IBM Tealeaf
Version 10 Release 0.0.1
November 12, 2015

iOS SDK Release Notes

Note

Before using this information and the product it supports, read the information in “Notices” on page 13.

This edition applies to version 10, release 0, modification 0.1 of IBM Tealeaf iOS SDK and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. IBM Tealeaf iOS SDK Release Notes

These release notes pertain to IBM Tealeaf iOS SDK.

For more information about IBM Tealeaf iOS SDK, see the *IBM Tealeaf iOS SDK Guide* .

New features and changes in 10.1.0

New features

- The iOS SDK has been restructured. The installation instructions and configuration files are updated to reflect the restructure. All APIs function the same in older versions and the new restructured SDK.
- The static images in an application need to be moved to the replay server for replay to function properly. There are two tools available to collect the images from the application:
  - Target Simulator
  - iOS Image Capture Tool
- The Tealeaf iOS SDK has been integrated with Xamarin. There are limited iOS APIs that are available for creating applications in Xamarin.

Changes

- There are no additional changes at the time of this publication.

New features and changes in 9.0.2 and 9.0.2A feature pack

The feature pack contains modifications and additions to the JSON messages that are generated by the SDKs.

New features

- The JSON messages affected include:
  - Type 1 (Client Environment) - the definitions for orientation, width, height, deviceWidth, and deviceHeight were updated. The referrer field was added to the message.
  - Type 11 (Gestures) - scrollX and scrollY fields were added.
  - Type 12 (DOM Capture) - now shows the changes in a page to support the DOM diff service.

New features and changes in 9.0.2 and 9.0.2A

New features

- The documentation was updated and modified for use in the IBM® Knowledge Center. With the IBM Knowledge Center documentation, the user can:
  - Search across all of the Tealeaf® documents and across all IBM product documentation
  - Create collections of specific topics that they use frequently
  - Create pdf documentation of the collections they create

- IBM Tealeaf iOS SDK now has the means to capture geolocation information on the user's device. The application programmer must ask for user's permission to collect the geolocation data. The information can be collected automatically when the application:
  - Starts
  - Goes to the background
  - Returns from the background

  Geolocation information can be manually collected as specific points in the application with the geolocation API.
- Integrating the iOS SDK with Swift is now supported.

Changes

- JSON messages were added for geolocation capture events. There are now 13 JSON message types supported. The message added to the documentation in this release is:
  - Type 13: Geolocation - Geolocation information. If the user does not give permission to collect geolocation data, a geolocation type 13 error message is sent.
- Dynamic IDs are no longer assigned to controls in an application that do not have IDs assigned by the application developer. Instead, the XPath for the control is used to identify the control. A specific ID enables the CX user to reliably define events and report on controls.

New features and changes in 9.0.1 and 9.0.1A

New features

- IBM Tealeaf iOS SDK now has the means to capture gestures the user makes on a mobile application. The gestures that are captured include:
  - Tap
  - Tap and hold
  - Double tap
  - Swipe vertically
  - Swipe horizontally
  - Pinch open
  - Pinch close
- DOM Capture is now an option for iOS applications that cannot use PCA. DOM Capture relies on the Document Object Model (DOM), which provides a structured representation of the web page (document). The DOM Capture Service captures a "snapshot" of the rendered DOM.

Changes

- The process for integrating Tealeaf with Worklight® 6.1 and 6.2 has been updated.
- There were changes in the iOS APIs.
  The `(BOOL)logNSExceptionEvent:(NSException *)exception level:(kTLFMonitoringLevelType)level;` and `(BOOL)logNSExceptionEvent:(NSException *)exception dataDictionary:(NSDictionary*)dataDictionary level:(kTLFMonitoringLevelType)level;` APIs were deprecated on 09/09/2014. Three new APIs were added:

  - `(BOOL)logNSExceptionEvent:(NSException *)exception`
  - `(BOOL)logNSExceptionEvent:(NSException *)exception dataDictionary:(NSDictionary*)dataDictionary;`
  - `(BOOL)logNSExceptionEvent:(NSException *)exception dataDictionary:(NSDictionary*)dataDictionary isUnhandled:(BOOL)unhandled;`
- The hybrid bridge has been expanded so that Native iOS APIs are now accessible from the UI Capture JavaScript.

  The iOS TLFApplicationHelper APIs that are available to JavaScript are:
  - `-(void)enableTealeafFramework;`
  - `-(void)disableTealeafFramework;`
  - `-(void)requestManualServerPost;`
  - `-(BOOL)startNewTLFSession;`
  - `-(NSString*)currentSessionId;`
  - `-(BOOL)setConfigurableItem:(NSString*)configItem value:(id)value;`
  - `-(id)valueForConfigurableItem:(NSString*)configItem;`
  - `-(id)defaultValueForConfigurableItem:(NSString*)configItem;`
  - `-(void) addAdditionalHttpHeader:(NSString*)value forName:(NSString*)name;`

  The iOS TLFCustomEvent APIs that are available to JavaScript are:
  - `- (BOOL)logEvent:(NSString*)eventName;`
  - `- (BOOL)logEvent:(NSString*)eventName values:(NSDictionary*)values;`
  - `- (BOOL)logPrintScreenEvent;`
- JSON messages were added for gesture capture and DOM capture events. There are 12 JSON message types supported. The messages added to the documentation in this release are:
  - Type 1: Client State - Current state of client.
  - Type 2: Application Context - Message to indicate divisions in application view which could be current page/view/activity user is on.
  - Type 3: Connection - Any request/response application performs during capture.
  - Type 4: Control - User interface control that fires an event we listen to capture.
  - Type 5: Custom Event - Any custom log event from any place in application.
  - Type 6: Exception - Any exception application can throw.
  - Type 7: Performance - Performance data from a browser.
  - Type 8: Web Storage - An object containing information about local storage information on the browser.
  - Type 9: Overstat® Hover Event - An object containing information about mouse hover and hover-to-click activity.
  - Type 10: Layout - Current display layout of native page.
  - Type 11: Gesture - Gesture that fires a higher touch event Tealeaf listens to and capturse.
  - Type 12: DOM Capture - An object containing serialized HTML data (DOM snapshot) of the page.
- Instructions for logging exceptions were added to the documentation.

New features and changes in version 9.0 and 9.0A - Enhanced International Character Support (EICS)

This section describes the new features and changes introduced versions 9.0 and 9.0A Enhanced International Character Support (EICS) of IBM Tealeaf iOS SDK.

Build 9.0.0.23

New features

- IBM Tealeaf iOS SDK now has functions to log screen layouts for screenviews of native mobile app sessions. This lets you replay a mobile app session in cxImpact Browser Based Replay as you would an HTML web session instead of viewing the mobile app session as a series of screen captures. The screen layouts of the native mobile app sessions are captured in IBM Tealeaf JSON format. The screen layouts are then sent back to replay server. The replay server uses a template engine, which interprets the JSON into HTML format. You can then replay the screen lay out from the native mobile app session as HTML pages in IBM Tealeaf Browser Based Replay.

  To implement this functionality, use the `logScreenLayout` method.
- New settings are added to sessionize in `TLFConfigurableItems.properties`.
  - `CookieExpires` can be set to `true` if you want to add an expiration property to the cookie.
  - `SessionTimeout` is used to determine the period, in minutes, of the cookie expiration after inactivity.
  - `SessionTimeoutKillSwitch` can be set to `true` or `false`. Setting `SessionTimeoutKillSwitch` to `false` means that the session timeout user does not recheck the `KillSwitchURL` to see if it is responding.
- There is a new `TLFCustomEvent` class API; `logNSURLSession`. Use this API to log failures that occur when a connection is attempted; typically from `NSURLConnectionDelegatedidFailWithError` or when `sendSynchronousRequest` returns an error. The first parameter is the connection object, and the second parameter is the error that you received. This setting can only be manually configured.
- The type of device that is used during capture, `osType`, is now included in the environment data that is captured at initialization.
- Three new APIs are added as part of the `TLFApplicatoinHelper` class, which is available for manual instrumentation.
  - `-(BOOL) sessionizeRequest:(NSMutableURLRequest*)request;` - Use this API so that the IBM Tealeaf iOS SDK can add various Headers and Cookies that can be used to tie all the application session hits together on the server.
  - `-(BOOL) isTealeafHybridBridgeRequest:(NSURLRequest*)request webView:(UIWebView*)webView;` - The API determines whether the request is specific to and meant for the iOS SDK from the UI Capture .
  - `-(BOOL) InjectTealeafHybridBridgeOnWebViewDidFinishLoad:(UIWebView *)webView;` - Use this API to inject IBM Tealeaf specific JavaScript into your web page. The JavaScript injection helps transfer data from the UI Capture to the iOS SDK.
- Two settings are added to configure the dimension, size, and format of the screen captures in `TLFConfigurableItems.plist`.
  - `ScreenshotFormat` can be used to set the format of the screen capture. You can choose between `PNG` or `JPG` format.

  - PercentOfScreenshotsSize can be used to set the percentage of screen capture's original pixel dimensions at which posted screen captures are submitted.
- There are two properties newly added to TLFConfigurableItems.plist, LogViewLayoutOnScreenTransition and GetImageDataOnScreenLayout.
  - For auto instrumentation, you can use LogViewLayoutOnScreenTransition = YES, which automatically logs screen layout. For manual instrumentation, you can use logScreenLayoutWithViewController with your desired view.
  - Along with LogViewLayoutOnScreenTransition, you can configure GetImageDataOnScreenLayout. If GetImageDataOnScreenLayout = YES, the image is logged as base64. If GetImageDataOnScreenLayou = NO, only MD5 checksum and png or jpg images are logged. This results in smaller payloads in production.

New features and changes in version 8.8

This section describes the new features and changes introduced in the 8.8 version of IBM Tealeaf iOS SDK.

Build 8.8.2.22

Bug fixes

The following issues were resolved in this release of IBM Tealeaf iOS SDK.

- logPrintScreenEvent now returns an image that matches the point resolution of the device. Previously, this image was limited to 320x480 pixels. This fixes image ratio distortion and overall quality.
- Capturing the screen is moved to main thread as Apple mandates using UIKit framework on main thread. Uploading screen image binary data is moved to background thread to speed up the performance of the logPrintScreenEvent API.
- In prior IBM Tealeaf iOS SDK versions, a Linux API was used to get IP address of the device. On certain networks, it was observed that this API takes more processor cycles than necessary. To improve the performance of getting an IP address, this API is now replaced by a new Linux API.

Build 8.8.2.6

Bug fixes

IBM Tealeaf iOS SDK version 8.8.2.6 fixes an issue with the Camera Preview that displayed as a black screen in certain cases. Apple does not recommend any class, variable, or API from UIKit framework to be referenced from a background thread. As a result, the Camera Preview is occasionally displayed as a black screen. The part of the iOS SDK that referred to UIKit framework on a background thread is now moved to main thread per Apple's guidelines to fix this problem.

Build 8.8.1.56

New features

The following API names were changed.

File: TLFPublicDefinitions.h

```
@protocol TLFCustomControlDelegate <NSObject>
```

Table 1. @protocol TLFCustomControlDelegate <NSObject>

Deprecated API name	Added API name
`- (BOOL)isControlHidden;`	`- (BOOL)isTLFCustomControlHidden;`
`- (NSInteger)controlTag;`	`- (NSInteger)tagTLFCustomControl;`
`- (id)controlParent;`	`- (id)parentTLFCustomControl;`
`- (CGRect)controlFrame;`	`- (CGRect)frameTLFCustomControl;`
`- (UIColor*)controlColor;`	`- (UIColor*)colorTLFCustomControl;`
`- (UIColor*)controlBackgroundColor;`	`- (UIColor*)backgroundColorTLFCustomControl;`
`- (NSString*)controlText;`	`- (NSString*)textTLFCustomControl;`
`- (BOOL)isTextHidden;`	`- (BOOL)isTLFCustomControlTextHidden;`
`- (CGRect)textFrame;`	`- (CGRect)textFrameTLFCustomControl;`
`- (UIColor*)textColor;`	`- (UIColor*)textTLFCustomControlColor;`
`- (UIColor*)textBackgroundColor;`	`- (UIColor*)textTLFCustomControlBackgroundColor;`
`- (UIImage*)controlImage;`	`- (UIImage*)imageTLFCustomControl;`
`- (BOOL)isImageHidden;`	`- (BOOL)isTLFCustomControlImageHidden;`
`- (CGRect)imageFrame;`	`- (CGRect)imageFrameTLFCustomControl;`
`- (UIColor*)imageOpacity;`	`- (UIColor*)imageOpacityTLFCustomControl;`
`- (UIColor*)imageBackgroundColor;`	`- (UIColor*)imageBackgroundColorTLFCustomControl;`

```
@protocol TLFCustomControlDelegateX <NSObject>
```

Table 2. @protocol TLFCustomControlDelegateX <NSObject>

Deprecated API name	Added API name
`- (NSArray*) imageViews;`	`- (NSArray*) imageViewsTLFCustomControl;`
`- (NSArray*) controls;`	`- (NSArray*) controlsTLFCustomControl;`

Build 8.8.1.8

New features

New APIs were created to allow for manual instrumentation of the IBM Tealeaf iOS SDK.

Note: Manual instrumentation is not recommended because of the large configuration effort, high chance of errors, and possibility of incomplete coverage. If you choose to use manual instrumentation, you are responsible for implementing theses changes.

When using iOS SDK with Auto-Instrumentation turned `OFF`, there is set of actions that you must complete that Auto-Instrumentation would otherwise do. Below is the list of required actions.

- View Controller changes must be logged using the API `logAppContext` from the `TLFCustomEvent` class.

- HTTP Connection updates must be logged using the API logConnection from the TLFCustomEvent class.
- There are three logConnection APIs; one each for initialization, successful response, and failure.
- Button click events must be logged using API logClickEvent from the TLFCustomEvent.
- UITableViewCell tap events must be logged using the API logValueChangeEvent from the TLFCustomEvent class.
- Text change events for UITextField, UITextView, and UILabel must be logged using the API logTextChangeEvent from the TLFCustomEvent class.
- All NSURLMutableRequest objects should be sessionized using the API sessionizeRequest from the TLFApplicationHelper class.
- All requests that are made by UIWebView should be tracked from UIWebViewDelegate's shouldStartLoadWithRequest using the API isTealeafHybridBridgeRequest from the TLFApplicationHelper class.
- All web page loads should inject Tealeaf's hybrid bridge into the JavaScript from UIWebViewDelegate's webViewDidFinishLoad using the API InjectTealeafHybridBridgeOnWebViewDidFinishLoad from the TLFApplicationHelper class.

In the TLFConfigurableItems.plist file, which is located in the TLFResources.bundle, set DisableAutoInstrumentation flag to YES. This disables Auto-Instrumentation. By doing this, no method swizzling occurs, application state is not monitored, and screen changes or any other events are not automatically tracked.

Use the following information to manually track various events using the TLFCustomEvent class.

- -(BOOL)logAppContext:(NSString*)logicalPageName applicationContext:(NSString*)applicationContext referrer:(NSString*)referrer
- -(BOOL)logEvent:(NSString*)eventName values: (NSDictionary*)values;
- -(BOOL)logConnection:(NSURLConnection*)connection error: (NSError*)error

  This API should be used to log failure while making a connection; typically from NSURLConnectionDelegate's didFailWithError or when sendSynchronousRequest returns error. The first parameter is the connection object, and second parameter is the error that you received.
- -(BOOL)logConnection: (NSURLConnection*)connection response: (NSURLResponse*)response responseTimeInMilliseconds:(long long)responseTime;

  This API should be used to log successful connections; typically from NSURLConnectionDelegate's didReceiveResponse or when sendSynchronousRequest returns success. The first parameter is connection object. The second parameter is the response that you received, and the third is connection's response type in milliseconds.
- -(BOOL)logConnection:(NSURLConnection*)connection request: (NSURLRequest*)request;

  This API should be used to log connection initialization; typically before or after calling NSURLConnection's initWithRequest. The first parameter is connection object, and the second parameter is the request object.
- -(BOOL)logClickEvent:(UIView*)view data:(NSDictionary*)data;

This API should be used to log button click events. Call this from your button click event handlers. The first parameter view is the `UIButton` object on which click event happened. The second parameter is optional, and is for future use. You can pass `Nil` for now.

- `-(BOOL)logValueChangeEvent:(UIView*)view data: (NSDictionary*)data;`

  This API should be used to log `UITableViewCell` tap events. Call this from your `UITableViewDelegate's didSelectRowAtIndexPath`. The first parameter view is the `UITableViewCell` object on which tap event happened The second parameter is optional, and is for future use. You can pass `Nil` for now.
- `-(BOOL)logTextChangeEvent:(UIView*)view data: (NSDictionary*)data;`

  This API should be used to log text change events for `UITextField`, `UITextView`, and `UILabel`. Call this from your application where ever contents of these three controls changed. If you add the `UITextViewTextDidEndEditingNotification` observer, you can call it from there. The first parameter view is the object of any of `UITextField`, `UITextView`, and `UILabel` whose text was edited. The second parameter is optional, and is for future use. You can pass `Nil` for now.
- All APIs are blocking calls. They are all optional and should be called based on your application's design and state machine.
- All the APIs return `YES` if data is logged, and `NO` in case of failure. The console debug log shows the reason for failure.

The following items from the `TLFApplicationHelper` class must be used for manual instrumentation.

- `-(BOOL) sessionizeRequest:(NSMutableURLRequest*)request;`

  This API should be invoked so that the Tealeaf iOS SDK can add various Headers and Cookies that can be used to tie all the application session hits together on the server. Call this API as soon as you create the `NSMutableURLRequest` object, and before you start HTTP connection. The first parameter is the object of `NSMutableURLRequest` that the Tealeaf SDK updates.
- `-(BOOL) isTealeafHybridBridgeRequest:(NSURLRequest*)request webView:(UIWebView*)webView;`

  This API should be invoked from `UIWebViewDelegate's shouldStartLoadWithRequest`. The first parameter is object of `NSURLRequest`, and the second is object of the current `UIWebView`. The API determines if the request is specific to and meant for Tealeaf's iOS SDK from Tealeaf's JavaScript SDK. If it is, the API consumes the data that is sent by Tealeaf's JavaScript SDK. If not, handle the request inside your `shouldStartLoadWithRequest`. For example, if this API returns `YES`, ignore the request and return `NO` from `shouldStartLoadWithRequest`. It was not an actual page navigation request from your HTML or JavaScript. If this API returns `NO`, handle the request as it came from your own HTML page or JavaScript.
- `-(BOOL) InjectTealeafHybridBridgeOnWebViewDidFinishLoad: (UIWebView *)webView;`

  This API should be used to inject the Tealeaf specific JavaScript into your webpage. The JavaScript injection helps transfer data from Tealeaf's JavaScript UIC SDK to Tealeaf's Native iOS SDK. The first parameter is the object of `UIWebView` in which current webpage is loaded. Call it every time new page is loaded into the `UIWebView`. The recommended place is `UIWebViewDelegate's webViewDidFinishLoad`.

Build 8.8.1.7

New features

- IBM Tealeaf iOS SDK added a timer setting in TLFConfigurableItems.properties to create a new session when the application is moved to the background and then returned to the foreground after a set period. The SessionTimeout setting is the set time interval in minutes for this setting.
- TLFConfigurableItems.properties now contains a setting to control the message types that are sent back to the server. If the FilterMessageTypes is set to TRUE, only the MessageTypes included in the comma-separated list are sent back to the server. If set to FALSE, all message types are sent back to the server.
- Changes were made to the frameworks and code for iOS integration without TLFApplication. For more information about these changes see *Quick start for instrumenting your iOS application* in the *IBM Tealeaf iOS Logging Framwork Reference Guide*.
- Support was added for JPG processing on server for images coming from iOS devices. In TLFConfigurableItems.properties, you can configure ScreenshotFormat to accept PNG or JPG file formats.

  JPG is a compressed format while PNG is an uncompressed format. The size of a JPG file is approximately between 6 KB and 15 KB. Comparably, the size of a PNG file is approximately between 20 KB and 35 KB.

New features and changes in version 8.7

This section describes the new features and changes introduced in the 8.7 version of IBM Tealeaf iOS SDK.

Build 8.7.5.25

A connection call caused iOS Framework to loop and crash application when loading an XML file.

Refer to Technical Q&A QA1490 Building Objective-C static libraries with categories, available at https://developer.apple.com/library/mac/#qa/qa2006/qa1490.html. In your applications build settings under, **Other Linker Flags** you must add -ObjC.

Build 8.7.5.21

New features

- Support was added for Visibility, textColor, textAlphaColor, textBackgroundColor, textBackgroundAlphaColor, backgroundColor, backgroundAlphaColor for UITextField and UIButton controls.
- Enhancements were made to custom control tracking.
- A timer setting was added to create a new session after a period of time for iOS.

Bug fixes

- Changes were made to overwrite offset that came from a website and use the offset from native library.
- iOS serialNumber now starts at 1 instead of 0 as it previously had.
- iOs clientEnvironment object was not contained in the sessions array. This issue was fixed.
- Fixed issues with method swizzling into objective-c class hierarchies.

Build 8.7.5.16

Bug fixes

The kill switch now allows for a `?` character in a URL path.

Performance enhancements were made to the iOS to reduce crashes and increase functionality.

The iOS SDK is now ARC compliant to help with memory management.

Build 8.7.4.13

Bug fixes

Performance enhancements were made to the iOS SDK for tablet hybrid deployment, ARC-compliant iOS SDK, and iOS devices to reduce crashes and increase functionality.

In the iOS Logging Framework, the `clientEnvironment` section was empty and no environment events were fired. This issue was resolved.

When adding a Canister Server, the processing pop-up list of servers that are displayed for some time then disappeared. This bug is unrelated to the capture library. This issue was resolved.

The crash that is experienced in the TLE library was fixed.

The iOS SDK console log no longer shows *Unknown host* message.

Instead of using third party JSON parser and serializer, now use Apple's NSJSONSerialization class.

Chapter 2. IBM Tealeaf documentation and help

IBM Tealeaf provides documentation and help for users, developers, and administrators.

Viewing product documentation

All IBM Tealeaf product documentation is available at the following website:

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

Use the information in the following table to view the product documentation for IBM Tealeaf:

Table 3. Getting help

To view...	Do this...
Product documentation	On the IBM Tealeaf portal, go to **?** > **Product Documentation**.
IBM Tealeaf Knowledge Center	On the IBM Tealeaf portal, go to **?** > **Product Documentation** and select *IBM Tealeaf Customer Experience in the ExperienceOne Knowledge Center*.
Help for a page on the IBM Tealeaf Portal	On the IBM Tealeaf portal, go to **?** > **Help for This Page**.
Help for IBM Tealeaf CX PCA	On the IBM Tealeaf CX PCA web interface, select **Guide** to access the *IBM Tealeaf CX PCA Manual*.

Available documents for IBM Tealeaf products

The following table is a list of available documents for all IBM Tealeaf products:

Table 4. Available documentation for IBM Tealeaf products.

IBM Tealeaf products	Available documents
IBM Tealeaf CX	• *IBM Tealeaf Customer Experience Overview Guide* • *IBM Tealeaf CX Client Framework Data Integration Guide* • *IBM Tealeaf CX Configuration Manual* • *IBM Tealeaf CX Cookie Injector Manual* • *IBM Tealeaf CX Databases Guide* • *IBM Tealeaf CX Event Manager Manual* • *IBM Tealeaf CX Glossary* • *IBM Tealeaf CX Installation Manual* • *IBM Tealeaf CX PCA Manual* • *IBM Tealeaf CX PCA Release Notes*

Table 4. Available documentation for IBM Tealeaf products (continued).

IBM Tealeaf products	**Available documents**
IBM Tealeaf CX	• *IBM Tealeaf CX RealiTea Viewer Client Side Capture Manual* • *IBM Tealeaf CX RealiTea Viewer User Manual* • *IBM Tealeaf CX Release Notes* • *IBM Tealeaf CX Release Upgrade Manual* • *IBM Tealeaf CX Support Troubleshooting FAQ* • *IBM Tealeaf CX Troubleshooting Guide* • *IBM Tealeaf CX UI Capture j2 Guide* • *IBM Tealeaf CX UI Capture j2 Release Notes*
IBM Tealeaf cxImpact	• *IBM Tealeaf cxImpact Administration Manual* • *IBM Tealeaf cxImpact User Manual* • *IBM Tealeaf cxImpact Reporting Guide*
IBM Tealeaf cxConnect	• *IBM Tealeaf cxConnect for Data Analysis Administration Manual* • *IBM Tealeaf cxConnect for Voice of Customer Administration Manual* • *IBM Tealeaf cxConnect for Web Analytics Administration Manual*
IBM Tealeaf cxOverstat	*IBM Tealeaf cxOverstat User Manual*
IBM Tealeaf cxReveal	• *IBM Tealeaf cxReveal Administration Manual* • *IBM Tealeaf cxReveal API Guide* • *IBM Tealeaf cxReveal User Manual*
IBM Tealeaf cxVerify	• *IBM Tealeaf cxVerify Installation Guide* • *IBM Tealeaf cxVerify User's Guide*
IBM Tealeaf cxView	*IBM Tealeaf cxView User's Guide*
IBM Tealeaf CX Mobile	• *IBM Tealeaf CX Mobile Android Logging Framework Guide* • *IBM Tealeaf Android Logging Framework Release Notes* • *IBM Tealeaf CX Mobile Administration Manual* • *IBM Tealeaf CX Mobile User Manual* • *IBM Tealeaf CX Mobile iOS Logging Framework Guide* • *IBM Tealeaf iOS Logging Framework Release Notes*

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: http://www.ibm.com/privacy/details/us/en section entitled "Cookies, Web Beacons and Other Technologies."

Printed in USA