BAPCo®

TabletMark®

An Overview of TabletMark Version 3

May 2015

Revision 1.0

Revision History:

1.0 (May 2015): first release

About BAPCo

Business Applications Performance Corporation (BAPCo®) is a non-profit consortium with a charter to develop and distribute a set of objective performance benchmarks for personal computers and computing devices based on popular usage models and industry standard operating systems.

For more information about BAPCo and a complete list of the current membership, see our website at http://www.bapco.com.

Table of Contents

1 Introduction

TabletMark® V3 is the latest version of the premier performance and battery life benchmark for tablet computers, featuring workloads that were created by experts to represent real world apps and usage models, and now supporting all major tablet operating systems, including Android 4.3+, iOS 7.1.2+ and Windows 8.1.

TabletMark gives commercial and government IT decision makers, media, channel buyers, consultants, consumers, and system and component designers and manufacturers an objective, easy-to-use tool to evaluate the performance and battery life of tablets across the wide range of activities that a tablet user may encounter.

TabletMark is designed for those who want to:

- Evaluate and compare tablet computers for purchase consideration based on system performance and application responsiveness.
- Provide useful information to their audiences to assist in the evaluation and purchase of tablet computers.
- Evaluate tablet computers to better optimize the battery life and performance of the system.

Unlike benchmarks that only measure battery life, TabletMark measures battery life and performance simultaneously, showing how well a system design addresses the inherent tradeoffs between performance and power management.

Unlike synthetic benchmarks, which artificially drive components to peak capacity or attempt to deduce performance using a static simulation of application behavior, TabletMark measures platform-level user experience, using the same platform APIs and services used by third-party app developers, real user workloads and real data sets to accurately measure how overall system performance impacts real-world user experience.

TabletMark builds upon BAPCo's 24-year history of building benchmarks to evaluate platform technologies. Benchmarks designed by BAPCo are the result of cooperative development among companies representing the breadth of the computing industry. They harness a consortium of knowledge to better reflect today's and tomorrow's emerging business trends.

This document describes the methodologies employed in the development of TabletMark. For detailed instructions on how to install and run TabletMark, please refer to the documentation provided in the app or on the BAPCo web site (www.bapco.com).

TabletMark is available free from Google Play, the iTunes App Store and the Windows Store. Just search for "TabletMark" on your tablet device, or follow these links:

- **Google Play** (for Android devices): https://play.google.com/store/apps/details?id=com.bapco.tabletmark
- **iTunes App Store** (for iPad): https://itunes.apple.com/us/app/tabletmark/id925058546
- **Windows Store** (for Windows devices): http://apps.microsoft.com/windows/en-us/app/tabletmark/7b9b0c60-8654-4c3c-880f-99f38d4d31bd

Alternative installation methods are explained on the BAPCo web site:

- http://bapco.com/products/tabletmark

2 BAPCo Development Process

BAPCo creates benchmarks in accordance with the BAPCo Development Process, a set of milestones and checkpoints collaboratively developed and agreed upon by the BAPCo membership.

Early in the process, prevailing business tablet computer usage models are identified and grouped into scenarios according to their fit within a workflow. Applications are selected for each usage model on the basis of market research and technical feasibility.

BAPCo members then join together with expert application users in development sessions to collaboratively develop a workload specification for each scenario, defining each user/computer interaction which is to be simulated by the benchmark.

The goal of the development sessions is to produce representative business application workloads for the benchmark. Each application workload consists of three elements: the input data set, the tasks performed on the input data set, and the generated output. An example of generated output would be an image generated through an iterative process of steps to create a desired appearance. These three elements of the workload are chosen to represent the workflow of a user skilled in each given application.

After the workload specifications are created at the development sessions, BAPCo developers implement the workloads according to those specifications while satisfying benchmarking constraints to ensure the stability of the benchmark, the consistency of results, and the feasibility of implementation and distribution of the benchmark.

2.1 Milestone Overview

The BAPCo development process is divided into six major phases (Initialization, Design and Planning, Implementation, Validation, Characterization and Launch). Each phase consists of a series of milestones, some of which may be worked on concurrently.

The membership must vote to close each milestone. Once all the milestones within a phase are complete, the membership must vote to exit the current phase and enter the next phase. BAPCo members work in a collaborative process where decisions regarding products are sometimes made by majority vote rather than unanimously.

The following is the list of the development phases and the corresponding milestones. Some of these milestones are explained in greater detail in the following sections, as noted in this list.

1. Initialization Phase
    a. Milestone 0 – Committee kickoff
    b. Milestone 1 – Benchmark market and customer analysis
    c. Milestone 2 – Product positioning and customer value proposition
    d. Milestone 3 – Preliminary marketing requirements document
    e. Milestone 4 – Final marketing requirements document
2. Design and Planning Phase
    a. Milestone 5 – Preliminary engineering requirements document
    b. Milestone 6 – Usage model selection (see section 2.2)
    c. Milestone 7 – Application selection (see section 2.3)
    d. Milestone 8 – Define member resource commitments
    e. Milestone 9 – Define development infrastructure
    f. Milestone 10 – Define scoring methodology (see section 2.4)
    g. Milestone 11 – Define application/API licensing requirements
    h. Milestone 12 – Plan and execute workload development sessions (see section 2.5, 2.6)
    i. Milestone 13 – Define product release criteria
    j. Milestone 14 – Select product name
3. Implementation Phase
    a. Milestone 15 – Create implementation schedule

   b. Milestone 16 – Determine calibration system (see section 2.7)
   c. Milestone 17 – Software development (see section 2.8)

4. Validation Phase
   a. Milestone 18 – Validation testing
   b. Milestone 19 – Define risk management plan
5. Characterization Phase
   a. Milestone 20 – Characterization testing (see section 2.9)
   b. Milestone 21 – Beta testing
6. Launch Phase
   a. Milestone 22 – Product pricing
   b. Milestone 23 – Pre-launch materials
   c. Milestone 24 – Release to manufacture vote and sign-off
   d. Milestone 25 – Duplicate and distribute media
   e. Milestone 26 – Post-launch materials

2.2 Usage Model/Scenario Selection

In Milestone 6 of the BAPCo Development Process, usage models are chosen for inclusion in a benchmark and related usage models are grouped into scenarios.

For TabletMark, BAPCo used market studies on tablet usage patterns to choose a wide variety of usage models relevant to tablet users.

Those usage models were then grouped into the following scenario groups. For more detailed descriptions of each scenario, please see section 2.6.

Web and Email Scenario
The Web and Email scenario models productivity and online activities, such as browsing professional and consumer web pages, reading and creating emails, and viewing notes and to-do lists on a tablet computer. The web browsing portion of the scenario uses a variety of web pages based on the most popular sites for professional networks, social media, photo sharing service, news/finance/sports news outlets, and online shopping. The email portion of the scenario includes compression/decompression and encryption/decryption operations to reflect common information security practices.

Photo and Video Sharing Scenario
The Photo and Video Sharing scenario models media creation and consumption activities, such as viewing and manipulating photos, setting up photo slideshows, playing back photo slideshows, performing High-Dynamic-Range (HDR) imaging operations, stitching existing video clips into new movies, and playing back these newly created movies.

Video Playback
The Video Playback scenario consists of playing high definition video from local storage. This scenario contains 9 minutes of 1080p video content, encoded in H.264 with an average bitrate of 8 Mbps at 29.97 fps. There is no performance rating for the Video Playback scenario.

2.3 Application Selection

In Milestone 7, after the usage models have been collected into scenarios, applications are chosen for the scenarios on the basis of market research and technical feasibility.

For TabletMark, BAPCo reviewed market-leading applications for Android, iOS and Windows for each chosen usage model. Then, in order to accurately reflect real world application behavior within the restrictions of modern tablet runtime environments, BAPCo created new applications for these usage models from scratch, taking inspiration from the design and behavior of the market leaders.

These “application models” were created separately for Android, iOS and Windows, built from the ground up using the same development tools, compilers, platform APIs and programming techniques an app developer would use for each platform.

Because these application models were purpose-built for cross-OS benchmarking, they have comparable user interfaces, features and output across all supported platforms. And because they rely upon the high-level functionality exposed by each platform, they take advantage of the efficiencies and optimizations intrinsic to each platform. See section 2.8.2 for more information.

2.4 Scoring Methodology

In Milestone 10, BAPCo decides the types of results that will be produced by a benchmark and the scoring methodology that determines how those results are calculated.

Importantly, BAPCo determines the scoring methodology before determining the content of the workloads, which helps ensure that a methodology is chosen for its ability to generate results that correspond to user experience, not for the results it produces on a pre-determined set of workloads.

For TabletMark, BAPCo evaluated the merits of a variety of scoring methodologies and chose a methodology on the basis of how it met the following criteria:

- The scoring methodology should give expected results:
    - The resulting score should differentiate between systems with different performance.
    - The resulting score should be repeatable and not have high variation.
    - The resulting score should not be affected by benchmark artifacts, such as the number of tasks within a scenario.
- The relative performance between any two systems should not be affected by the selection of the calibration system.
- The scoring methodology should be easy to understand.

2.4.1 Performance Rating

TabletMark measures system performance by measuring the response time of tasks on a tablet computer using representative application models, user inputs and workloads. In the TabletMark scoring methodology, task response times are used to generate performance ratings that reflect the user experience. The faster a tablet computer responds to the application workloads in TabletMark, the higher its TabletMark performance ratings will be. For more information on how task response times are measured, see section 2.8.

The performance rating is given in relative terms, with respect to a calibration system. It is calibrated in such a way that a tablet computer with performance equivalent to this calibration system for a given scenario will have a performance qualification rating of 1,000. A system twice as fast as the calibration system on a given scenario (or, equivalently, that responds in half the time on average) will have a performance qualification rating of 2,000.

A complete TabletMark run will output two scenario performance ratings (one for Web and Email and the other for Photo and Video Sharing), plus an overall performance rating.

Scenarios can also be run selectively in any combination, in which case TabletMark will output performance ratings only for the scenarios that were run. In order to produce an overall performance rating, both the “Web and Email” and “Photo and Video Sharing” scenarios must be run at a minimum.

Scenario Rating
Each scenario has a rating calculated by taking the sum of the response times of tasks in that scenario as performed on the test system and then comparing it with the sum of those same task response times as performed on the calibration system (see section 2.7). The calibration sum is divided by the measured sum on the test system and multiplied by 1,000. The result is then rounded to the nearest integer.

Overall Rating
The TabletMark Overall Rating is calculated by taking the geometric mean of all the scenario ratings (prior to rounding). The result is then rounded to the nearest integer. To obtain an overall rating, you must run all performance scenarios.

A performance qualification rating p_i is first calculated for each successfully completed iteration of a scenario (excluding any conditioning iteration and partial iterations):

$$p_i = 1000 \left(\frac{s_c}{s_t} \right)$$

Where:
s_c = the sum of the response times of tasks in that iteration as performed on the calibration system, explained below.

s_t = the sum of the response times of tasks in that iteration as performed on the test system.

The calibration sum s_c is a fixed value BAPCo calculates by performing three runs of the scenario—each including a conditioning run—on a calibration system (see section 2.7). For each successfully completed iteration among all three runs (excluding any conditioning iterations and partial iterations), the sum is taken of the response times of all the tasks in that iteration. s_c is the arithmetic mean of all those iteration sums.

p_i = rounded to the nearest 1/100th. For display purposes only, this is further rounded to the nearest integer.

Then the performance qualification rating p_s is calculated by taking the arithmetic mean of the performance qualification ratings pi for all successfully completed scenario iterations $\mathrm{i} = 1, \dots, \mathrm{n}$:

$$p_s = \frac{1}{n} \sum_{i=1}^{n} p_i$$

p_s is rounded to the nearest integer.

2.4.2 Battery Life Rating

The battery life rating is calculated by measuring the actual battery duration observed (from a fully charged state to a fully depleted state), in whole minutes, while running all three scenarios in succession, interspersed with idle periods to simulate user idle time (as described in section 2.9.1), then repeating until the battery dies.

A battery life rating is only produced if the user selects the “Battery Rundown” option when starting the test, which will in turn select all scenarios for the test. TabletMark can only produce battery life results for tests in which all scenarios are run.

2.4.3 Comparing TabletMark V3 Ratings

One of the key features of TabletMark V3 is the comparability of its performance and battery life ratings across different operating systems. The workloads in TabletMark V3 were designed to accomplish equivalent work across all supported platform OSes (see section 2.8), and all versions of TabletMark V3 uses the same calibration sum for scoring (see section 2.7).

TabletMark V3 ratings are not directly comparable to results obtained from previous versions of TabletMark.

2.5 Workload Development Sessions

Once the usage models, scenarios, application models, and scoring methodology for the benchmark are decided, BAPCo members and application experts meet to create the workloads that will be used in the benchmark.

For TabletMark, the workload development sessions consisted of two sets of face-to-face meetings that included representatives from BAPCo member companies and expert application users who had professional experience with the applications chosen for the benchmark. The application experts included professionals in the fields of small business marketing, financial forecasting, graphic design, video editing, web development, and enterprise IT deployment.

In the workload development sessions, the experts take the lead, weaving the usage models supplied by BAPCo into a storyboard of user interactions with a series of application models. Each user/tablet interaction is written down in a workload specification, which is later used to automate the workloads.

At the end of the workload development sessions, BAPCo comes away with a detailed workload specification for each of the benchmark scenarios and all of the input data sets needed to reproduce the workloads created at the sessions.

Additional Workload Considerations

The following additional factors were considered at the workload development sessions:

Input Data Set

Frequently in the sessions, the experts need raw digital content to serve as input data set for a workload. Examples of such content might include a video to transcode, an email to modify, or photos to manipulate. When experts need such content, care is taken to ensure that they use something that is functionally representative of content they might use or encounter professionally.

For instance, if pictures are needed in order to create a photo slideshow, an expert might walk outside and take pictures using the same equipment he/she uses professionally. If a song track is needed as the background music for creating a movie, an expert might purchase a stock track from his/her usual online resource. Like the user interactions, all of these source materials are captured at the development session and used later in the development of automated workloads.

2.6 Scenario Workload Descriptions

The scenario workloads created at the workload development sessions for TabletMark are described below:

Web and Email Scenario

Web
The web workload models a user browsing a variety of professional and personal websites, scrolling through web pages, pausing at times to read page contents, viewing video clips and animated advertisements embedded into the pages. These websites are based on popular websites for professional networks, social media, photo sharing services, news outlets, finance and sports sites, and online retailers.

Email
The email workload models a user opening an email client, looking at a list of email messages, opening messages, pausing to read messages, and composing new emails.

Notes
The notes workload models a user opening a list of notes, including to-do and shopping lists, browsing through them and updating one of them.

Photo and Video Sharing Scenario

Photo
The photo workload models a user loading a set of high-resolution pictures, adjusting them for optimal exposure, brightness, color saturation, rotation, and crop. The user then creates a slideshow using this set of pictures and plays the slideshow. Finally, a series of high dynamic range (HDR) imaging operations are performed on multiple sets of exposure-bracketed photos. The resulting HDR images show clear details across the entire brightness spectrum, from the lightest to the darkest areas.

Video

The video workload models a user starting a video editing application and loading a set of existing video clips. Some of the clips are selected, trimmed, and merged into a new video, adding a new background music track in the process. The resulting movie file is then transcoded and played back in the application. The user proceeds to select a different set of video clips, applies video effects such as grayscale, and creates a second video timeline. This new movie is then transcoded and played back in the application.

Video Playback Scenario

Video Playback

The video playback scenario models a user watching nine minutes of 1080p high definition video. The video formats used are representative of what can be found on popular video sharing sites.

2.7 Calibration System

The calibration system is a system chosen in Milestone 16 as a reference point for all other TabletMark results. BAPCo chose the configuration below for its wide availability and its representation of a typical mainstream tablet computer at the time of release of TabletMark.

TabletMark V3 has been calibrated in such a way that a tablet computer with performance equivalent to this calibration system for a given workload will have a performance rating of 1,000. A system twice as fast as the calibration system on a given workload (or, equivalently, that responds in half the time on average) will have a performance rating of 2,000. This holds for both overall ratings and scenario ratings.

The calibration system for TabletMark has the following configuration:

- **Tablet Model:** Lenovo* ThinkPad Tablet 10*
- **CPU:** Intel* Atom* CPU Z3795 (1.60GHz, 4 Cores/4 Threads, 2MB L2 cache)
- **Operating System:** Microsoft* Windows 8.1* (32-bit)
- **Operating System Language:** US English
- **System Memory:** 2GB LPDDR3 SDRAM
- **Storage:** eMMC4.51 64GB Solid-State Drive
- **Display:** 10.1" @ 1920x1200

A fresh operating system installation is performed on the system.

This calibration system serves as the reference point for the performance ratings produced by all versions of TabletMark V3, including the Android, iOS and Windows releases.

For more details about the configuration of the calibration system, please contact support@bapco.com.

Please note that the minimum requirements to run TabletMark are listed in Appendix B and are not equivalent to the calibration system.

2.8 Benchmark Implementation

Once the workload specifications have been created, BAPCo begins the work of translating workload specifications into an automated benchmark in Milestone 17.

2.8.1 Benchmark Framework

TabletMark is built upon simulated user behaviors and interactions, using controls like buttons, text input boxes, and menus to navigate applications. (See Appendix C for screenshots of the benchmark in action.)

To ensure that TabletMark has deterministic behavior, BAPCo uses a framework to collect system information, run workloads within a series of application models for each scenario, record performance measurements, calculate performance ratings, and display test results. The framework is kept lightweight, consuming a minimal amount of memory and compute resources, in order to ensure that performance measurements reflect the workload performance and do not include overhead from the framework.

The fundamental performance unit in TabletMark is response time. Response time is defined as the time it takes the tablet computer to complete a task that has been initiated. A task can be initiated by a mouse click or a keystroke. The duration of each task is measured by the framework. Examples of tasks include loading documents, finding text in a document, performing an image manipulation, and transcoding a video.

The framework measures the duration time for each task. Simulated pauses mimicking user behaviors (e.g., reading a web page or email) is excluded from the measurement.

2.8.2 Cross-Platform Development Methodology

As explained in section 2.3, application models and workloads in TabletMark were implemented from the ground-up using standard development tools and practices for each target platform. They make full use of the APIs provided by each platform.

In order to ensure results remain comparable across these disparate platforms, the workloads were developed in a way that ensures all platforms:

- perform comparable work (e.g., using similar video codecs and bitrates, using the same encryption algorithm and security level),
- operate on comparable input data (e.g., similar file size, equivalent perceived image/video/audio quality),
- produce comparable output data (e.g., similar file size, equivalent perceived image/video/audio quality).

To illustrate with an example, the Web and Email scenario includes email encryption/decryption operations. To accomplish this in a representative fashion on each platform, the following libraries were used:

- **Android:** javax.crypto.Cipher
- **iOS:** CCCrypt
- **Windows:** Windows.Security.Cryptography

To ensure comparability of results, the workload utilizes the same 128-bit AES-CBC encryption on all platforms.

For another example, the Photo and Video Sharing scenario includes video encoding operations that utilize the following libraries:

- **Android:** OpenMAX
- **iOS:** AV Foundation framework
- **Windows:** DirectX Video Acceleration (DXVA)

Though the library varies from one platform to the next, the codec is consistent across all platforms:

- H.264/AVC codec
- 1920x1080 resolution
- 30 progressive frames per second
- 8 Mb/sec
- 10 second I-frame interval

This methodology was chosen to balance the desire to reflect real-world platform performance while ensuring the platforms are all being asked to perform work that is equivalent from the user's perspective.

2.8.3 Development Tools

The information in this section reflects current information as of TabletMark v3.0.0.69 and v3.0.0.73, but is subject to change in future releases to support future devices and development tools. Please contact support@bapco.com with any questions.

Generally, the workloads were built using the following software development tools:

Development Tools			
Platform	Language	Compiler	Notes
Android	Java	default included with Android SDK 21 + OpenJDK 1.7.0_55	Compiler options: *default*
iOS	Objective C	default included with Xcode 6.0.1	Compiler options: *default* *(-Os and Automatic Reference Counting)*
Windows	C#	default included with Microsoft Visual Studio 2013	Compiler options: *default*

Table 2.8.1: Development Tools

If a usage model called for functionality not available in platform APIs, the functionality was implemented using native C code, which was shared across all platforms. Only the HDR workload in the Photo and Video Sharing scenario was implemented in this way.

This workload was compiled using the following tools and settings:

Development Tools for Native C Code		
Platform	Compiler	Notes
Android	default included with Android NDK r10c	Compiler options: *-O3 (enables many general optimizations)* *--ftree-vectorize (implied by -O3)* *--ffast-math (enables common math optimizations for code that doesn't require strict IEEE compliance)* *--fomit-frame-pointer (implied by -O3, reduces memory consumption to support lower RAM—e.g., 1 GB—devices)*
iOS	default included with Xcode 6.0.1	Compiler options: *default* *(-Os and Automatic Reference Counting)*
Windows	default included with Microsoft Visual Studio 2013	Compiler options: *default (/O2) + /Oi (generate intrinsic functions)*

Table 2.8.2: Development Tools for Native C Code

The workloads are compiled for the following platform/architecture combinations (using the nomenclature of each platform's build tools):

Architecture Targets	
Platform	Notes
Android	armeabi, armeabi-v7a, arm64-v8a, x86, x86_64, mips *(packaged together in one .apk)*
iOS	armv7, armv7s, arm64 *(packaged together in one .ipa)*
Windows	x86, x64, ARM *(packaged separately)*

Table 2.8.3: Workload Architecture Targets by Platform

2.8.4 Output Validation

As an additional check that the scenario workloads ran correctly and produced the desired output, TabletMark checks various attributes of output files produced by the workloads to ensure they fall within acceptable bounds. These checks are performed each iteration.

2.9 Workload Characterization

Once the scenario workloads are implemented and validated against the workload specifications created at the development sessions, BAPCo members then run the benchmark on a wide variety of systems to ensure that the benchmark produces results that are valid, representative, and reproducible.

During this process, BAPCo members share data, raise concerns, and suggest workload changes. Any workload change requires a majority vote of the committee.

TabletMark is a tool for measuring both tablet performance and battery life. It is important that its workloads and power profiles of its activities are reasonably representative of user experiences and expectations. BAPCo members work together to arrive at an estimation of a representative level of user activity, but acknowledge that individual user experience could vary from the results reported by the benchmark.

The battery life reported by TabletMark is an approximation of the battery life a user who performs that same workloads would expect from the same system.

2.9.1 User Idle Time

One way BAPCo members adjust the power profile of the workload is by determining an appropriate proportion of user idle time relative to active time.

Users of tablets often leave their systems idle for a period of time between sessions of active use. When measuring battery life, TabletMark simulates this behavior by interspersing occasional periods of user inactivity throughout the workload. These idle periods are only relevant for battery life tests. In order to minimize test runtime, the idle periods are bypassed for tests that produce only performance results.

Inclusion of these idle periods for battery life tests better models real-world mobile usage and allows the hardware and software power management features of the system under test to behave in a realistic manner.

Furthermore, the scenario workloads themselves are paced at a human speed, pausing briefly between individual tasks to simulate the user reading, inspecting the output of a previous task, or considering the next task.

TabletMark measures battery life by requiring that the tablet battery be fully charged (at least 97%) prior to testing, and then performing the following activities in a loop (with each loop referred to as an "iteration") until the battery dies:

1. Web and Email scenario (duration varies depending upon tablet performance)
2. User Idle period (duration is fixed, 3 minutes)
3. Photo and Video Sharing scenario (duration varies depending upon tablet performance)
4. User Idle period (duration is fixed, 3 minutes)
5. Video Playback scenario (duration is fixed, 9 minutes)
6. User Idle period (duration varies depending upon tablet performance, extends iteration to 50 minutes total)

Different systems may take different amounts of time complete the Web and Email scenario (activity 1) and the Photo and Video Sharing scenario (activity 3). To compensate for this, TabletMark waits at the end of the iteration (activity 6) to ensure the iteration has a fixed duration of 50 minutes. This ensures that all systems have done the same amount of work after each 50 minutes of battery life testing.

BAPCo sampled a variety of systems that met the TabletMark V3 minimum system requirements and found they all finished an iteration of the workload within 50 minutes. In the hypothetical case that a device does not finish an iteration within 50 minutes, the benchmark will allow the iteration to run to completion (past the 50 minute mark) and then omit the final idle period (activity 6) for that iteration, immediately beginning the next iteration.

During the idle periods, TabletMark shows a solid colored screen and prevents the device from entering any power state that might shut down the display or reduce its brightness (e.g., display dimming/standby, system standby/hibernation). The heads-up display showing test status remains on the screen.

3 Workload Characteristics

This section provides data illustrating the performance characteristics of TabletMark.

3.1 Sensitivity Analysis

The series of tables below shows the sensitivity of TabletMark to different system characteristics, including the amount of system memory (RAM), number of CPU cores, type of storage device, and display resolution.

Within each study only one system component (e.g., memory) is varied. All the other system components are held constant. To best illustrate the sensitivity, the minimal configuration is chosen as a baseline for each table and the ratings for the other configurations are shown as the percentage difference relative to that baseline.

A high-end Windows-based desktop system has been chosen as the basis for these studies for two reasons:

- Using a Windows-based desktop configuration offers the most flexibility in component selection and configurability, allowing us to measure many vectors of component performance by testing a wide variety of components and configurations.
- Using a high-end system with high-end baseline components allows us to focus on one component at a time (the independent variables) without the performance sensitivity being heavily limited by any of the baseline components (the control variables).

Note that component sensitivities will vary from one configuration to the next.

System configuration (unless otherwise specified):

- **Motherboard:** ASUS X79-DELUXE
- **Motherboard Firmware:** 0605 (Dec 20, 2013)
- **CPU:** Intel Core i7-4960X (3.6 GHz, 6 cores, with Turbo Boost and Hyper-Threading disabled in UEFI settings to facilitate frequency and core sensitivity tests)
- **Storage:** Intel SSD 530 SSDSC2BW240A4 (240 GB, 2.5”)
- **Graphics:** ASUS Radeon R9 290X w/4 GB GDDR5
- **Memory:** 8 GB DDR3-1333 RAM (2 x 4 GB DIMMs)
- **OS:** Windows 8.1 Update 2 64-bit
- **OS Language:** US English
- **Resolution:** 1920 x 1080

For the tables below, the following components were substituted into the system described above as indicated in the chart headings:

- **Memory**
  - 1 GB DDR3-1333 RAM (2 x 1 GB DIMM, software limited to 1 GB using msconfig)
  - 2 GB DDR3-1333 RAM (2 x 1 GB DIMM)
  - 4 GB DDR3-1333 RAM (2 x 2 GB DIMMs)
  - 8 GB DDR3-1333 RAM (2 x 4 GB DIMMs)
  - 16 GB DDR3-1333 RAM (2 x 8 GB DIMMs)
- **CPU**
  - Number of cores adjusted in msconfig (1, 2, 4, 6 cores), with Hyper-Threading and Turbo Boost disabled
  - Frequency adjusted in UEFI settings (2.0, 2.5, 3.0 GHz), with Hyper-Threading and Turbo Boost disabled
- **Storage**
  - Seagate Momentus Thin ST500LT012 (500 GB, 2.5”, 5400 RPM)
  - Intel SSD 530 SSDSC2BW240A4 (240 GB, 2.5”)
- **Graphics**
  - ASUS Radeon R7 240 w/2 GB DDR3
  - ASUS Radeon R7 250 w/1 GB GDDR5
  - ASUS Radeon R9 270 w/2 GB GDDR5
  - ASUS Radeon R9 290X w/4 GB GDDR5
- **Resolution**
  - 1366 x 768
  - 1920 x 1080
  - 2560 x 1440
- **Operating System**
  - Windows 8.1 Update 2 32-bit
  - Windows 8.1 Update 2 64-bit

3.1.1 Sensitivity to System Memory

The total system memory is varied from 1 GB up to 16 GB using DIMM combinations and software settings as described in section 3.1.

Performance vs. 1 GB					
System Memory	1 GB (2 x 1 GB*)	2 GB (2 x 1 GB)	4 GB (2 x 2 GB)	8 GB (2 x 4 GB)	16 GB (2 x 8 GB)
Web and Email Scenario	Baseline	+6.5%	+9.9%	+10.1%	+10.2%
Photo and Video Scenario	Baseline	+1.2%	+1.4%	+1.9%	+1.3%
TabletMark Overall Rating	Baseline	+3.8%	+5.6%	+5.9%	+5.7%

Table 3.1.1: Performance Sensitivity to System Memory

**1 GB achieved in software, see memory configuration notes in section 3.1*

3.1.2 Sensitivity to CPU Cores

The total number of CPU cores available to the system was varied from 1 core up to 6 cores. This was achieved by changing the number of processors available to the operating system using the msconfig utility.

Performance vs. 1 Core				
CPU Cores	1 Core	2 Cores	4 Cores	6 Cores
Web and Email Scenario	Baseline	+2.6%	+3.9%	+4.7%
Photo and Video Scenario	Baseline	+1.0%	+1.1%	+1.4%
TabletMark Overall Rating	Baseline	+1.8%	+2.5%	+3.0%

Table 3.1.2: Performance Sensitivity to CPU Cores

3.1.3 Sensitivity to CPU Frequency

The CPU clock frequency was varied from 2.0 GHz up to 3.0 GHz. This was achieved by changing base clock frequency and core ratio in the motherboard UEFI settings.

Performance vs. 2.0 GHz			
CPU Frequency	2.0 GHz	2.5 GHz	3.0 GHz
Web and Email Scenario	Baseline	+24.0%	+43.3%
Photo and Video Scenario	Baseline	+7.2%	+13.8%
TabletMark Overall Rating	Baseline	+15.3%	+27.7%

Table 3.1.3: Performance Sensitivity to CPU Frequency

3.1.4 Sensitivity to Storage Subsystem

The primary storage device was changed from a laptop hard drive to a solid state device (SSD).

Performance vs. 5400 RPM HDD		
Storage Subsystem	500 GB 5400 RPM HDD	240 GB SSD
Web and Email Scenario	Baseline	+0.8%
Photo and Video Scenario	Baseline	+6.2%
TabletMark Overall Rating	Baseline	+3.5%

Table 3.1.4: Performance Sensitivity to Storage Subsystem

3.1.5 Sensitivity to Graphics Subsystem

The video card was varied from an ATI R7-240 up to an ATI R9-290X.

Performance vs. ATI R7-240				
Graphics Subsystem	ATI R7-240 (2 GB)	ATI R7-250 (1 GB)	ATI R9-270 (2 GB)	ATI R9-290X (4 GB)
Web and Email Scenario	Baseline	+0.1%	+1.1%	+0.8%
Photo and Video Scenario	Baseline	+2.2%	+4.3%	+4.1%
TabletMark Overall Rating	Baseline	+1.1%	+2.6%	+2.4%

Table 3.1.5: Performance Sensitivity to Graphics Subsystem

3.1.6 Sensitivity to Display Resolution

The system display resolution was varied from 1366x768 up to 2560x1440.

Performance vs. 1366x768			
Display Resolution	1366x768	1920x1080	2560x1440
Web and Email Scenario	Baseline	-0.2%	+0.4%
Photo and Video Scenario	Baseline	+0.2%	-0.5%
TabletMark Overall Rating	Baseline	+0.0%	+0.0%

Table 3.1.6: Performance Sensitivity to Display Resolution

3.1.7 Sensitivity to Operating System

The operating system was changed from a 32-bit version to a 64-bit version.

Performance vs. 32-bit		
Operating System	Windows 8.1 32-bit	Windows 8.1 64-bit
Web and Email Scenario	Baseline	+10.5%
Photo and Video Scenario	Baseline	+5.3%
TabletMark Overall Rating	Baseline	+7.9%

Table 3.1.7: Performance Sensitivity to Operating System

3.2 Application Contribution to Performance Rating

The following charts show the contribution of each application model to the TabletMark overall performance rating on three tablets, one example for each supported platform.

These results are valid only for the specific systems tested. They will vary from system to system.

3.2.1 Android

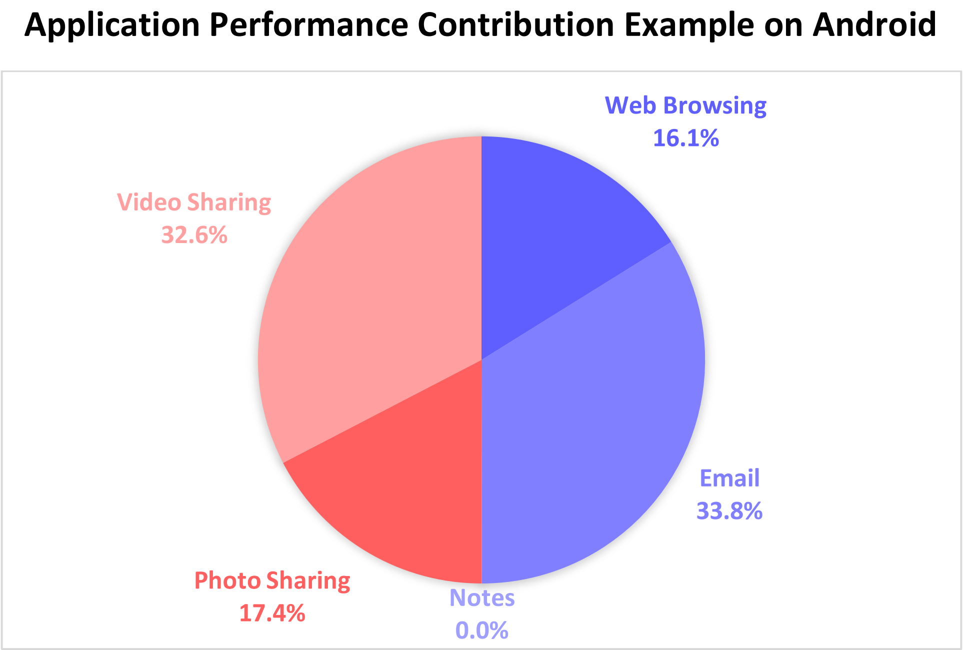


Figure 3.2.1: Application Contribution to TabletMark Overall Performance Rating Google Nexus 9, Android 5.0.1

3.2.2 iOS

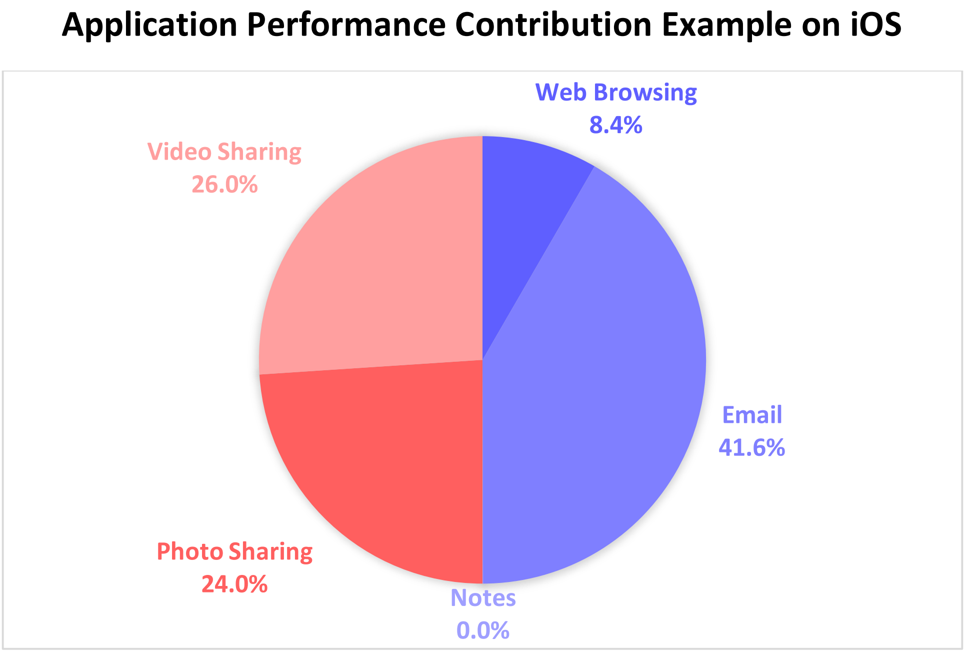

Application Performance Contribution Example on iOS

Figure 3.2.2: Application Contribution to TabletMark Overall Performance Rating
Apple iPad Air 2 (Wi-Fi), iOS 8.1.3

3.2.3 Windows

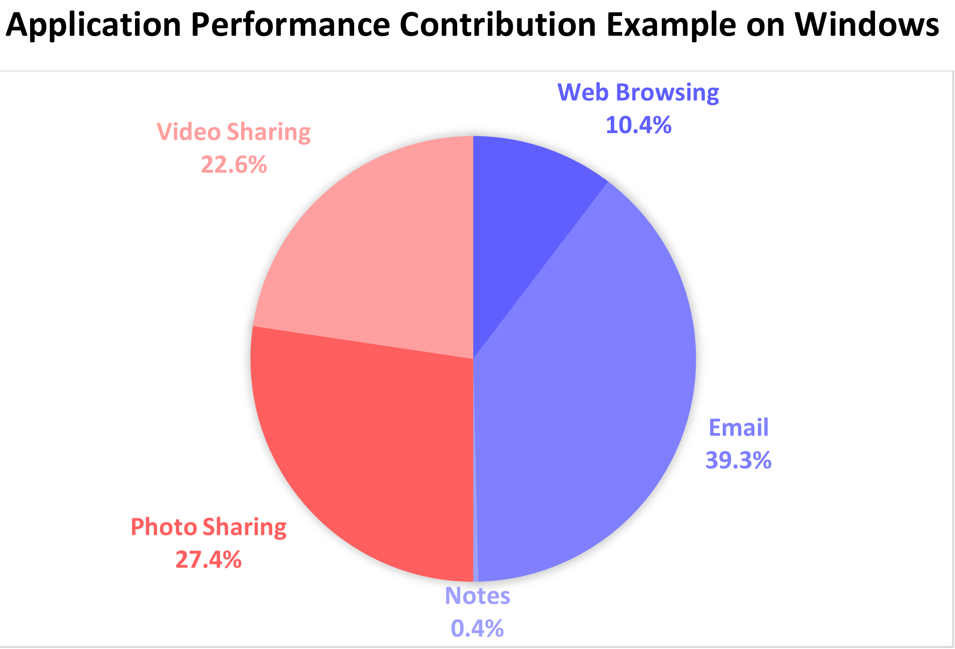


Figure 3.2.3: Application Contribution to TabletMark Overall Performance Rating
Lenovo ThinkPad 10, Windows 8.1 (32-bit)

3.3 Application Contribution to Battery Life Rating

The following charts show the time spent in each application model and in user idle time on the TabletMark calibration system during a battery life test.

3.3.1 Iteration Timeline

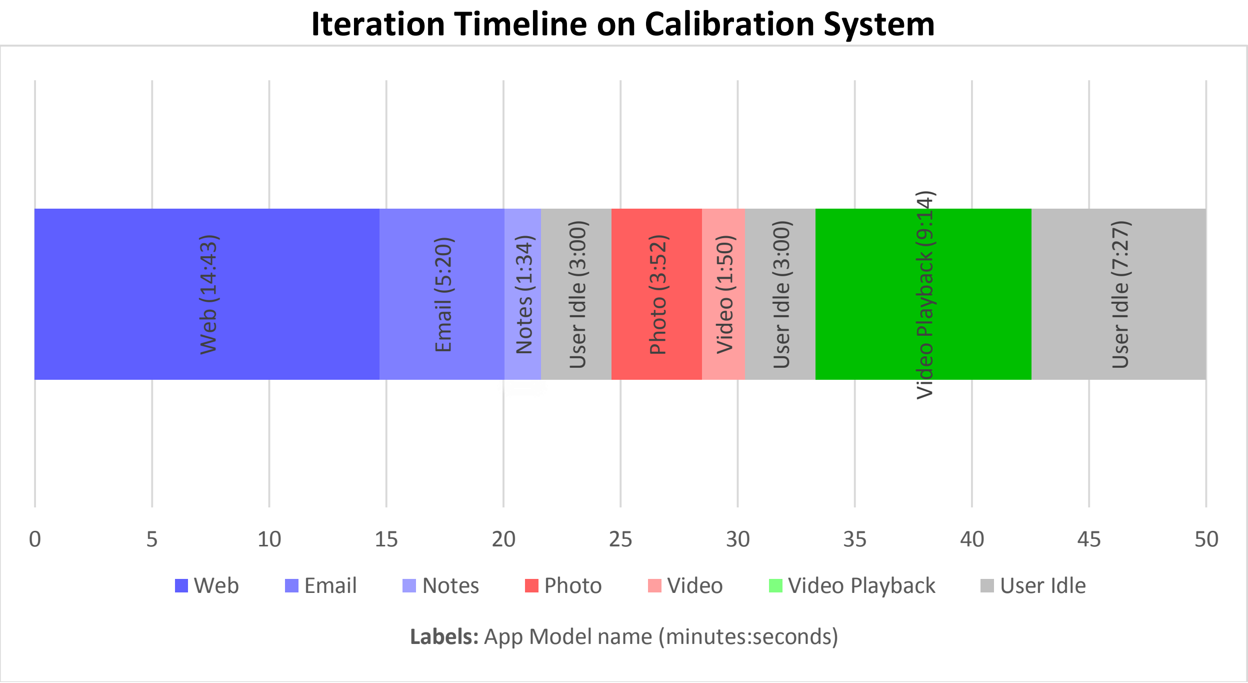


Figure 3.3.1: Time spent per application model in one 50-minute iteration of battery rundown test on calibration system

3.3.2 Application Model Contribution to Runtime

Application Models by Runtime Contribution on Calibration System

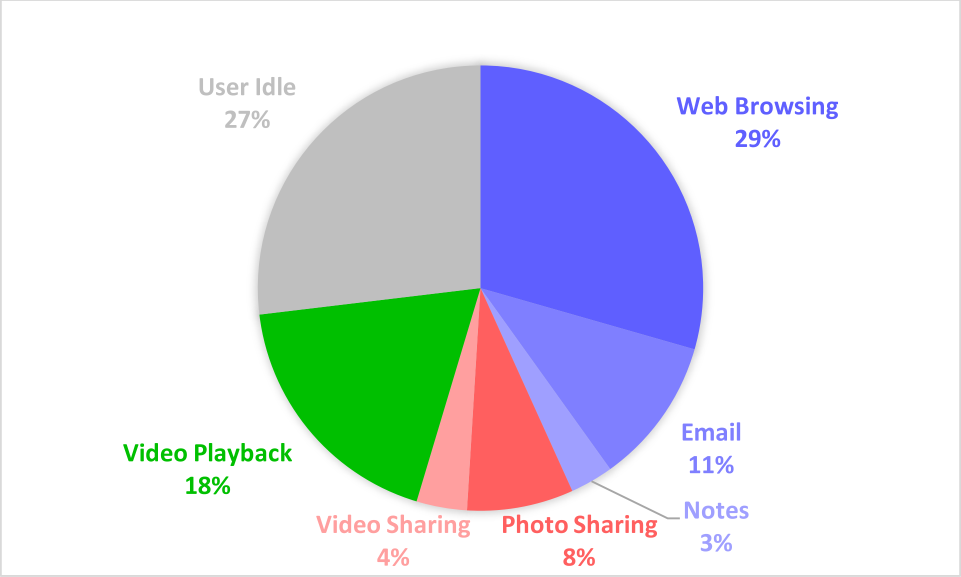


Figure 3.3.2: % time spent per application model
In battery rundown test on calibration system

APPENDIX A: Minimum System Requirements

TabletMark V3 for Android:

- Android 4.3 or newer (32-bit and 64-bit)
- Dual core processor (ARM or Intel)
- 1 GB system memory (500 MB free memory)
- 2 GB free storage memory
- 1024 x 768 display, 7” or bigger

TabletMark V3 for iOS:

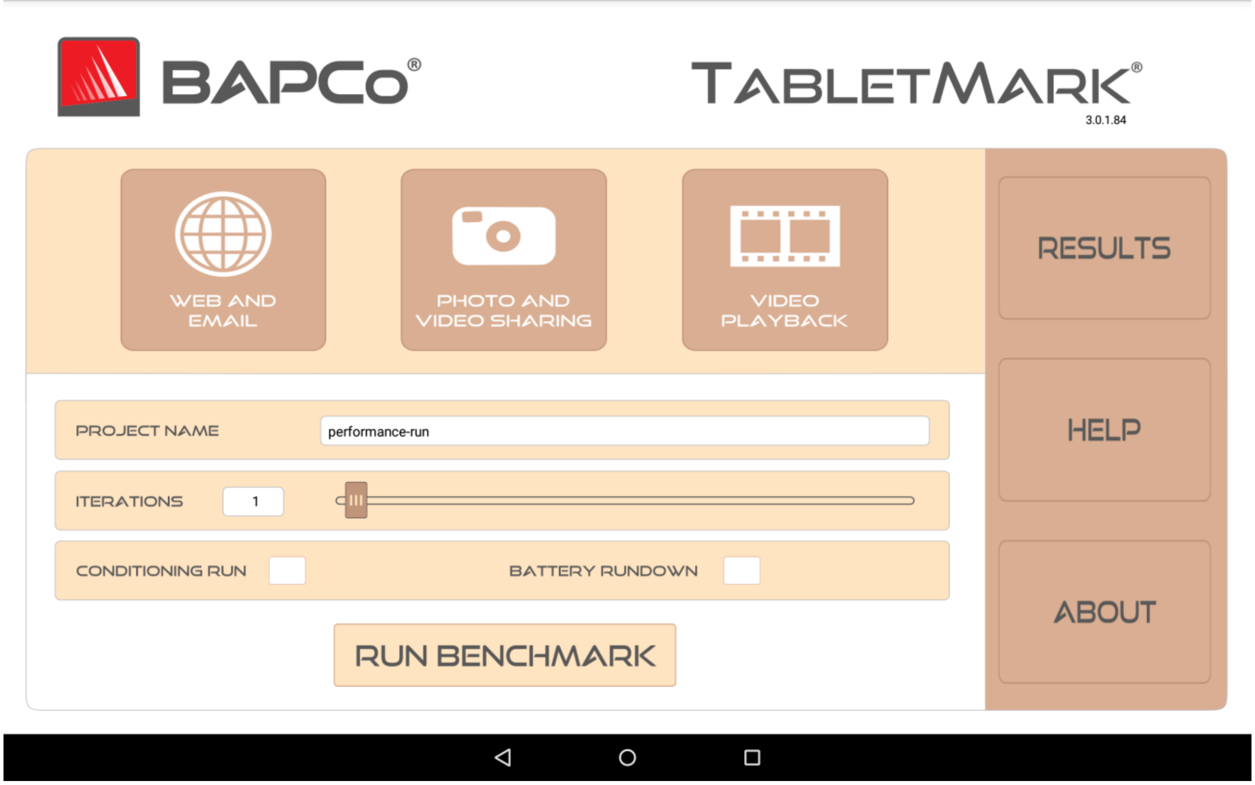
- iOS 7.1.2 or newer (32-bit and 64-bit)
- A6X processor
- 1 GB system memory
- 2 GB free storage memory
- 1024 x 768 display, 7” or bigger

TabletMark V3 for Windows:

- Windows 8.1 or 8.1 RT (32-bit or 64-bit)
- 1.3 GHz dual core processor (AMD, ARM or Intel)
- 2 GB system memory
- 2 GB free storage memory
- 1024 x 768 display, 7” or bigger

APPENDIX B: Screen Shots

The screen shots below are included to illustrate the user interface and workloads in TabletMark. Future releases may change the appearance of some elements of the benchmark.

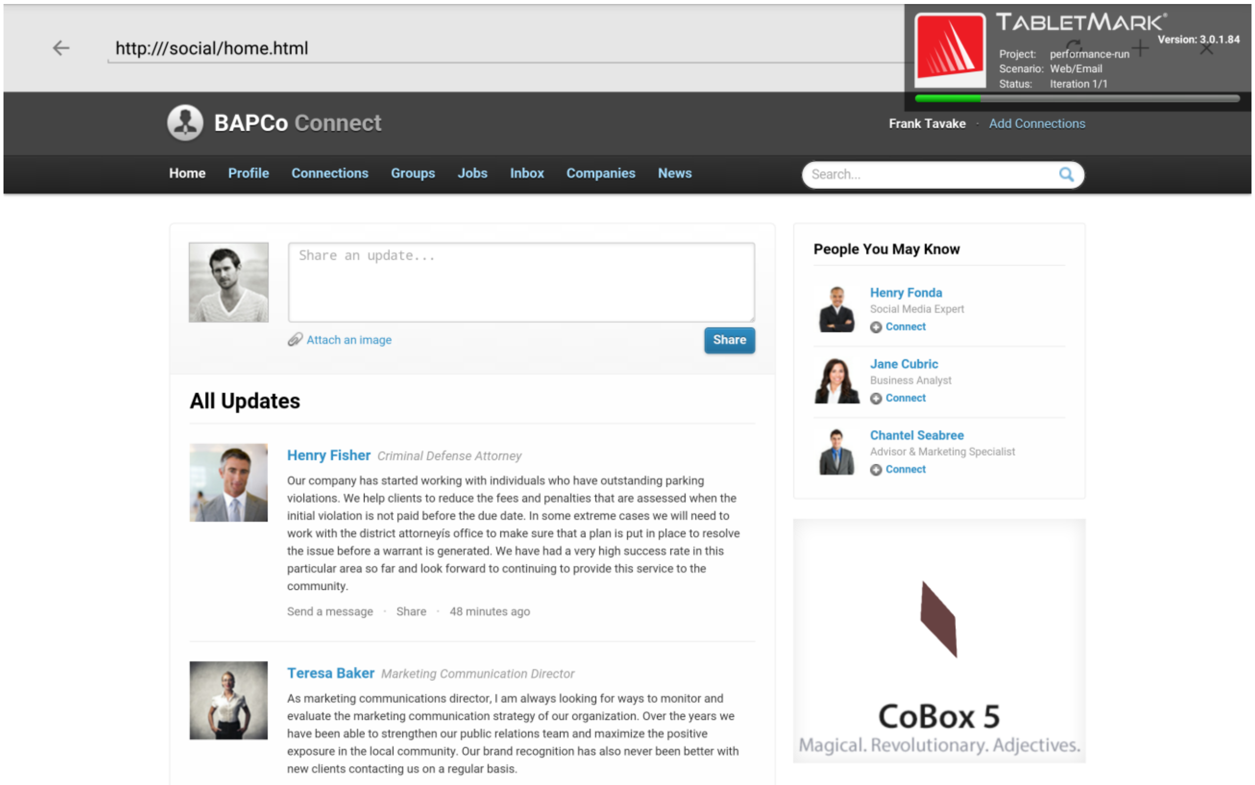


PROJECT NAME performance-run

ITERATIONS 1

CONDITIONING RUN □ BATTERY RUNDOWN □

Figure B.1: Test setup user interface



All Updates

Henry Fisher *Criminal Defense Attorney*

Our company has started working with individuals who have outstanding parking violations. We help clients to reduce the fees and penalties that are assessed when the initial violation is not paid before the due date. In some extreme cases we will need to work with the district attorneyís office to make sure that a plan is put in place to resolve the issue before a warrant is generated. We have had a very high success rate in this particular area so far and look forward to continuing to provide this service to the community.

Send a message · Share · 48 minutes ago

Teresa Baker *Marketing Communication Director*

As marketing communications director, I am always looking for ways to monitor and evaluate the marketing communication strategy of our organization. Over the years we have been able to strengthen our public relations team and maximize the positive exposure in the local community. Our brand recognition has also never been better with new clients contacting us on a regular basis.

Figure B.2: Web and Email scenario

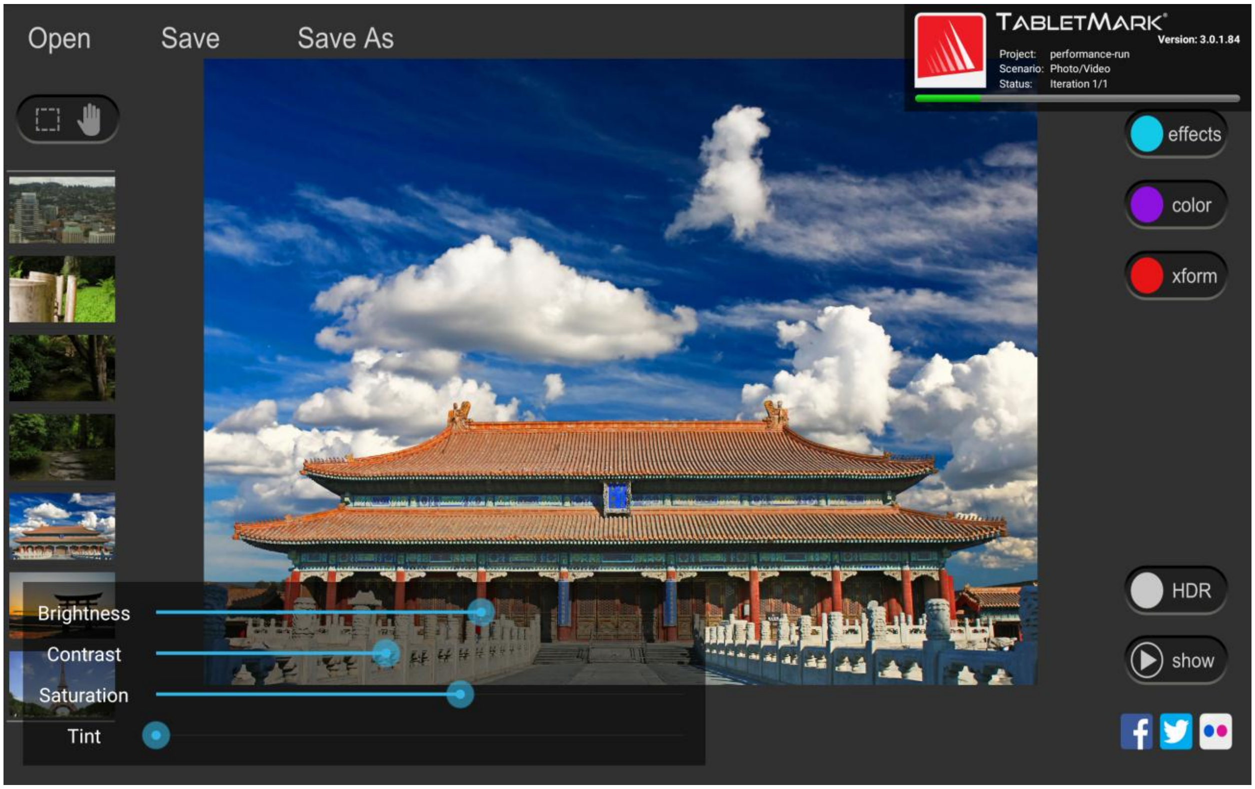
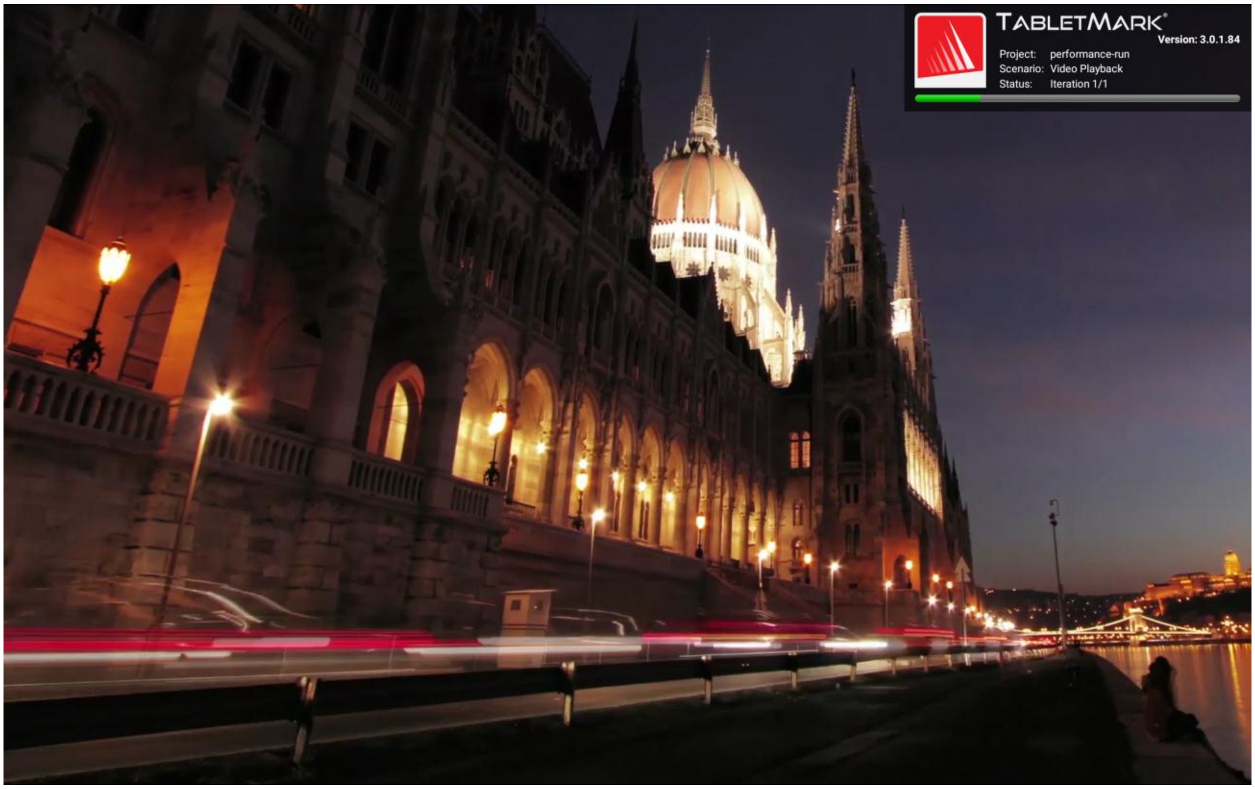


Figure B.3: Photo and Video Sharing scenario



Version: 3.0.1.84

Project: performance-run

Scenario: Video Playback

Status: Iteration 1/1

Figure B.4: Video Playback scenario