à quel moment un utilisateur ouvre ou ferme un contrôle, vous pouvez instrumenter OnDrawerOpenListener et OnDrawerCloseListener.

```
// Obtenez un contrôle SlidingDrawer
final SlidingDrawer sd = (SlidingDrawer) findViewById(R.id.sd);

sd.setOnDrawerOpenListener(nouveau OnDrawerOpenListener() {
    public void onDrawerOpened(){
        Tealeaf.logEvent(slidingDrawer_c5, Tealeaf.TLF_ON_DRAWER_OPENED);
    }
});

sd.setOnDrawerCloseListener(nouveau OnDrawerCloseListener(){
    public void onDrawerClosed(){
        Tealeaf.logEvent(slidingDrawer_c5, Tealeaf.TLF_ON_DRAWER_CLOSED);
    }
});
```

---

## Comment masquer les contrôles

Le masque personnalisé est une fonctionnalité qui correspond à des ID spécifiés et des expressions régulières et qui effectue des substitutions de caractères. Dans l'exemple qui suit, le masque personnalisé convertit les valeurs actuelles en lettres fournies en remplacement. Si le masque personnalisé est défini sur false, il renvoie une chaîne vide. Vous devez spécifier le masque dans le fichier TLFConfigurableItem.properties qui se trouve dans le dossier de documents de l'application Android.

```
#Paramètres de masque
HasMasking=true
#Il peut s'agir d'une série d'id et d'expressions régulières délimités par une virgule
MaskIdList=com.tealeaf.sp:id\/EditText*,com.tealeaf.sp:id\login.password
#S'il est défini sur false il renverra une chaîne vide
HasCustomMask=true
#Il transformera des lettres minuscules en valeurs données
SensitiveSmallCaseAlphabet=x
#Il transformera des lettres majuscules en valeurs données
SensitiveCapitalCaseAlphabet=X
#Il transformera des symboles en valeurs données
SensitiveSymbol=#
#Il transformera des chiffres en valeurs données
SensitiveNumber=9
```

---

## Fonction d'échantillonnage KillSwitch côté serveur

Lorsque KillSwitch est activé dans la configuration client, Structure de journalisation mobile d'Android CX interroge l'URL de KillSwitch pour déterminer s'il est activé ou désactivé pour la session.

Si Structure de journalisation mobile d'Android CX est désactivé, la session n'est pas capturée et est exclue de l'échantillon de données.

Dans les échantillons de code qui suivent, l'URL KillSwitch renvoie 1 pour activer Structure de journalisation mobile d'Android CX et 0 pour désactiver Structure de journalisation mobile d'Android CX.

## Fonction d'échantillonnage pour ASPX

### killswitch.aspx

Cet échantillon ASPX affiche un retour de 1 afin d'activer Structure de journalisation mobile d'Android CX et de 0 afin de désactiver Structure de journalisation mobile d'Android CX.

```
<%@ Page Language="C#" AutoEventWireup="true"%>
<script runat="server">
    public int Sampler()
    {
        Random rand = nouveau Random();
        int nextRandom = rand.Next(1,100);
        int samplepercent = Convert.ToInt32(ConfigurationManager.AppSettings["rate"]);
        si(nextRandom <= samplepercent){
            retour 1;
        }
        autre{
            retour 0;
        }
    }
}
</script>
<%
    si (ConfigurationManager.AppSettings["killswitchtype"].Equals("percentagesample")) {
%>
    <%= Sampler() %>
<% } autre{ } %>
```

Figure 1. killswitch.aspx

### Fichier de configuration web.config pour ASPX

```
<?xml version="1.0"?>
<!--
    Pour plus d'informations concernant la configuration de votre configuration ASP.NET,
    consultez la page suivante : http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>

    <appSettings>
        <add key="killswitchtype" value="percentagesample"/>
        <add key="rate" value="50"/>
    </appSettings>
</configuration>
```

Figure 2. web.config

## Fonction d'échantillonnage pour JSP

### killswitch.jsp

Cet échantillon JSP affiche un retour de 1 afin d'activer Structure de journalisation mobile d'Android CX et de 0 afin de désactiver Structure de journalisation mobile d'Android CX.

- Si la requête ne dispose pas de paramètres, l'environnement de capture côté client est toujours désactivé.
- Si le paramètre de requête d'id existe, il est utilisé pour vérifier la liste blanche.

- Si le paramètre `randomsample` existe, le taux de pourcentage du fichier `config.properties` est utilisé pour déterminer la manière dont le serveur répond.

Les journaux de débogage sont générés si la propriété `debug` est définie sur `true`.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="java.util.Properties"%>
<%@page import="java.util.Date" %>
<%@ page import="java.net.*"%>
<%@ page import="java.io.*" errorPage=""%>
<%
    Flux InputStream = application
        .getResourceAsStream("/config.properties");
    Propriétés props = nouveau Properties();
    props.load(stream);
    Booléen DEBUG = false;

    DEBUG = ("true").equals(props.getProperty("debug"));
    Chaîne id = request.getParameter("id");
    Chaîne randomsample = request.getParameter("randomsample");
    Chaîne killSwitchResponse = "";
    Chaîne debugstr = "";

    // liste blanche
    si (id != null && !id.isEmpty()) {
        InputStream whitestream = application.getResourceAsStream(props
            .getProperty("WhiteListFile"));
        Entrée BufferedReader = nouveau BufferedReader(
            nouveau InputStreamReader(whitestream));
        Chaîne line = "";
        Booléen correspondance = false;
        pendant que ((line = input.readLine()) != null) {
            ligne = line.trim();
            si (line.equals(id)) {
                killSwitchResponse = "1";
                correspondance = true;
                rupture;
            }
        }
        input.close();
        si (!match) {
            killSwitchResponse = "0";
        }
    }

    // Si le commutateur d'arrêt est par taux d'échantillon
    autre si (randomsample != null) {
        int rand = (int) (Math.random() * 100);
        int sampleRate = Integer.parseInt(props
            .getProperty("samplerate"));
        si (rand <= sampleRate) {
            killSwitchResponse = "1";
        } autre {
            killSwitchResponse = "0";
        }
    }
    } autre {
        killSwitchResponse = "0";
    }
}

out.print(killSwitchResponse);
```

Figure 3. `killswitch.jsp`

```

//toujours indiquer le chemin à partir de la racine. Ainsi, vous êtes sûr
qu'il fonctionne à chaque fois.
Chaîne nameOfTextFile = props.getProperty("logfile");
PrintWriter pw;

si (DEBUG) {
    essai {
        pw = new PrintWriter(nouveau FileOutputStream(nameOfTextFile,
            true));
        Date date = new java.util.Date();
        debugstr = date.toString() + "\t";
        si (request.getQueryString() != null) {
            debugstr += request.getQueryString();
        }
        si("0".equals(killSwitchResponse))
            pw.println(debugstr + "\tDisable");
        autre
            pw.println(debugstr + "\tEnable");
        //nettoyage
        pw.close();
    } interception (IOException e) {
        out.println(e.getMessage());
    }
}
%>

```

## Fichier de configuration web.config pour JSP

```

WhitelistFile=whitelist.txt
samplerate =50
debug=true
logfile=/killswitchlog.txt

```

Figure 4. *config.properties*

## Fonction d'échantillonnage pour PHP

### killswitch.php

Cet échantillon PHP affiche un retour de 1 afin d'activer Structure de journalisation mobile d'Android CX et de 0 afin de désactiver Structure de journalisation mobile d'Android CX.

```

<?php
$ini_array = parse_ini_file("config.ini", true);
//print_r($ini_array);

// si échantillon par pourcentage
si($ini_array['configtype']['killswitchtype'] === 'percentagesample'){
    $sampleRate = intval($ini_array['percentagesample']['rate']);
    killbysamplerate($sampleRate);
}
// si échantillon par la liste blanche
autre {

}

fonction killbysamplerate($sampleRate){
    $randomnumber = rand(1,100);
    si($randomnumber <= $sampleRate){
        écho '1';
    }
    autre {
        écho '0';
    }
}

fonction killbywhitelist($whitelistpath){

}
?>

```

Figure 5. killswitch.php

## Fichier de configuration web.config pour PHP

; Il s'agit d'un échantillon de fichier de configuration  
; Les commentaires commencent par ';', comme dans le fichier php.ini

```

[configtype]
killswitchtype=percentagesample

[percentagesample]
taux = 50

[whitelist]
x
y
z

```

Figure 6. config.ini

---

## Chapitre 7. Traitement des incidents

---

### Conseils relatifs à l'utilisation de résolution

#### Inclusion de requête brute et d'en-têtes de réponse

A des fins de débogage, vous pouvez inclure la requête brute les en-têtes de réponse des données passées par Application de capture passive CX vers le canister.

Cette méthode est utile si vous ne voyez aucune donnée JSON qui analysée dans IBM Tealeaf.

**Remarque :** Inclure ces en-têtes dans vos sessions IBM Tealeaf peut considérablement augmenter les exigences de stockage pour les sessions mobiles. Activez ces options uniquement pour déboguer.

Dans l'onglet Application de capture passive CX **Pipeline**, définissez les propriétés suivantes.

#### Paramètre

Valeur

#### **Include Raw Request**

true

#### **Include Response Headers**

true

Voir "PCA Web Console - Pipeline Tab" dans le document *IBM Tealeaf Application de capture passive CX*.

---

### Gestion des problèmes client

La section qui suit décrit comment IBM Tealeaf Structure de journalisation mobile d'Android CX gère les pannes, les exceptions ou les autres problèmes d'application qui peuvent se produire sur le périphérique mobile ou le réseau.

#### Exceptions ou pannes

Les exceptions d'application sont consignées et signalées IBM Tealeaf au format JSON.

Pour les périphériques qui exécutent iOS, une tentative de transmission de l'exception en cours vers le serveur est effectuée. Une copie est ajoutée à l'ensemble des messages mis en file d'attente localement et est envoyée la prochaine fois que l'application est lancée.

Pour les périphériques Android, toutes les données locales du périphérique sont vidées. L'objet d'exception est transmis au serveur.

Voir "Tealeaf JSON Object Schema Reference" dans le document *IBM Tealeaf Client Framework Data Integration Guide*.

## Pannes d'alimentation

Les pannes d'alimentation sur un périphérique ne peuvent pas être anticipées ou gérées par le logiciel. Les données sont perdues.

**Remarque :** Une exception `TL Library Error: File Not Found` peut être causée par une interruption de l'application surveillée. Si l'utilisateur a mis l'application hors tension ou s'il l'a fermée, la tâche d'imputation est désactivée. Lorsque l'application redémarre, la bibliothèque commence à envoyer les messages JSON en file d'attente. Cependant, si certaines de ces images de référence ne sont plus disponibles, l'erreur `File Not Found` est générée.

- Si les échecs d'alimentation persistent, vous pouvez configurer l'application client afin d'effectuer des enregistrements sur le disque local à de plus petits intervalles, et d'effectuer des envois sur le serveur à des intervalles plus fréquents.
- Pour plus d'informations concernant la configuration de la taille du cache local cache, voir Chapitre 2, «Fichier de configuration», à la page 35.
- Pour plus d'informations concernant la configuration de POST du client, voir Chapitre 2, «Fichier de configuration», à la page 35.

## Commutateur d'arrêt

Si le périphérique ne parvient pas à se connecter à la page cible, il ne capture pas de données.

## Problèmes de réseau

S'il existe des problèmes de connectivité au réseau, ces événements consignés comme objets de connexion avec des détails sur les problèmes.

- Pour les problèmes GET dus aux interruptions de l'application, un objet d'exception est généré.
- Si la connexion réseau est interrompue, les actions utilisateur sont sauvegardées et envoyées plus tard.

## Mémoire faible ou stockage local

Si la mémoire ou le stockage local deviennent faibles, un message de journal personnalisé est généré.

Pour les périphériques qui exécutent iOS, les données utilisateur sont tronquées dans la mémoire jusqu'à ce qu'un espace de mémoire supplémentaire devienne disponible.

Pour les périphériques Android, toutes les données collectées sur le périphérique sont vidées, et Structure de journalisation mobile d'Android CX est désactivé pour le périphérique.

---

## Chapitre 8. Documentation et aide d'IBM Tealeaf

IBM Tealeaf fournit une documentation et une aide aux utilisateurs, développeurs et administrateurs.

### Affichage de la documentation du produit

L'intégralité de la documentation des produits IBM Tealeaf est disponible sur le site Web suivant :

<https://tealeaf.support.ibmcloud.com/>

Utilisez les informations du tableau suivant pour afficher la documentation des produits d'IBM Tealeaf :

Tableau 23. Obtention de l'aide

Pour afficher...	Procédez comme suit...
Documentation du produit	Dans le portail IBM Tealeaf , accédez à ? > <b>Documentation du produit.</b>
Aide d'une page sur le portail IBM Tealeaf	Dans le portail IBM Tealeaf , accédez à ? > <b>Aide de cette page.</b>

### Documents disponibles pour les produits IBM Tealeaf

Utilisez le tableau suivant pour afficher une liste des documents disponibles pour tous les produits IBM Tealeaf :

Tableau 24. Documentation disponible pour les produits IBM Tealeaf

Produits IBM Tealeaf	Documents disponibles
IBM Tealeaf CX	<ul style="list-style-type: none"><li>• <i>IBM Tealeaf Customer Experience - Guide de présentation</i></li><li>• <i>IBM Tealeaf CX Client Framework - Guide d'intégration des données</i></li><li>• <i>IBM Tealeaf CX - Guide de configuration</i></li><li>• <i>Guide d'IBM Tealeaf CX Cookie Injector</i></li><li>• <i>IBM Tealeaf CX - Guide des bases de données</i></li><li>• <i>Guide d'IBM Tealeaf CX Event Manager</i></li><li>• <i>IBM Tealeaf CX - Glossaire</i></li><li>• <i>IBM Tealeaf CX - Guide d'installation</i></li><li>• <i>Guide d'IBM Tealeaf CX PCA</i></li><li>• <i>IBM Tealeaf CX PCA - Notes sur l'édition</i></li></ul>

Tableau 24. Documentation disponible pour les produits IBM Tealeaf (suite)

Produits IBM Tealeaf	Documents disponibles
IBM Tealeaf CX	<ul style="list-style-type: none"> <li>• <i>Guide d'IBM Tealeaf CX RealTea Viewer Client Side Capture</i></li> <li>• <i>IBM Tealeaf CX RealTea Viewer - Guide d'utilisation</i></li> <li>• <i>IBM Tealeaf CX - Notes sur l'édition</i></li> <li>• <i>IBM Tealeaf CX Release - Guide de mise à niveau</i></li> <li>• <i>IBM Tealeaf CX Support - FAQ sur le traitement des incidents</i></li> <li>• <i>IBM Tealeaf CX - Guide de traitement des incidents</i></li> <li>• <i>Guide d'IBM Tealeaf CX UI Capture j2</i></li> <li>• <i>IBM Tealeaf CX UI Capture j2 - Notes sur l'édition</i></li> </ul>
IBM Tealeaf cxImpact	<ul style="list-style-type: none"> <li>• <i>IBM Tealeaf cxImpact - Guide d'administration</i></li> <li>• <i>IBM Tealeaf cxImpact - Guide d'utilisation</i></li> <li>• <i>IBM Tealeaf cxImpact - Guide de génération de rapports</i></li> </ul>
IBM Tealeaf cxConnect	<ul style="list-style-type: none"> <li>• <i>IBM Tealeaf cxConnect for Data Analysis - Guide d'administration</i></li> <li>• <i>IBM Tealeaf cxConnect for Voice of Customer - Guide d'administration</i></li> <li>• <i>IBM Tealeaf cxConnect for Web Analytics - Guide d'administration</i></li> </ul>
IBM Tealeaf cxOverstat	<i>IBM Tealeaf cxOverstat - Guide d'utilisation</i>
IBM Tealeaf cxReveal	<ul style="list-style-type: none"> <li>• <i>IBM Tealeaf cxReveal - Guide d'administration</i></li> <li>• <i>IBM Tealeaf cxReveal - Guide de l'API</i></li> <li>• <i>IBM Tealeaf cxReveal - Guide d'utilisation</i></li> </ul>
IBM Tealeaf cxVerify	<i>IBM Tealeaf cxVerify - Guide d'administration</i>
IBM Tealeaf cxView	<i>IBM Tealeaf cxView - Guide d'utilisation</i>
IBM Tealeaf CX Mobile	<ul style="list-style-type: none"> <li>• <i>IBM Tealeaf CX Mobile Android Logging Framework Guide</i></li> <li>• <i>IBM Tealeaf Android Logging Framework - Notes sur l'édition</i></li> <li>• <i>IBM Tealeaf CX Mobile - Guide d'administration</i></li> <li>• <i>IBM Tealeaf CX Mobile - Guide d'utilisation</i></li> <li>• <i>Guide d'IBM Tealeaf CX Mobile iOS Logging Framework</i></li> <li>• <i>IBM Tealeaf iOS Logging Framework - Notes sur l'édition</i></li> </ul>

---

## Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Contactez votre interlocuteur IBM IBM local pour plus de détails sur les produits et les services actuellement disponibles dans votre pays. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet en cours couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous accorde aucune licence sur ces brevets. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

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

Pour le Canada, veuillez adresser votre courrier à :

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

Pour toute demande au sujet des licences concernant les jeux de caractères codés sur deux octets (DBCS), contactez le service Propriété intellectuelle IBM de votre pays ou adressez vos questions par écrit à :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEF AUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

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

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

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Bay Area Lab  
1001 E Hillsdale Boulevard  
Foster City, California 94404  
U.S.A.

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

Le programme sous licence décrit dans le présent document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Livret Contractuel IBM, des Conditions internationales d'utilisation des Logiciels IBM ou de tout autre contrat équivalent.

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

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

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

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins

illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

#### LICENCE DE COPYRIGHT :

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

---

## Marques

IBM, le logo IBM et [ibm.com](http://ibm.com) sont des marques d'International Business Machines Corp., dans de nombreux pays. D'autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. Une liste de marques IBM est actuellement disponible sur Internet sur le site «Copyright and trademark information» à l'adresse [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Remarques sur les règles de confidentialité

Les produits IBM Software, notamment les logiciels sous forme de services ("Offre logicielles") peuvent utiliser des cookies ou d'autres technologies pour collecter des informations d'utilisation en vue d'améliorer l'expérience de l'utilisateur final, d'ajuster les interactions avec l'utilisateur final ou à d'autres fins. Un cookie est une donnée qu'un site Web peut envoyer à votre navigateur et qui peut ensuite être stockée sur votre ordinateur sous la forme d'une balise identifiant ce dernier. Dans la plupart des cas, aucune information personnelle n'est collectée par ces cookies. Si vous utilisez une Offre logicielle qui vous permet de collecter des informations personnelles via des cookies et des technologies similaires, tenez compte des spécificités suivantes.

En fonction de la configuration déployée, cette Offre logicielle peut utiliser des cookies de session et des cookies permanents qui collectent le nom de chaque utilisateur, ainsi que d'autres informations personnelles à des fins de gestion des sessions, de convivialité améliorée pour l'utilisateur ou d'autres objectifs de suivi de l'utilisation ou fonctionnels. Ces cookies peuvent être désactivés mais leur désactivation élimine également la fonctionnalité qu'ils activent.

Diverses juridictions régulent la collecte d'informations personnelles via les cookies et autres technologies similaires. Si la configuration déployée pour cette Offre logicielle vous permet, en tant que client, de collecter des informations personnelles d'utilisateurs finaux via des cookies et autres technologies, vous devez rechercher votre propre avis légal concernant les lois applicables à cette collecte de données, dont toutes les exigences de mention et d'accord, le cas échéant.

IBM exige que les Clients (1) fournissent un lien clair et visible vers les conditions d'utilisation du site Web du client (par exemple, les règles de confidentialité), avec

un lien vers les collectes de données et les pratiques d'utilisation d'IBM et du Client, (2) indiquent que des cookies et des alarmes Web/gifs invisibles sont placés sur l'ordinateur du visiteur par IBM pour le compte du Client, en expliquant l'objectif de cette technologie et (3) selon les conditions requises par la loi, obtiennent le consentement des visiteurs du site Web avant de placer les cookies et les alarmes Web/gifs par le Client ou IBM sur leurs unités.

Pour plus d'informations sur l'utilisation des diverses technologies, notamment des cookies, reportez-vous à la section intitulée Cookies, Web Beacons and Other Technologies d'IBM à l'adresse <http://www.ibm.com/privacy/details/us/en>.



