

INFOnline
SZM Mobile Applications

INFOnline SZM-Library Integration Guide

Plattform:
WindowsPhone 8.0/8.1
SZM-Library Version: 1.1.3

INFOnline GmbH
Forum Bonn Nord
Brühler Str. 9
53119 Bonn

Tel.: +49 (0) 228 / 410 29 - 77
Fax: +49 (0) 228 / 410 29 - 66

<http://www.INFOnline.de>
support@INFOnline.de

1 Einleitung	1
1.1 SZM Mobile Applications	1
1.2 Unterstützte Plattformen	1
1.2.1 Hybride APPs	2
1.3 Über dieses Dokument	3
2 INFOonline SZM-Library (WP8.0/8.1)	4
2.1 Bereitstellung	4
2.2 Support	4
2.3 Anforderungen	5
2.3.1 Betriebssystem	5
2.3.2 Kompatibilität	5
2.4 Funktionsweise	5
2.4.1 Wann muss die SZM-Library aufgerufen werden?	5
2.4.2 Offline-Nutzung	5
2.4.3 Übertragung der Messdaten	6
2.4.4 Opt-out-Funktion und Datenschutzerklärung	6
3 Integration	7
3.1 IOLib Dateien	7
3.2 Integration der IOLib	7
3.3 SZM-Library Funktionen	9
3.3.1 Start einer Session:	9
3.3.2 Logging eines Events:	9
3.3.3 Versand der Messdaten	13
3.3.4 Session beenden	13
3.3.5 Einbindung Opt-out-Funktion	13
3.4 Hybrid-Messung	14
3.4.1 Manuelle Übergabe der Messdaten	14
3.5 Debug-Informationen	14
4 Vorgaben zum Aufruf der SZM-Library	15
4.1 Allgemeines	15
4.1.1 Interpretation von Events als mobile PI	15
4.2 Richtlinien zur Vergabe der Codes	16
4.3 Events	17
4.3.1 Automatisch durch die SZM-Library gemessene Events	17
4.3.2 PI-Events	18
4.3.3 Non-PI-Events	19
4.3.4 Sonderfall Event „ViewRefreshed“	20

1 Einleitung

1.1 SZM Mobile Applications

Seit annähernd 10 Jahren betreibt die INFOOnline GmbH mit dem SZM (Skalierbaren Zentralen Messverfahren) das etablierte Standardsystem zur technischen Messung von Seitenzugriffen auf Internetangebote nach den einheitlichen Standards der Arbeitsgemeinschaft Online-Forschung (AGOF) und der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern (IVW).

Die Anforderungen an das Messsystem haben sich seitdem erheblich weiterentwickelt. Neue Nutzungsarten des Internets, die Wünsche unserer Kunden und Partner sowie neue technologische Möglichkeiten führten zu dem Entschluss, eine neue Generation unseres Messsystems zu entwickeln.

Wichtigste Grundlage ist dabei, dass die etablierten Standards der IVW und der AGOF durch das neue System erfüllt werden. Aus diesem Grunde wird das neue SZM in enger Zusammenarbeit mit der spring GmbH – dem Lizenzgeber des bisherigen SZM-Systems – entwickelt. Um die heutigen und zukünftigen Anforderungen der IVW und der AGOF abbilden zu können, werden auch diese beiden Partner in die Definition und Entwicklung des neuen Systems eingebunden.

Der erste Meilenstein für das neue SZM soll es werden, die Anforderungen an die technische Messung von mobilen Seitenzugriffen für die AGOF Studie ‚mobile facts‘ zu erfüllen. Gleichzeitig müssen die grundlegenden Prüfungen durch die IVW und die Datenschnittstellen für die Partner ermöglicht werden.

1.2 Unterstützte Plattformen

Die SZM Nutzungsmessung unterstützt folgende mobile Plattformen:

1. Android (ab 2.3.3)
2. iOS (iPhone, iPod touch und iPad, ab iOS Version 6.0)
3. Windows Phone 8 / 8.1
4. Blackberry (6 und 7)

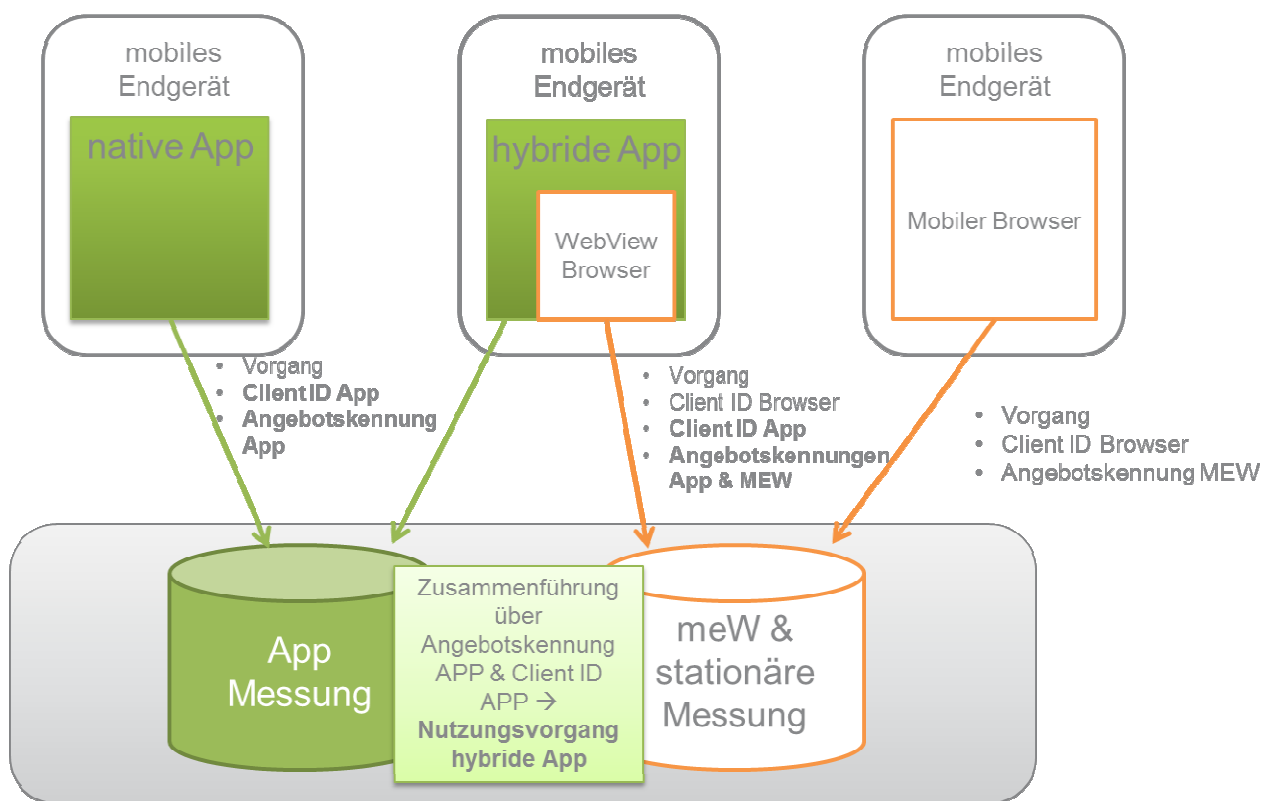
Dieses Dokument beschreibt ausschließlich die Integration und Nutzung der **SZM-Messung für Windows Phone 8.0/8.1 Apps**.

1.2.1 Hybride APPs

Die SZM-Library ist in der Lage, die Nutzung von Hybrid APPs zu messen, d.h. auch Nutzeraktionen innerhalb mobiler Inhalte, welche in sog. WebViews dargestellt werden, können erfasst und mit den Messdaten des nativen APP-Rahmens zusammengeführt werden.

Voraussetzung hierfür ist, dass die über den WebView aufgerufenen Webseiten ebenfalls mit dem SZM-Tag für mobile-enabled Websites vertaggt sind. Außerdem muss der von der SZM-Library bereitgestellte WebView verwendet werden, um die beiden Mess-Datensätze aus der APP Messung und der MEW Messung zusammenzuführen. Die APP und die aus der APP aufgerufenen MEWs müssen unterschiedliche, von INFOOnline vergebene Angebotskennungen verwenden. Nachfolgendes Schaubild zeigt eine Übersicht mobiler Nutzung im SZM:

Messung mobiler Nutzung SZM



Für technische Details zur Einbindung der Hybridmessung s. Kapitel [3.4](#).

1.3 Über dieses Dokument

Das vorliegende Dokument dient dazu, alle notwendigen Informationen für die technische Integration der SZM-Library in die teilnehmenden APPs zur Verfügung zu stellen.

Es ist in 4 Kapitel gegliedert:

1. Einleitung
2. Rand- und Rahmenbedingungen der INFOnline SZM-Library
3. Technische Informationen zur Integration der SZM-Library WP 8
4. Vorgaben zur Verwendung der SZM-Library im Kontext der SMZ Mobile Applications Messung

2 INFOnline SZM-Library (WP 8.0/8.1)

Die INFOnline SZM-Library für WP 8 (im Folgenden auch „IOLib“ genannt) ist eine Software-Bibliothek, welche die Nutzung von WP 8.0/8.1-Apps erfasst, speichert und zum Zwecke der Validierung und Kontrolle an ein geeignetes Backend versendet. INFOnline stellt dieses Backend bereit.

2.1 Bereitstellung

Die IOLib WP8 wird von INFOnline zum Download zur Verfügung gestellt. Die Zugangsdaten erhalten Sie per E-Mail.

Im Download Bereich befinden sich

- die Release Notes als Textdatei
- **MessLib:** Verzeichnis **INFOnlineLibrary** mit **Unterverzeichnissen für Windows Phone mit der entsprechenden DLL „INFOnlineLib.dll“** (s. Kap. 3)
- **Samples:** Verzeichnis **INFOnlineLibrarySample** mit **Unterverzeichnissen für Windows Phone:** Beispiel-APPs mit integrierter IOLib

2.2 Support

Der Support der Messung SZM wird von INFOnline bereitgestellt.

Dieser Support umfasst:

- organisatorischen Support (z.B. Fragen zum Ablauf)
- technischen Support der Messung (z.B. Analyse der eingehenden Requests)
- Entwickler-Support (z.B. Fragen zum Einbau der SZM-Library)

Das Service & Support-Team ist von Montag bis Freitag von 9 bis 18 Uhr erreichbar via

Telefon: 0228 / 410 29 - 77

Mail: support@INFOnline.de

2.3 Anforderungen

Folgende Entwicklungsumgebungen sind für die Integration der SZM-Library für WP8 erforderlich:

- WP 8.0/8.1: Mindestens Visual Studio 2010 Express für Windows Phone

Im Installationsumfang von Visual Studio enthalten ist neben der eigentlichen Entwicklungsumgebung das jeweilige Windows (Phone) Developer Toolkit, welches natürlich auch erforderlich ist.

2.3.1 Betriebssystem

Die SZM-Library setzt für den Betrieb Windows Phone 8.0 oder 8.1 voraus.

2.3.2 Kompatibilität

Eine vollständige Aufwärtskompatibilität kann u.U. nicht gewährleistet werden.

2.4 Funktionsweise

2.4.1 Wann muss die SZM-Library aufgerufen werden?

Der Aufruf der Funktionen der IOLib innerhalb der APP ist an bestimmte Events gebunden. Hierzu werden für die APP-Anbieter seitens der AGOF und IVW Vorgaben bzw. Empfehlungen formuliert, an welchen Stellen der APP (bzw. bei welchen Nutzer-Aktionen) die SZM-Library aufgerufen werden soll und welche Informationen zu übermitteln sind.

Die Vorgaben zum Aufruf der IOLib innerhalb der APP sind in Kapitel 4 Vorgaben zum Aufruf der SZM-Library beschrieben.

2.4.2 Offline-Nutzung

Die IOLib unterstützt die Nutzung mobiler APPs ohne aktive Internet-Verbindung. Die Events der Offline-Nutzung werden gespeichert und bei nächster Gelegenheit (sobald eine Internet-Verbindung besteht) an das Mess-Backend übertragen. Pro Event werden ein Timestamp, der jeweilige InternetConnection-Status sowie weitere Daten erfasst und dokumentieren damit den Zeitpunkt und die Rahmenbedingungen der Offline-Nutzung.

2.4.3 Übertragung der Messdaten

Um die Datenübertragung möglichst effizient zu gestalten und die Offline-Nutzung zu ermöglichen, werden die Mess-Daten nicht direkt synchron zum Zeitpunkt der Messung an das Backend übertragen, sondern in einer „Messdaten-Queue“ gesammelt.

Der zu übertragende Datensatz wird kontinuierlich mit den neuen Messdaten konkateniert, sobald ein bestimmtes Größenlimit erreicht ist, wird ein neuer Datensatz gebildet und der vorherige zum Versand freigegeben.

Der Versand der Daten selbst findet asynchron statt. Dadurch wird die Interaktion des Benutzers mit der APP nicht verzögert oder blockiert.

2.4.4 Opt-out-Funktion und Datenschutzerklärung

Die User einer APP müssen informiert werden, dass die APP die Nutzeraktionen misst und an das Mess-System der INFOnline überträgt. Für diesen Zweck stellt INFOnline eine Datenschutzerklärung zur Verfügung, die unter <https://www.infonline.de/de/extra/downloads/> abgerufen werden kann. Bitte binden Sie diesen Text an entsprechender Stelle in die APP ein.

Darüber hinaus muss dem Nutzer eine Opt-out Funktion gegeben werden. Die Implementierung dieser obliegt dem Entwickler der jeweiligen APP. Nach Einbindung der Funktion können die Nutzer der APP das Opt-out aktivieren und deaktivieren. Sofern Opt-out aktiviert wird, wird kein Zählimpuls ausgelöst.

Für technische Details der Einbindung siehe Kapitel 3.3.5.

3 Integration

Dieses Kapitel beschreibt die technische Integration der SZM-Library.

3.1 IOLib Dateien

Die SZM-Library für iOS umfasst folgende Dateien

- [RELEASE_NOTES.txt](#)
Diese Datei enthält Informationen zum Release der IOLib.
- [INFOnlineLibrary](#)
Die IOLibrary zur Erfassung der Nutzungsdaten einer WP 8.0/8.1 APP
- [INFOnlineLibrarySample](#)
Ein Beispiel-Projekt, welches den Einsatz der IOLibrary für WP 8.0/8.1 demonstriert

3.2 Integration der IOLib

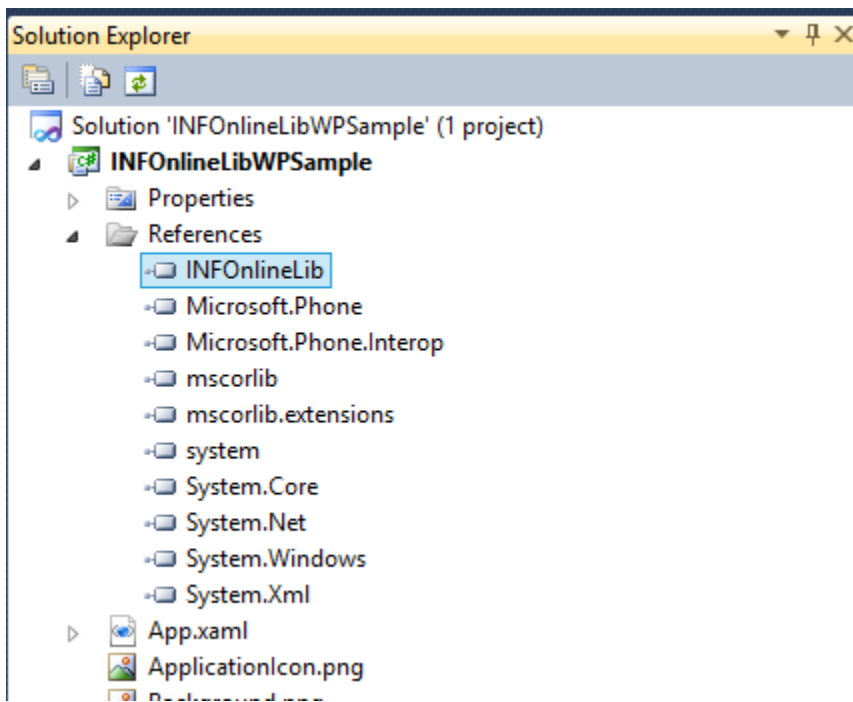
Die Integration erfolgt in wenigen einfachen Schritten.

Im Betriebssystem:

Die ZIP-Datei der IOLib für WP8 herunterladen und entpacken.

In Visual Studio:

Im entsprechenden Projekt der zu entwickelnden App einen neuen Verweis auf die entpackte Datei „INFOnlineLib.dll“ hinzufügen.



In Visual Studio: App.xaml.cs

Mit `IOLSession.Current.Start()` wird die IOLib initialisiert und eine neue Session gestartet. Wichtig: Dies sollte immer in der Methode `Application_Launching` (Windows Phone) der Klasse `App` geschehen, da nur dadurch gewährleistet ist, dass alle automatisch geloggtten Events vollständig erfasst werden.

Beispiel:

```
private void Application_Launching(object sender, LaunchingEventArgs e) {
    IOLSession.Current.Start("iamtest");
}
```

In Visual Studio: Code

Über die Eigenschaft „DebugEnabled“ kann der Debugmodus aktiviert werden. Debugnachrichten können danach über die Registrierung des Events „OnDebugMessage“ erhalten werden. Es muss dabei beachtet werden, dass nur dann alle Initialisierungsmeldungen dem Debughandler zur Verfügung stehen, wenn dieser vor dem Start der Session registriert wird.

Beispiel:

```
IOLSession.Current.DebugEnabled = true;
IOLSession.Current.OnDebugMessage +=
    new EventHandler<DebugEventArgs>(OnDebugMessage);
```

3.3 SZM-Library Funktionen

Die SZM-Library bietet die im Folgenden beschriebenen Funktionen:

3.3.1 Start einer Session:

```
public void Start(string offerIdentifier) // WindowsPhone
```

Die IOLib **muss** vor der Erfassung der Events gestartet werden. Dabei muss die Angebotskennung der APP als Parameter übergeben werden.

Parameter:

- **Angebotskennung (mandatory)**
Die eindeutige Kennung des Angebots der jeweiligen APP. Angebotskennung wird von INFOnline pro APP und pro Betriebssystem eindeutig vergeben.

Beispiel (Windows Phone):

```
IOLSession.Current.Start("offerXYZ");
```

3.3.2 Logging eines Events:

Die Messdaten werden mittels des Aufrufs **LogEvent** erfasst. Dabei können bis zu drei Parameter übergeben werden, zwei davon sind optional.

```
public void LogEvent(INFOnlineLib.Events.Event event)
```

```
public void LogEvent(INFOnlineLib.Events.Event event,  
                    string category, string comment)
```

Der erste Aufruf ist eine Convenience-Funktion, welche intern die Letztgenannte aufruft. Die fehlenden Werte werden dann um Default-Werte ergänzt.

Einige der Events werden durch die IOLib automatisch erfasst. Weitere Details hierzu finden sich im Anhang unter 4.3 Events.

Parameter:

- **event (mandatory)**

Die einzelnen Events können verschiedene Zustände einnehmen. So kann ein Download z.B. gestartet, durch den User abgebrochen, erfolgreich durchgeführt oder fehlerhaft beendet worden sein.

Folgende Zustände können gemessen werden:

- DeviceOrientationChanged
- BackgroundTaskStart
- BackgroundTaskEnd
- ViewAppeared
- ViewDisappeared
- ViewRefreshed
- ApplicationStart
- ApplicationTerminate
- ApplicationCrashed
- ApplicationEnterBackground
- ApplicationEnterForeground
- DataCancelled
- DataRefresh
- DataSucceeded
- DataFailed
- DocumentOpen
- DocumentClose
- DocumentEdit
- DownloadCancelled
- DownloadStarted
- DownloadSucceeded
- DownloadFailed
- GameStarted
- GameFinished
- GameWon

- GameLost
- GameNewHighscore
- GameNewAchievement
- GestureShake
- IAPStarted
- IAPFinished
- IAPCancelled
- InternetConnectionLost
- InternetConnectionEstablished
- InternetConnectionSwitchedInterface
- LoginSucceeded
- LoginFailed
- LoggedOut
- AudioPlay
- AudioPause
- AudioStop
- AudioNext
- AudioPrevious
- AudioReplay
- AudioSeekBack
- AudioSeekForward
- VideoPlay
- VideoPause
- VideoStop
- VideoNext
- VideoPrevious
- VideoReplay
- VideoSeekBack
- VideoSeekForward
- PushReceived
- UploadCancelled
- UploadStart

- UploadSucceeded
- UploadFailed
- HardwareButtonPushed
- OpenAppMaps
- OpenAppOther

Für weitere Details zu den messbaren Events: s. Anhang

- **category (optional): Inhaltscode**

Der Inhaltscode wird im Parameter "category" übermittelt. Dieser Code wird vom Anbieter selbst festgelegt (syntaktische Vorgaben siehe 4.2 Richtlinien zur Vergabe der Codes). Der Code dient zur inhaltlichen Kennzeichnung des angezeigten Content und wird vom Anbieter im INFOOnline Kundencenter dem IVW Kategoriensystem 2.0 zugeordnet.

Der Anbieter entscheidet anhand der im Kapitel 4 Vorgaben zum Aufruf der SZM-Library beschriebenen Richtlinien, ob ein Event eine mobile PI im Sinne der IVW Richtlinien darstellt. Wenn ein Event unter die Definition einer mobilen PI fällt, ist zwingend ein Inhaltscode mitzugeben. Stellt ein Event keine mobile PI dar, soll *string.empty* übergeben werden. Die Länge dieses Feldes ist nicht beschränkt.

- **Kommentar**

Kommentarfeld. Die Länge dieses Feldes ist nicht beschränkt. Übergabe dieses Wertes ist optional, ist er nicht definiert, soll *string.empty* übergeben werden.

Beispiele:

ViewAppeared

```
IOLSession.Current.LogEvent(
    EventTypes.ViewAppeared,
    "myComment",
    "myCategory"
);
```

ViewRefreshed

```
IOLSession.Current.LogEvent(
    EventTypes.ViewRefreshed,
    "myComment",
    "myCategory"
);
```

AudioPlay

```
IOLSession.Current.LogEvent(
    EventTypes.AudioPlay,
);
```

3.3.3 Versand der Messdaten

```
public void SendLoggedEvents()
```

Die IOLib steuert den Versand der Messdaten selbständig und völlig transparent für den Enduser. Um den Versand der Daten zu forcieren, **kann** SendLoggedEvents aufgerufen werden. Die IOLib versucht dann, die Messdaten sofort zu versenden bzw. alternativ nochmals, sobald eine Datenverbindung aufgebaut wurde.

Parameter:

- -

Beispiel:

```
IOLSession.Current.SendLoggedEvents();
```

3.3.4 Session beenden

```
public void TerminateSession()
```

Die aktive Session der IOLib kann explizit beendet werden. Dies ermöglicht ein Opt-out während der App-Laufzeit. Die bis dahin erfassten Daten werden nicht mehr versendet.

Beispiel:

```
IOLSession.Current.TerminateSession();
```

HINWEIS: IOLib Session muss anschließend neu gestartet werden! (s. 3.3.1)

3.3.5 Einbindung Opt-out-Funktion

Den Nutzer einer APP muss eine Opt-out Funktion gegeben werden. Die Implementierung obliegt dem Entwickler der jeweiligen APP und sollte bei Aktivierung durch den Benutzer dazu führen, dass die SZM-Library entweder gar nicht initialisiert wird oder die laufende Session explizit beendet wird (s. 3.3 Session beenden).

Nach Einbindung der Funktion können die Nutzer der APP das Opt-out aktivieren und deaktivieren. Sofern Opt-out aktiviert wird, wird kein Zählimpuls ausgelöst.

HINWEIS: Wird die laufende Session explizit beendet, dann werden alle bis dahin erfassten, aber noch nicht versandten Messdaten verworfen.

Wird das Opt-out revidiert, dann sollte die MessLib wieder gestartet werden (s. 3.3 Session starten).

3.4 Hybrid-Messung

Um die Messung von Hybrid-APPs zu ermöglichen, muss das WebBrowser-Control an die IOLib gebunden werden. Dies geschieht durch den Aufruf der Methode:

```
public void AttachToWebBrowser(Microsoft.Phone.Controls.WebBrowser browser)
```

3.4.1 Manuelle Übergabe der Messdaten

```
public void GetMultiIdentifizier()
```

Falls die Daten der Hybridmessung manuell übergeben werden sollen, können diese mit der Methode „GetMultiIdentifizier“ abgerufen werden.

Beispiel:

```
string multiIdentifizier = IOLSession.Current.GetMultiIdentifizier();
```

3.5 Debug-Informationen

Zum Zweck der allgemeinen Fehleranalyse und insbesondere des Versands der Messdaten kann die SZM-Library in einen Debug-Modus versetzt werden. In diesem Debug-Modus können über das Abonnement des Events *OnDebugMessage* die Debugmeldungen gelesen werden.

Beispiel:

```
IOLSession.Current.DebugEnabled = true; // enable debug mode
IOLSession.Current.OnDebugMessage += new EventHandler<DebugEventArgs>((obj, args) =>
{
    Deployment.Current.Dispatcher.BeginInvoke(() => {
        Console.WriteLine(args.Message);
    });
});
```


4 Vorgaben zum Aufruf der SZM-Library

4.1 Allgemeines

Die Erfassung der APP-Nutzung durch den Benutzer erfolgt, indem die APP die SZM-Library bei definierten Ereignissen, welche eine Nutzer-Interaktion kennzeichnen, aufruft. Die Nutzer-Interaktion wird als Event bezeichnet.

WICHTIG: Die SZM-Library muss von der APP bei Eintreten des Events explizit aufgerufen werden.

Weiterhin misst die IOLib bestimmte System oder APP-spezifische Werte automatisch. **Zu diesem Zweck muss die Integration der IOLib exakt wie im Kapitel 3.2 beschrieben erfolgen.**

4.1.1 Interpretation von Events als mobile PI

Die nachfolgend aufgeführten Vorgaben definieren, wie die Lib im Kontext der SZM APP Messung zu verwenden ist.

Aus technischer Sicht wird zwischen 2 Typen von Events unterschieden:

1. PI-Events

Bei den PI-Events wird der Event dazu benutzt, analog zum stationären Web, eine PageImpression zu erzeugen. Diesem Event muss ein Code zugeordnet werden. Dieser Code kann anschließend den unterschiedlichen Kategorien zugeordnet werden und dient als Grundlage für die Bildung von Belegungseinheiten. Bei den PI-Events sind die Vorgaben zur Mobile Impression der IVW zu beachten:

„Eine Mobile Impression ist eine Nutzeraktion innerhalb eines mobilen Angebots, die zu einem Aufruf eines Werbemittels führt oder führen könnte. Jede Nutzeraktion darf nur einmal gezählt werden. Nutzeraktionen, die zu keiner potentiellen Werbeauslieferung führen, dürfen nicht gezählt werden.“

„Voraussetzungen für die Zuweisung einer MI zu einem Angebot:

Der ausgelieferte Inhalt muss (bei mobile-enabled Websites) den FQDN bzw. (bei APPs) den APP-Namen des Angebots (oder Alias/Redirect) oder den zugewiesenen MEW- oder APP-Namen des Angebots tragen.

Nutzeraktion:

Eine MI wird ausgelöst durch eine vom Nutzer durchgeführte Aktion.

Darunter fallen ebenfalls: Reload, Öffnen einer APP, Öffnen eines Browsers

Keine Nutzeraktion:

Aufruf eines Inhalts durch eine automatische Weiterleitung (außer Redirects und Alias), automatischer Reload, das Aufrufen eines Inhaltes beim Schließen (auch: Background) eines Browserfensters oder einer APP, das Aufrufen von Inhalten über Robots/Spider und Ähnliches.

Keine Mobile Impression:

Das Scrollen innerhalb eines bereits geladenen Inhalts.“

2. Non-PI- Events

Non-PI-Events sind Nutzeraktionen, die als Event im SZM System erfasst werden, jedoch nicht zur Zählung einer Mobile Impression führen. Diesem Event **darf kein Code** zugeordnet werden. Beispiele für Non-PI-Events sind

- automatisch von der IOLib erfasste Events (siehe 4.3.1 Automatisch durch die SZM-Library gemessene Events)
- vom Anbieter festgelegte Events, welche keine mobile PI darstellen und dennoch als Events gemessen werden sollen, um z.B. die Nutzung der APP durch die Nutzer besser nachvollziehen zu können.

4.2 Richtlinien zur Vergabe der Codes

Bei den **PI-Events** ist der Code als eindeutige Kennzeichnung des angezeigten Inhalts mitzugeben. Der Code wird vom APP-Anbieter spezifiziert.

Bei der Spezifikation der Inhalts-Codes sind die Code-Richtlinien der INFOnline zu beachten

- Länge der Codes: Ein Code darf maximal 255 Zeichen enthalten
- Anzahl der Codes: es dürfen maximal 2000 Codes verwendet werden
- Erlaubte Zeichen: a-z, A-Z, 0-9; Komma „,“; Bindestrich „-“; Unterstrich „_“; Slash „/“

Eine ausführliche Dokumentation der INFOnline Code-Richtlinien finden Sie unter

https://www.infonline.de/downloads/9-1356-548/INFOnline_Neue_Code-Richtlinien.pdf

4.3 Events

In den nachfolgenden Tabellen sind Events aufgeführt, welche in der Messung erhoben werden, bzw. erhoben werden können. Unter welchen Umständen ein Event zu einer PageImpression führen kann wird im Folgenden ebenfalls erläutert.

4.3.1 Automatisch durch die SZM-Library gemessene Events

Die nachfolgende Tabelle beschreibt die Events, bei denen die SZM-Library automatisch aufgerufen wird. Bei den Events handelt es sich um Nutzeraktionen, welche aus technischen Gründen erhoben werden, jedoch nicht zur Zählung einer Mobile Impression führen. Diesem Event darf kein Code zugeordnet werden.

Event Type	Event	Bemerkung
ApplicationEnterBackground, ApplicationEnterForeground, ApplicationStart, ApplicationTerminate,	APP-spezifische Event	EnterFore/Background: APP geht in den Hintergrund Start/Terminate: APP wird gestartet/beendet
InternetConnectionEstablished InternetConnectionLost InternetConnectionSwitchedInterface	Art der Konnektivität ändert sich	Verbindung aufgebaut bzw. verloren Wechsel von Mobile auf Wifi bzw. umgekehrt
WebViewInit	Hybrid-Messung wird aktiviert	

HINWEIS: Die „automatisch durch die SZM-Library gemessenen Events“ werden erfasst, sobald die Session in der Methode *Application_Launching* der Klasse *App* gestartet wird.

4.3.2 PI-Events

Im folgenden sind Events aufgeführt, welche typischerweise zur Auslösung einer PI führen. Die Übernahme des Schemas wird empfohlen. Die Events müssen manuell ausgelöst werden (siehe 3.3.2 Logging eines Events), eine automatische Erhebung erfolgt nicht. PI-Events muss ein Code zugeordnet werden. Dieser Code kann anschließend den unterschiedlichen Kategorien zugeordnet werden und dient als Grundlage für die Bildung von Belegungseinheiten. Bei den PI-Events sind die Vorgaben zur Mobile Impression der IVW zu beachten.

Event Type	Event	Bemerkung
DeviceOrientation	Ausrichtung des Gerätes	Portrait, Landscape, PortraitUp, PortraitDown, Landscape-Left, LandscapeRight
GestureShake	Gerät wird geschüttelt	
ViewAppeared ViewRefreshed	Ein View (aka „Page“) wurde angezeigt oder mit neuen Daten aktualisiert	Beispiele: Appeared: initialer Aufruf einer Seite Refreshed: Suchfilter o. Aktualisierung von Daten
GameAction GameStarted	Gaming-Events	Aktion innerhalb eines Spiels Spiel gestartet
AudioPlay AudioPause AudioStop AudioNext AudioPrevious AudioReplay AudioSeekBack AudioSeekForward	Audio Playback	Wiedergabe, Pause, Stop, Nächster/vorheriger Titel, Vor/Zurückspulen, Wiederholung (Übergabe des Parameters Audio oder Video beim Event-Logging verpflichtend)
VideoPlay VideoPause VideoStop VideoNext VideoPrevious VideoReplay VideoSeekBack VideoSeekForward	Video Playback	Wiedergabe, Pause, Stop, Nächster/vorheriger Titel, Vor/Zurückspulen, Wiederholung (Übergabe des Parameters Audio oder Video beim Event-Logging verpflichtend)

4.3.3 Non-PI-Events

Im folgenden sind Events aufgeführt, welche typischerweise nicht zur Zählung einer PI führen. Sollten jedoch im Einzelfall Umstände vorliegen, unter denen auch hier eine Mobile PI erzeugt werden soll, können diese Events benutzt werden um dies zu ermöglichen. Bevor Sie diese Events zur Erzeugung von PIs benutzen, klären Sie den jeweiligen Sachverhalt bitte unmittelbar mit der IVW-Geschäftsstelle ab. Sollen diese Events zur Zählung von PIs führen, muss das auch hier manuell ausgelöst werden. Diesem Event muss dann ein Code zugeordnet werden. Dieser Code kann anschließend den unterschiedlichen Kategorien zugeordnet werden und dient als Grundlage für die Bildung von Belegungseinheiten.

Event Type	Event	Bemerkung
ViewDisappeared	Ein View (aka „Page“) wurde verlassen	Beispiele: Disappeared: Screen verlassen
OpenAppMaps OpenAppOther	Eine andere APP wird gestartet bzw. APP wird über eine URL verlassen	Maps: Die auf dem Device installierte Karten-APP (Google Maps etc.) wird aufgerufen Other: andere APPs bzw. URIs werden aufgerufen (Email, Phone, Websites, etc.)
DocumentOpen DocumentEdit DocumentClose	Dokument / Liste Bearbeitung	Liste editiert Dokument gespeichert
DataCancelled DataRefresh DataSucceeded DataFailed	Datenverbindung/ -verarbeitung	Datenverbindung abgebrochen Daten wurden aktualisiert Daten wurden erfolgreich übertragen Daten wurden nicht übertragen
DownloadCancelled DownloadStart DownloadSucceeded DownloadFailed	Download von Daten	Download wurde initiiert Download wurde abgebrochen Download erfolgreich beendet Download fehlgeschlagen
UploadCancelled UploadStart UploadSucceeded UploadFailed	Upload von Daten	Upload wurde initiiert Upload wurde abgebrochen Upload erfolgreich beendet Upload fehlgeschlagen
LoginSucceeded LoginFailed LoginLogout	Login	Login erfolgreich durchgeführt Login fehlgeschlagen Logout durchgeführt/Session beendet
HardwareButtonPushed	Schalter oder Knopf am Gerät gedrückt	Lautstärke über Schalterwippe am Gerät geändert Device gelocked (Power Taste betätigt) Druck auf Home-Button

GameFinished GameWon GameLost GameNewHighscore GameNewAchievement	Gaming-Events	Spiel beendet Spiel(runde) gewonnen Spiel(runde) verloren Neuer Highscore erreicht Neues Achievement erreicht
PushReceived	Push Notifications	Push innerhalb der APP empfangen (der Empfang von Push Nachrichte außerhalb der APP wird nicht gemessen)
IAPStarted IAPFinished IAPCancelled	In-APP Purchase / Kauf von digitalen Inhalten innerhalb einer APP	Kaufvorgang initiiert Kaufvorgang (erfolgreich) abgeschlossen Kaufvorgang abgebrochen

Sobald in der APP ein unter „manuell auszulösende Events“ beschriebenes Ereignis ausgelöst wird, ist die SZM-Library per LogEvent aufzurufen. Hierbei ist die in der Spalte „Event“ gelistete Konstanten als Parameter zu übergeben (siehe auch 3.3.2)

Die Übermittlung der Non-PI-Events an das SZM System ist optional. Die Non-PI-Events haben **keinen Einfluss auf die Reichweitenermittlung** Ihrer APP.

Diese Events **können** von Ihnen zur quantitativen Ermittlung der Häufigkeit des Auftretens dieser Events (in den INFOnline Auswertesystemen) verwendet werden. Hierbei ist zu beachten, dass die Erfassung und Übermittlung der Non-PI-Events technische Ressourcen (CPU-Zeit, Netzwerkverkehr, Batterie) auf dem Endgerät verwendet.

Mit Hinblick auf die Inanspruchnahme der technischen Ressourcen auf dem Endgerät bitten wir Sie, in der Planung der Umsetzung der Integration der Mess-Libs zu entscheiden, ob ihre APP NonPI-Events an das SZM System übermitteln soll.

4.3.4 Sonderfall Event „ViewRefreshed“

Für den Fall, das ein View aktualisiert wird (Event ViewRefreshed), ist folgendes zu beachten:

Das Event darf nur geloggt werden (bzw. die SZM-Library aufgerufen werden) wenn der Refresh der Daten manuell durch den Nutzer ausgelöst wurde. Bei einem automatischen Refresh darf das Event nicht geloggt werden.