**Integrating CA Mobile App Data into CA APM**

Contents

[CA Mobile Application Analytics and CA APM 2](#_Toc433364499)

[Architecture 3](#_Toc433364500)

[Available data 3](#_Toc433364501)

[Setup 5](#_Toc433364502)

[Scheduling the integration 8](#_Toc433364503)

This documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2015 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Engineering Services, CA Technologies 2015

# CA Mobile Application Analytics and CA APM

CA MAA allows monitoring of mobile applications, such as ones developed for Android and iOS platforms. Monitoring of mobile apps during the development, testing and production phases is imperative in order to manage quality and user experience.

The CA MAA solution consists of a client SDK component – either wrapped into a binary or attached into app source - which transmits performance, session, event and transaction data to the CA MAA server component. The data flows via http calls into the RESTful API of the server component. With the new 15.2.1 and forthcoming 15.3, also hybrid apps, meaning applications developed using javascript will be supported.

CA APM is an application performance management solution and mainly used for monitoring java and .net applications. CA APM (Introscope) deploys probes, or agents, into java virtual machines and .Net runtime components in order to gather response time data. These probes do not touch the source code.

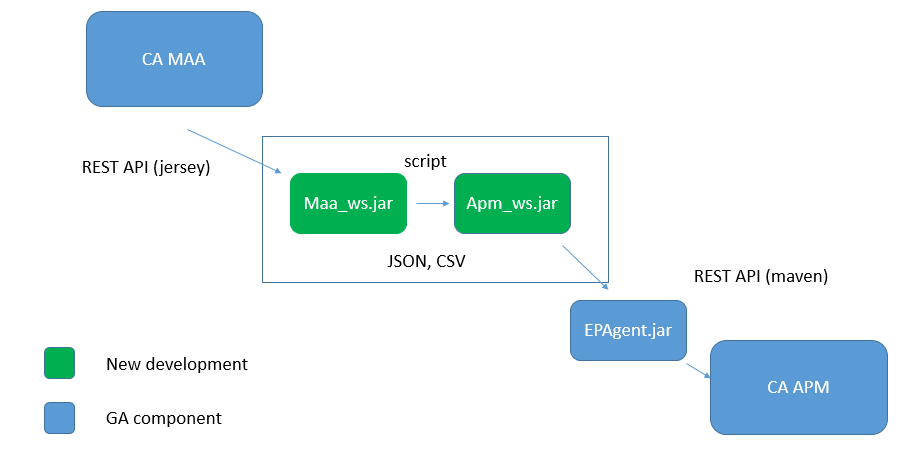
As corporates often develop both static and mobile applications, it would be beneficial to be able to monitor both from the same interface. CA MAA / APM currently offer an integration which pushes transactional data from the CA MAA client SDK directly into CA APM, but this header data is very limited. Currently there is no way of extending this integration.

This document outlines ta new way to mine data directly from the CA MAA RESTful interface in order to push it to the CA APM EPAgent automatically. This will make the integration between these two products more comprehensive, thus strengthening our mobile-to-mainframe message.

# Architecture

The solution consists of the following components

1. CA APM EPAgent
2. Maa\_ws.jar to pull data from CA MAA RESTful API
3. Apm\_ws.jar to push data to EPAgent -> CA APM
4. Maa\_apm.bat script to handle java calls and scheduling



*Image 2: flow of CA MAA data into CA APM via the new and existing components*

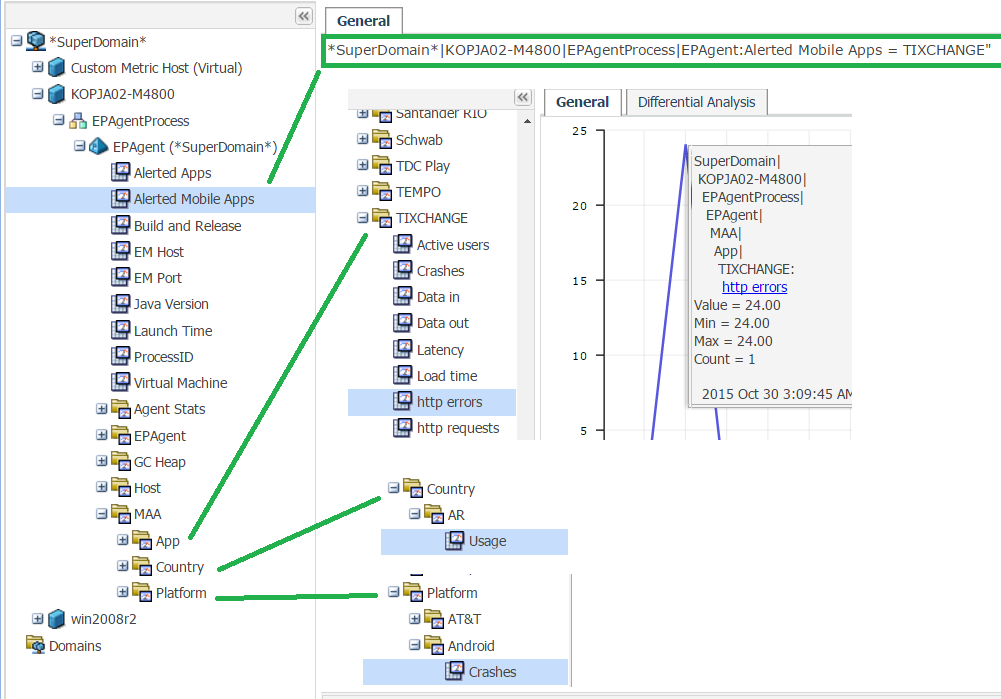
The new java components were created using Eclipse Kepler and have been tested with the following platforms and third party software

* JDK 1.7.0\_55
* CA MAA 15.1 and 15.2
* CA APM 10.1
* EPAgent 10.0.0.5
* Windows / Linux (Please note that at the time of writing, the integration script is only available for windows. It is, however, easy to convert into shell script format)

# Available data

The new integration currently supports the following types of CA MAA data

* Alerted Mobile Apps: StringEvent
  + Alert threshold set in CA MAA for app has been exceeded
* Application performance (MAA/Apps)
  + Active users
  + Crashes
  + Data in
  + Data out
  + Latency
  + Load time
  + http errors
  + http requests
* Country (MAA/Country)
  + Usage (sessions) per country
* Platform (MAA/Platform)
  + Number of crashes per app



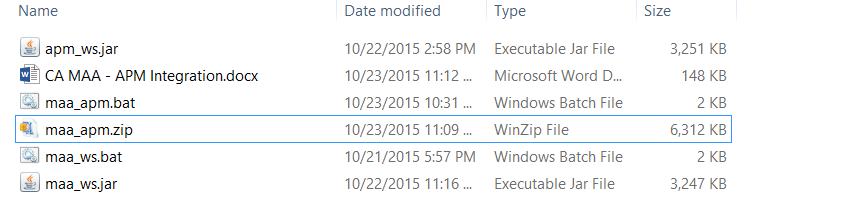
*Image 2: CA MAA Data available in CA APM after integration*

# Setup

1. Obtain network or internet access to both CA MAA server and CA APM WebViewer – make sure they are accessible from both the EPAgent client machine, and the machine where the integration mechanism is running. These can be the same machine
2. Download and install the CA APM EPAgent. Instructions on how to do this can be found at:

<https://communities.ca.com/docs/DOC-231150915>

1. Start the EPAgent – make sure no connection errors occur
2. Copy / paste the maa\_apm.zip onto the EPAgent or other available machine. Unzip into a folder



*Image 3: Contents of the new integration*

1. Open the maa\_apm.bat for editing
2. Modify the parameters to enable connection to both EPAgent and CA MAA server
3. Modify the MAA filter parameters to obtain the required data (Monthly, weekly or daily)

|  |
| --- |
| set EPAgentURL=130.119.30.141:8081  set APMdataType=IntCounter  set MAAURL="https://mdo.mobility.ca.com"  set MAAtenant="tenant organisation"  set MAAuser="username"  set MAApwd="password"  set MAAperiod="week"  set MAAstart="2015-10-01"  set MAAend="2015-10-23" |

Note that the date parameters must be in the above format. Data type IntCounter is typically used with CA MAA data because the data is collected over a minimum period of a day.

**The APMdataType –variable has been tested with following values:**

**IntAverage** (useful for response times, like “averagee time in seconds”; you don’t calculate the average yourself; just report all the applicable metrics (like in a loop) and the calculation will be performed automatically at the end of the interval)

**IntCounter** (useful for tally metrics, like “msgs in queue”, and does not change until a new value is reported)

**The following data types are also available for the EPAgent, but have not been tested with CA MAA data**

**PerIntervalCounter** (useful for rate metrics, like “miles per hour” or “errors per interval”; resets to zero at each new interval)

**LongCounter**  (same as above, but for very large numbers)

**LongAverage** (same as above, but for very large numbers

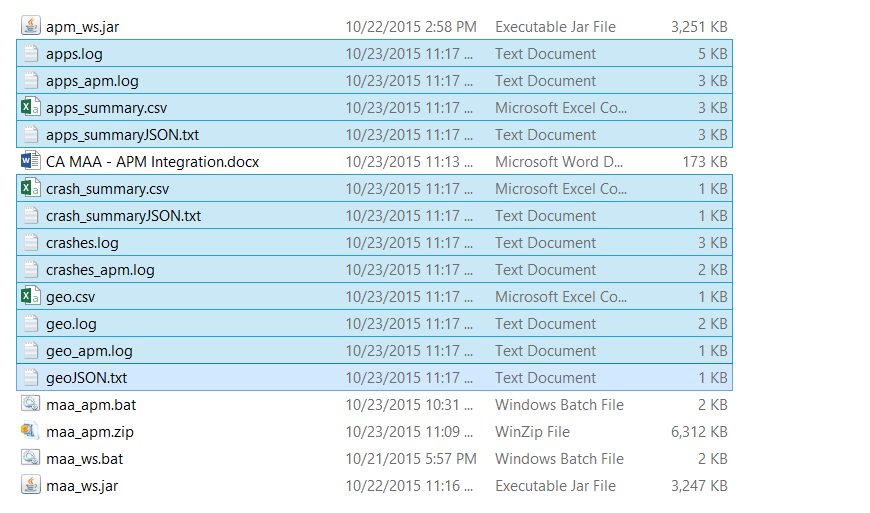
**The following data type is always used for Alerted Apps data. The variable APMdataType -setting has no effect on it.**

**StringEvent** (use to report string values, like “startup command-line” or a log entry). NOTE: StringEvents are not stored historically; only current values are used.

1. Modify the batch file by keeping or removing the data calls that are required for your particular integration. Currently the integration supports data for application performance, geographical user sessions and crash data by platform. Same jar files are used for each of these types. The types are distinguishable by the command line parameters like */mdo/v1/performance/apps\_summary* (maa\_ws.jar) or *appPerformance* (apm\_ws.jar)
2. Run the batch file by double-clicking on it, or by typing its name in command prompt

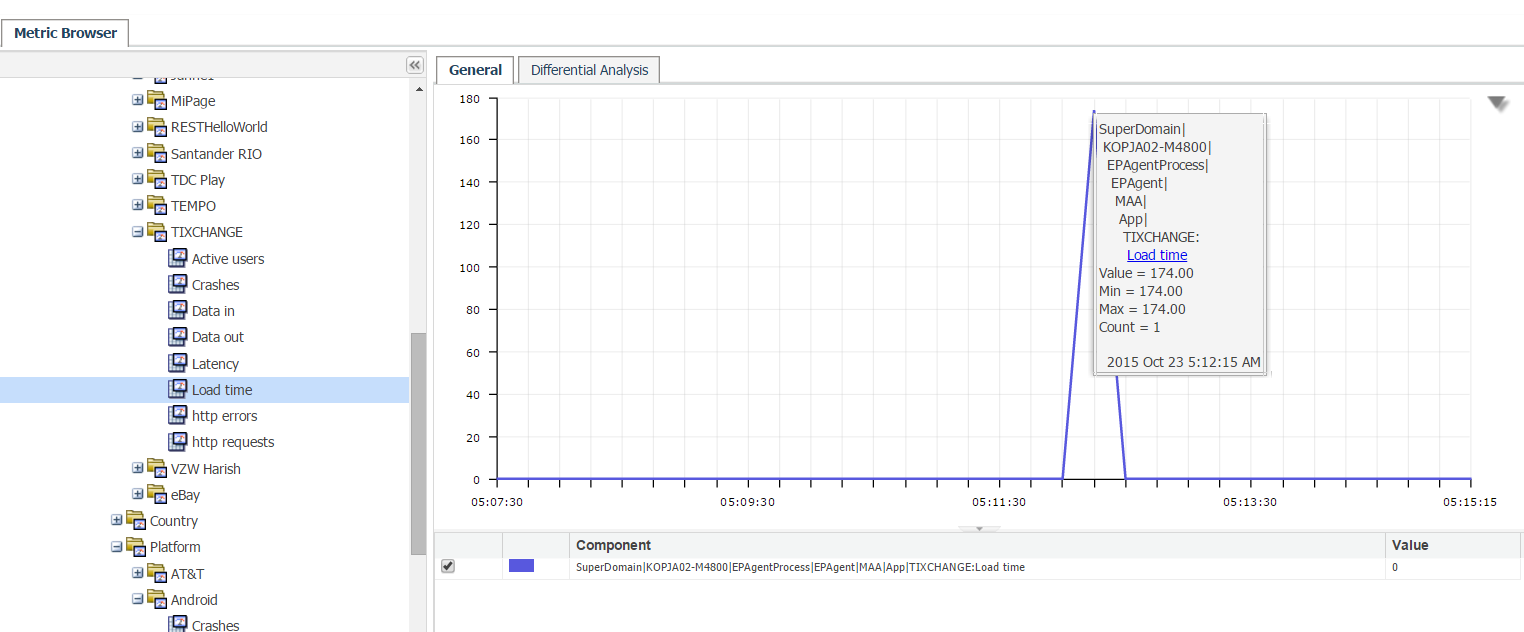
|  |
| --- |
| ================= Fetching app performance data from CA MAA ============  See apps.log for details  See geo.log for details  See crashes.log for details  See apps\_alerted.log for details  ================= Finished fetching data from CA MAA ===================  ================= Pushing data to APM EPAgent ========================  See apps\_apm.log for details  See geo\_apm.log for details  See geo\_apm.log for details  See apps\_alerted\_apm.log for details  ================= Finished pushing data to APM EPAgent ================== |

Any output and errors will be piped into the above log files. For more information on the operations performed, please refer to the output and log files generated by the new executables



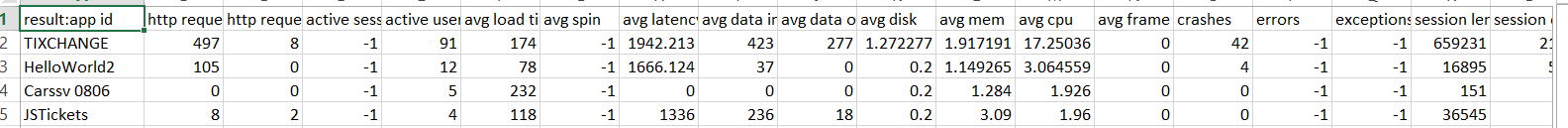
*Image 4: Output and log files created by the new integration mechanism*

Result of the operation is that CA MAA data is visible in the CA APM WebViewer



*Image 5: CA MAA data in CA APM WebViewer*

Output from CA MAA can also be viewed using the CSV files generated by the maa\_ws.jar. This utility as standalone can also be used to extract more CA MAA for other integration purposes



*Image 6: CA MAA data in apps\_summary.csv generated by maa\_ws.jar*

# Scheduling the integration

As with any integration, user would expect automation. This is the reason why the invoking of the various \*.jar calls is done in a command line script. These scripts, both on windows and linux, can be scheduled to run using windows scheduler or linux cron jobs. In this case, if required, the date parameters in the invoking script should be adjusted accordingly by using date or current date functions.