MicroStrategy | 2021

Mobile Architect

Planning, Implementing, and Optimizing the Intelligent Enterprise

Core learning for: Mobile Architect

CONTENTS

1

The Intelligent Enterprise

Introducing the Intelligent Enterprise

The Intelligent Enterprise is the ultimate data-driven organization. It effectively designs and implements Business Intelligence solutions while promoting effective use of data across your enterprise. This fosters growth and development, with a focus on data governance and alignment of strategic business goals to technology investments.

Getting there requires the right tools and structure to balance traditionally counteractive forces. Agility and governance, convenience and security, ease of use and enterprise functionality are all critical capabilities that the MicroStrategy platform is positioned to support with its unique intelligence architecture.

A successful Intelligent Enterprise:

- Drives adoption and success of enterprise Business Intelligence
- Coordinates BI implementations
- Maintains sound data governance and a single version of the truth
- Provides a formal approach to documenting processes, creating content, and ongoing maintenance

- Ensures that BI is aligned with enterprise strategy

Along with quick and easy ad-hoc departmental solutions, MicroStrategy has the robust, proven ability to support high-scale deployments and establish a single source of the truth. MicroStrategy's tools include data-governance features, administrative controls, and management capabilities with the enterprise platform software, all critical to the Intelligent Enterprise.

The Intelligence Center

The Intelligence Center is comprised of a team of expert architects who define, develop, and provide guidance across the enterprise. With the collective know-how to maximize BI investments, the Intelligence Center drives optimization of system architecture, upgrades, configuration, performance, scalability and stability.

Your role: The Mobile Architect

Your company, Bee Good Health (BGH), is a forward-leaning health care analytics company that has decided to transform into an Intelligent Enterprise. As MicroStrategy users, Bee Good Health wants to leverage their existing investments and successfully deliver powerful analytics and mobility solutions

across the enterprise. With your extensive background in mobile solutions and design, your CEO has selected you to become Bee Good Health's Mobile Architect.

As the Mobile Architect, you lead mobile solutions and development for your enterprise. The scope of the Mobile Architect includes resolving challenging technical problems, such as creating an effective caching strategy, implementing Mobile Application Management to ensure security and control over enterprise applications, and optimizing mobile performance and application stability. You oversee a team of mobile designers and their leads who need corporate guidelines and standards as they contribute to your organization's Enterprise Mobility.

As the Mobile Architect in your organization, you should have the following experience and qualifications:

- Experience architecting MicroStrategy Mobile Applications end to end.
- Thorough knowledge of all capabilities in the MicroStrategy Suite, including Developer, cube creation, management, automation, and Web.
- Extensive hands on experience with the following technologies: AppConfig-compliant EMM providers, Android, and Apple iOS platforms.
- Ability to debug and fix technical issues.
- Ability interact with business users to gather requirements and provide support.
- Ability to develop comprehensive project plans, define project scope, and establish key performance indicators (KPIs).
- Performance tuning and troubleshooting experience.
- Knowledge of performance best practices and how to build apps that are resource efficient.

You can gain this knowledge through a combination of experience and training, found in the following MicroStrategy classes:

- Overview of Enterprise Mobility
- Administration for Enterprise Mobility
- Advanced Documents: Interactivity and Joining Datasets
- Advanced Mobile Applications
- Database Write-back: Mobile Transactions
- Advanced Administration for Enterprise Mobility

Enterprise Mobile deployment: Requirements and challenges

Put intelligence in the hands of your workforce and transform applications and business processes. MicroStrategy delivers more ways for organizations to quickly deploy mobile productivity apps on any standard iOS or Android mobile device. To fully harness the power of mobility, enterprises need to plan for a number of different requirements and challenges. The ideal enterprise mobile deployment:

- Implements an iterative user-centered app lifecycle
- Supports an enterprise-ready infrastructure
- Integrates easily with other technology investments
- Provides optimized user experience

In this course, we revisit the mobility topics you know, but with a holistic approach of the Mobile Architect in your new Intelligent Enterprise. This course focuses on the key competencies to help you succeed as the Mobile Architect and lead your Intelligent Enterprise to produce a best-in-class mobile solution.

2

Design Thinking Methodology

As Bee Good Health (BGH) transitions to becoming an Intelligent Enterprise, you lead and advise on the development of mobile solutions, and establish guidelines to craft best-in-class enterprise applications. As the Mobile Architect, you are responsible for the apps your team creates, serving as the advocate for all things mobile at BGH.

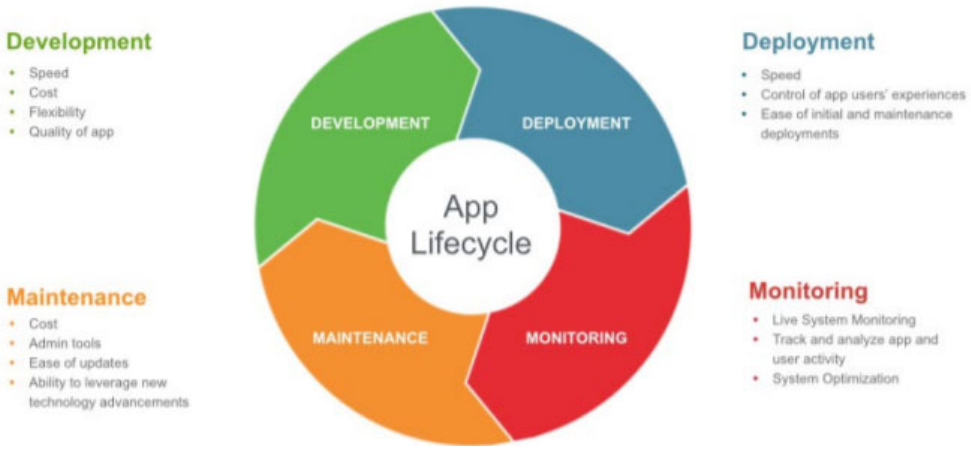
This chapter introduces different methodologies used by designers and developers to solve practical problems and create enterprise mobile solutions.

As the Mobile Architect, you guide your team to:

- Choose the methodology that best works for your solution.
- Identify the problem that the app solves.
- Narrow the scope of the app based on the wants and needs of the key stakeholders.
- Establish corporate guidelines for best-in-class enterprise applications.
- Develop graphical organizers to visualize the app.

Mobile app lifecycle

Mobile applications are iterative by their very nature. The pace of mobile technological advancements and the changing, evolving needs of business users means that developers must regularly update and enhance their apps. This iterative nature of mobile apps is not a bad thing, in that it gives developers the opportunity to constantly improve their efforts, it does however make having a tool that allows them to quickly and easily navigate the app lifecycle paramount.



Development methodologies

Designing and developing a mobile application is not an easy task. But, as the Mobile Architect, you can lower development cost and increase performance by choosing the correct methodology for your team and application.

Two popular mobile app development methodologies are Waterfall and Agile.

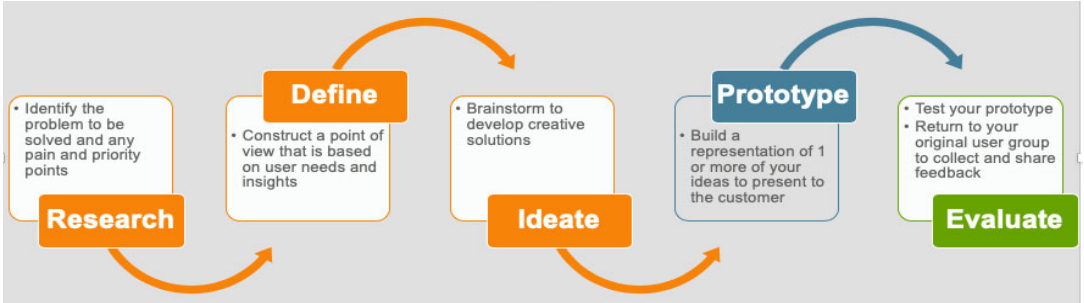
- Waterfall methodology is a sequential design process where development proceeds steadily onward. Stakeholder and customer requirements are gathered at the beginning of the project and not revisited during development.
- Agile development, by contrast, is quick and iterative. Tasks are divided into short work phases with frequent assessment and adaptation of plans. This methodology allows for a more flexible app development process.

The agile development process has become increasingly popular because of its proven success. The agile development process works well alongside the design thinking methodology, which is a human-centered methodology that allows designers to focus on their users' most pressing concerns. Together, agile and

design thinking present a new way to conceptualize development, and together they offer a collection of hands-on methods to help apply this novel framework. This is the approach that we follow for this course.

Design thinking methodology

Design thinking is a solution-based approach to solving complex problems, such as creating a mobile application for multiple users. This approach is user-centric and draws upon logic, creativity, experience, and systemic reasoning to explore the possibilities of what could be created to benefit the end users. Solution-based thinking focuses on finding solutions and coming up with something constructive to effectively tackle a problem.



The design thinking methodology is an iterative design process that normally consists of the following stages:

- Research
- Define
- Ideate
- Prototype
- Evaluate

The stages are not always performed sequentially, and can occur simultaneously as well as repetitively.

Research: Gathering requirements

Since the design thinking methodology is a human-centered process, as a starting point, you and your team should gain an understanding of the wants and needs of the users, as well as the problems to be solved. A Visual Reference guide,

like the one below, can be used to list the wants and requirements for each key stakeholder all in one place.

			
TITLE			
ROLE			
RELATION TO PROJECT			
NEEDS (USER &/OR ORGANIZATIONAL)			
REQUIREMENTS (MUST-HAVES)			
NICE TO HAVES			
PREFER NOT TO HAVE			

In addition to the wants and needs of the stakeholders, your team also needs to:

1. Define the business requirements of the app that are necessary to meet enterprise objectives. Questions such as what is the purpose of the app and how does it improve the current process can be asked.

2. Establish the product and technical specifications. For example, which platforms should the app be built for? Does your organization have an existing provisioning profile?

3. Consider the end users of the app. Who is the target audience?

4. Define any aspects that the app or app creator must rely on to meet the app objectives, such as API documentation.

5. Determine constraints, such as scope, budget, and time.

6. Determine if transaction services are necessary for the app.

Best practices for mobile application requirements include:

Best Practice

- Requirement documents should remain high level. Details are documented by designers during the storyboard and wireframe phases.
- Work with the Application Architect and System Administrator to build requirements and understand enterprise assets that can be leveraged in a mobile design.

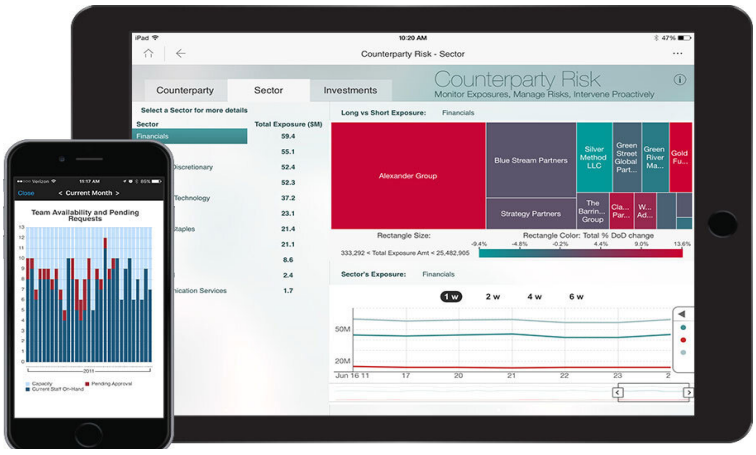
Mobile solutions for your enterprise

To assist in developing the technical specifications of the app, it is important for you and your team to have an understanding of the mobile solutions available through MicroStrategy. The mobile solutions that MicroStrategy offers are:

- MicroStrategy Mobile
- MicroStrategy Library Mobile
- MicroStrategy HyperIntelligence for Mobile

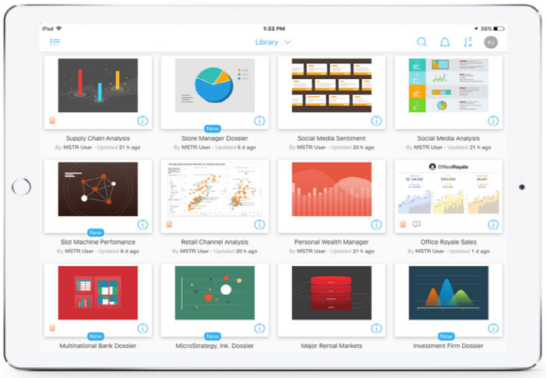
MicroStrategy Mobile

MicroStrategy Mobile is an interactive interface of the MicroStrategy BI platform that allows mobile business users to harness the analytical power of MicroStrategy through the use of their iPhone, iPad, and Android devices. MicroStrategy Mobile provides application developers a new way to develop and deploy mobile applications that are faster, easier, and more maintainable than using traditional Integrated Development Environments. The MicroStrategy Mobile app can be customized and re-branded for your organization using the Mobile SDK.



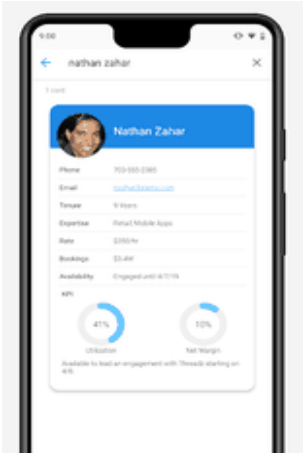
MicroStrategy Library Mobile

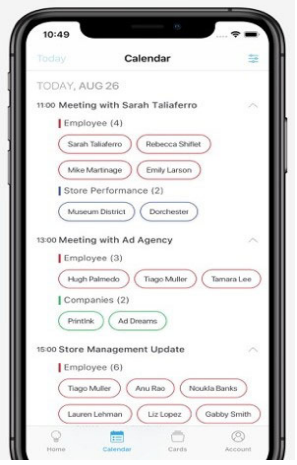
MicroStrategy Library Mobile is a personalized portal that allows end users to access dossiers and documents saved to their Library from both iOS and Android mobile devices. Library combines powerful search functionality, real-time collaboration, and a simple design. MicroStrategy Library Mobile app can be customized and re-branded for your organization using the Library SDK.



MicroStrategy HyperIntelligence for Mobile

HyperIntelligence for Mobile is a stand-alone iOS and Android application that allows you to access HyperIntelligence Cards using a mobile device. From the app you can connect to any environment and search for cards with an in-app search, or have cards come to you through calendar-based and reminder-based notifications.





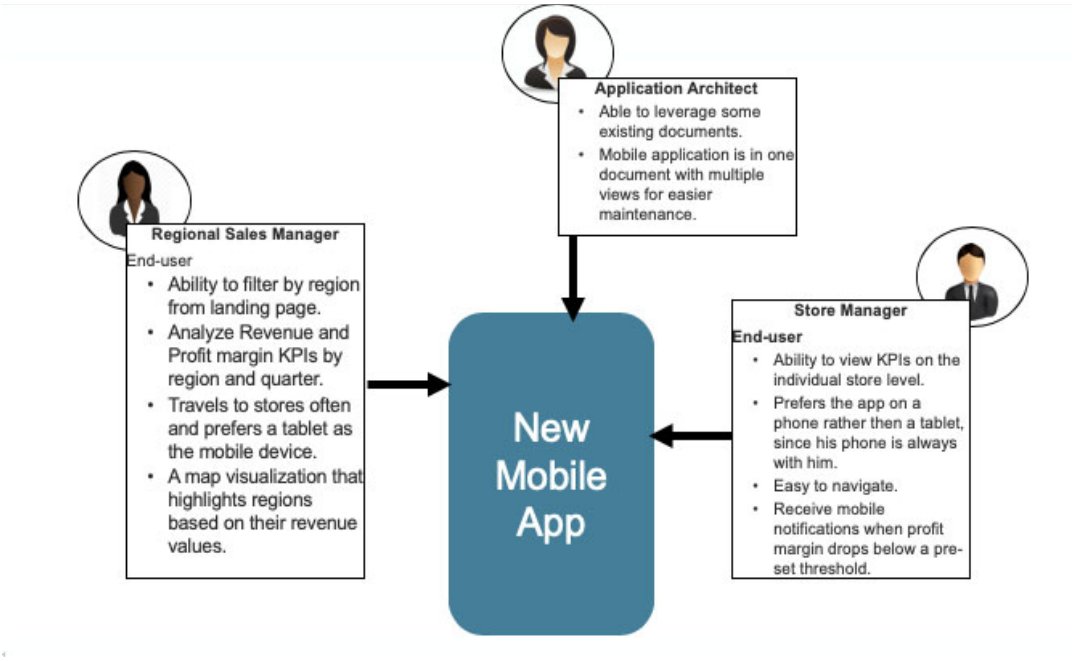
Exercise 2.1: Complete the Visual Reference guide based on the stakeholder's requirements

In this exercise, decide which mobile solution, Mobile app, Library Mobile, HyperMobile, or a combination of the three, best suits BGH based on the requirements. Next, complete a Visual Reference guide based on the wants and needs from the key stakeholders at BGH.

Complete the Visual Reference guide

1 In your exercise files, open the Visual Reference guide file.

2 Fill in the guide based on the following requirements from the application's stakeholders.



Define: Documenting and designing the app

Like other organizations, before becoming an Intelligent Enterprise, BGH's apps did not adhere to any established standards. One app designer might choose to build apps only for Android devices and another for iOS. One designer documents all of the data used in the apps while another designer only documents the app's workflow. This leads to quality control issues and a lack of cohesion in BGH's apps.

Intelligent Enterprises provide a formalized approach to the documentation processes and creating content. As the Mobile Architect, you should take this formal approach and apply it to the mobile application creation process.

Establishing standards can help guide your team from early concept stages to the final build achieving consistency and serving as the foundation of all enterprise applications. Your team can also use these standards to update the current apps to ensure they provide an intuitive, self-explanatory, and enhanced user experience for users to obtain insights into enterprise data.

Streamlined documentation

The Mobile Architect should require that app designers record all information about their app in a streamlined document, allowing the team and broader organization to understand what objects and parameters were used to build the app. This accomplishes several goals:

- If the original designer leaves the company or moves to another project, new team members can easily maintain the app by understanding how and why the app was built.
- Track application objects for troubleshooting app issues.
- Keep others in the Intelligence Center abreast of objects your team is using.

What do you think should be documented during the app creation process?

For example documentation, see the Mobile Application Documentation file provided within the exercise files for the class.

App building blocks: MicroStrategy objects

To support the creation process, set standards for what objects are used and how app pages are designed. The Mobile Architect needs to ensure that app designers are using the correct, approved MicroStrategy objects. Data governance is an important factor in a successful Intelligent Enterprise, because organizations need one set of information that analysts can access and trust. Each of these important components of an intelligent mobile app are described below.

Schema objects

Schema objects are logical representations of the structures of a data warehouse, such as facts and attributes within a MicroStrategy project. Typically, the Analytics Architect creates the schema objects for app designers to use in their documents.

However, there may be times when your team needs specific schema objects that are not readily available in the metadata.

For example, BGH wants to create an HR app that analyzes employee information. Though the Employee fact table exists in the warehouse, you need to work with your Analytics Architect to create the attributes needed for the app.

Application and document objects

These objects, such as prompts and metrics, are used to create reports and documents. Application objects are generally created by a report designer or metric designer and are built from schema objects. As the Mobile Architect, you oversee the creation of these objects, as well as objects specific to a document for mobile apps, ensuring that they are used appropriately. For example, the Mobile Architect can specify that designers should avoid using nested panel stacks due to the complexity they can create for future changes, enhancements, and maintenance on the application.

Best Practice

You should also ensure that application objects are stored in the correct folders within a project so you and your team know where to find what you need. For example, you can require that all finished documents are placed in a folder named Completed Mobile Documents in the Shared Reports folder.

Pre-design style guide

The Mobile Architect should define pre-design guidelines before the app creation process begins. This not only streamlines the app development, but these guidelines also help avoid reworking an app later in the process. These guidelines should establish an accurate focus on how to define the appropriate audience, and build overarching common requirements for every application your team creates.

The style guide should include guidelines for such items as spacing and positioning of objects, maximum number of visualizations per page, and navigation placement. With BGH's unique branding of style, color palette, fonts, and images, this is also important to incorporate into your style guide. Corporate branding is important, because it provides context and distinguishes your company from others. To ensure BGH enterprise apps are designed according to the corporate style guide, work with your marketing and advertising team to identify and manage the image assets for mobile application branding.

Exercise 2.2: Create BGH's pre-design guidelines

In this exercise, you create the corporate design standards and color scheme for BGH mobile applications based on the logo. Be sure to include standards for style/ layout, colors, fonts, images, and navigation.

Create the Design standards

1 Using BGH's official logo, create the design standards that the mobile application should follow. Use the space below to create the style guide for Bee Good Health.

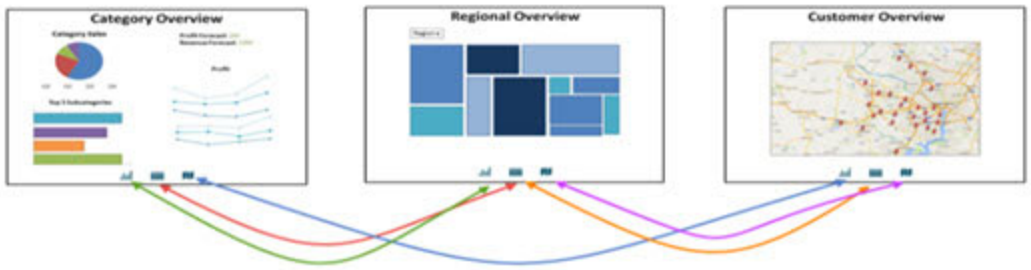
Ideate: Mapping the app

With a solid understanding of your users and the problem to be solved, it is time to start working on potential solutions. In an Intelligent Enterprise, mobile apps focus on delivering and supporting departmental decision-making, and a user-centered design empowered by data discovery at every level.

Storyboards

Now that you've created BGH's pre-design guidelines, your team is ready to storyboard the app. Storyboards are a key part of the app design process. They are a graphical organizer of images displayed in sequence to visualize the app. Storyboards assist in revealing whether an app concept works, identifying errors at an early stage, forcing thinking about the user flow, and putting people at the center of the design process.

In the following storyboard example, the app consists of three pages. Users navigate the app using a tab bar that links to each page.



The Mobile Architect leads the storyboard process to determine:

- What visualizations are contained in the various sections in the app.
- Which interactive controls, such as links and selectors, are included in the app.
- What data is going to be displayed in the app.
- The navigation structure of the app.

Once key features and the workflow of the app have been determined, work with the User Interface (UI) and User Experience (UX) teams to create the app's storyboard. The storyboard defines the layout, navigation, and data presentation based on the mobile application requirements. The storyboard should illustrate and outline the app showing screens of content and transitions between them.

Since storyboards don't need a high level of detail, they can be drawn by hand or created using a computer application, such as Adobe XD.

Best Practice

Best practices for building a storyboard include:

- Determine the app navigation scheme before creating the storyboard to minimize development time and issues. Several things can influence the app navigation scheme like the type of device, the amount of content within the app (navigation bar vs hamburger or popup navigation), and data-driven

navigation (clicking attribute elements in a grid to dynamically answer a prompt on another page or screen).

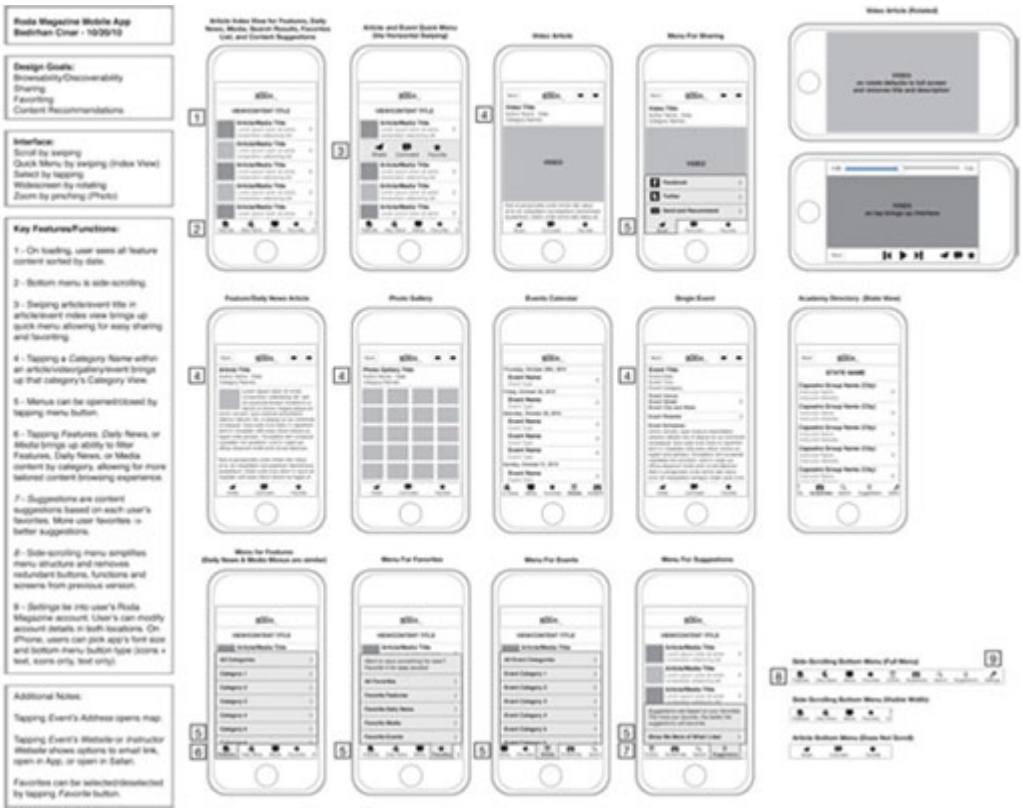
- Draw initial storyboards during the research process on white boards so multiple people can contribute and brainstorm together.

Wireframes

After storyboarding, the Mobile Architect works with the UI/UX team to create the wireframe standard for a prototype of the mobile app. A wireframe is the blueprint of the app. It focuses on function and structure, while mapping out the skeleton of each screen that a user encounters. The wireframe lays out the structure, hierarchy, and relationship between the components that make up your app.

As with storyboards, a standard wireframe process focuses on the user experience to design an effective navigation workflow and ensure app usability. The intention of these renderings is to focus on what the screen does, and not exactly what it looks like, so the wireframe should be simple. Wireframes are especially useful to keep track of complex apps with a lot of pages and sections.

The example below shows a mobile app wireframe. Notice that each screen is displayed and the illustration includes various orientations of the device.



For example, your standards may require the following wireframe steps:

1. Create your wireframe template and determine the wireframe layout.
2. Define an information hierarchy with typography that uses different font sizes to differentiate between different levels of information.
3. Test the wireframe on future app users.

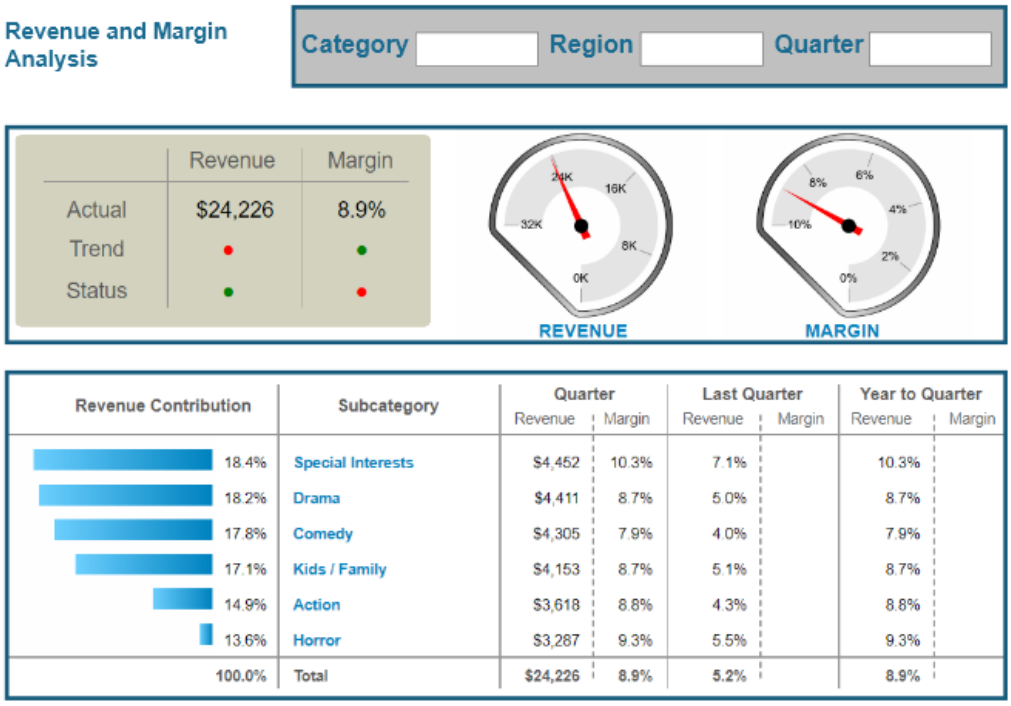
Best Practice

Best practices for building a wireframe standard include:

- Wireframe designers should collaborate with potential app users during the process to help tailor the app to the end user and provide a first-rate app experience.
- Wireframes should include navigation patterns to convey the workflow of each app page.
- Keep the wireframe simple and do not overuse color. Wireframes test navigation and usability, not graphic design.

Exercise 2.3: Create a BGH standardized storyboard with Adobe XD

The mobile app development team of BGH is now ready to create the storyboard for the organization's new mobile app. As the Mobile Architect, you ensure that the storyboard shows what visualizations, data, links, and selectors are contained within the app. Your team also uses the storyboard to map out the navigation structure of the app, as well as align the color scheme to match BGH's corporate branding.



In the following series of exercises, you and your team storyboard a pre-designed document that meets corporate requirements to standardize the design process. Once the storyboarding process has been documented, your team then designs the landing page for BGH's new mobile application.

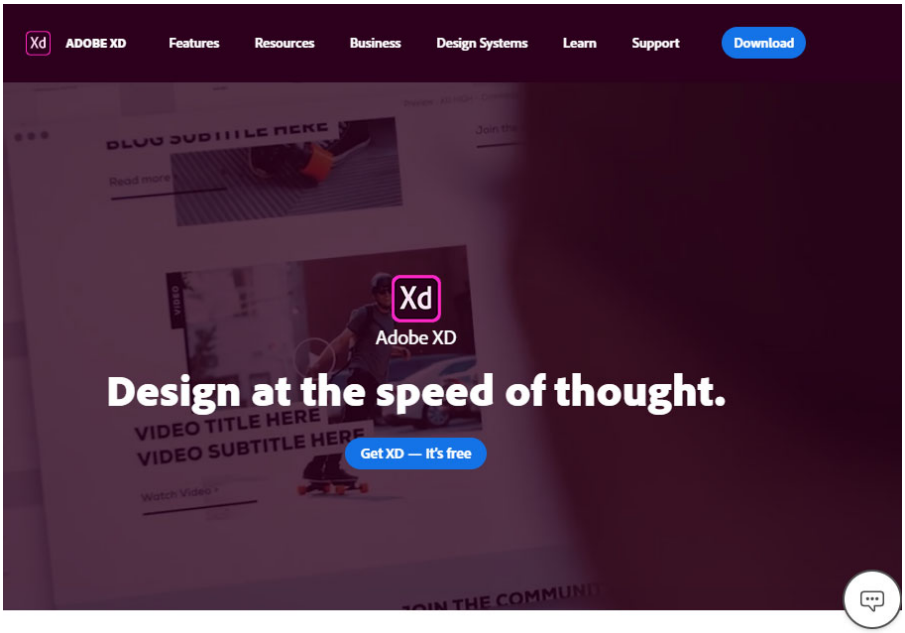
Design a mobile app: Download Adobe XD

Adobe XD is a free, all-in-one application made for designers and teams to design, wireframe, prototype, present, and share a mobile app. As the Mobile Architect, you want all storyboards to be created in XD because your app designers can

seamlessly iterate and share interactive prototypes with team members across devices and platforms, including Windows, Mac, iOS, and Android.

If you are unable to download Adobe XD to your computer Draw.io can be used as an alternative. Draw.io is a free online diagram software that can be accessed by following this link: ***https://app.diagrams.net/***.

1 In a web browser, navigate to **https://www.adobe.com/products/xd.html**. Click **Download**.



2 Open the **XD_Set-Up.exe** file to launch the XD Installer.

3 Login to your existing Creative Cloud account or create one for free. Then, click **Start Installing**.

4 Once your download is complete, open **Adobe XD**.

Create a new artboard

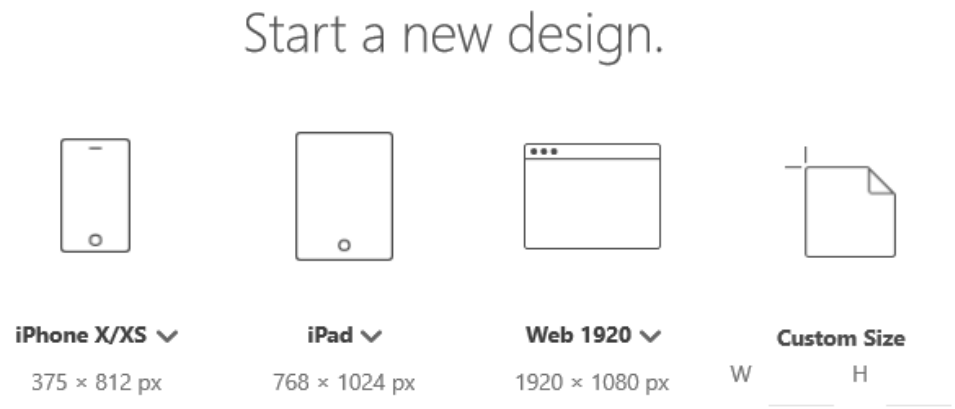
Artboards represent screens within an app. To create a storyboard, add an artboard for each screen of your mobile app all in one XD file.

Best Practice

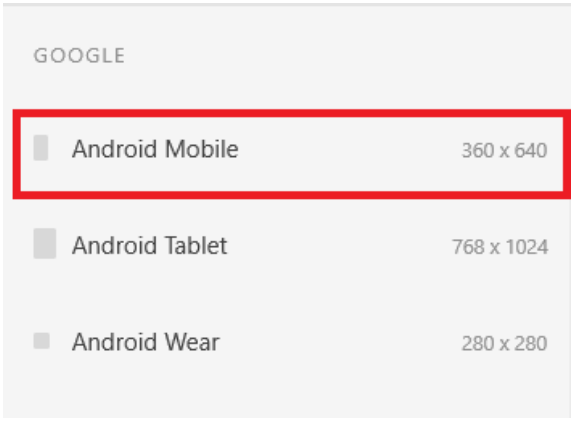
Based on the requirements from the stakeholders, the app needs to be accessible from both phones and tablets. Like phones, tablets also have a variety of dimensions and sizes. As a best practice, you should require app layouts to be as flexible and responsive as possible. Instead of defining your layout with rigid

dimensions, layouts should respond to different screen sizes and orientations. Designers can leverage the Responsive Resize feature in Adobe XD and MicroStrategy's Responsive Design to achieve app flexibility.

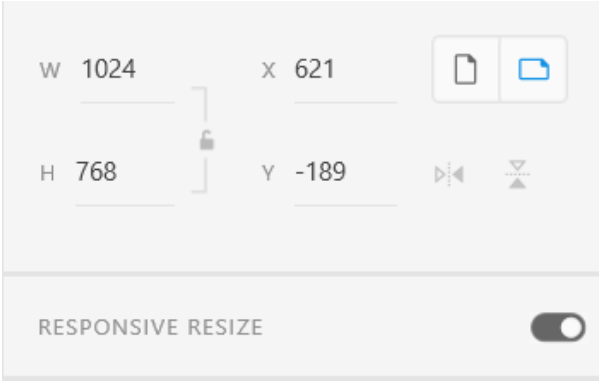
1 In the Home tab, click **Custom Size** to start a new artboard.



2 To set the device you want to design for, click the **Artboard** icon on the left. Then, select **Android Mobile** under Google.



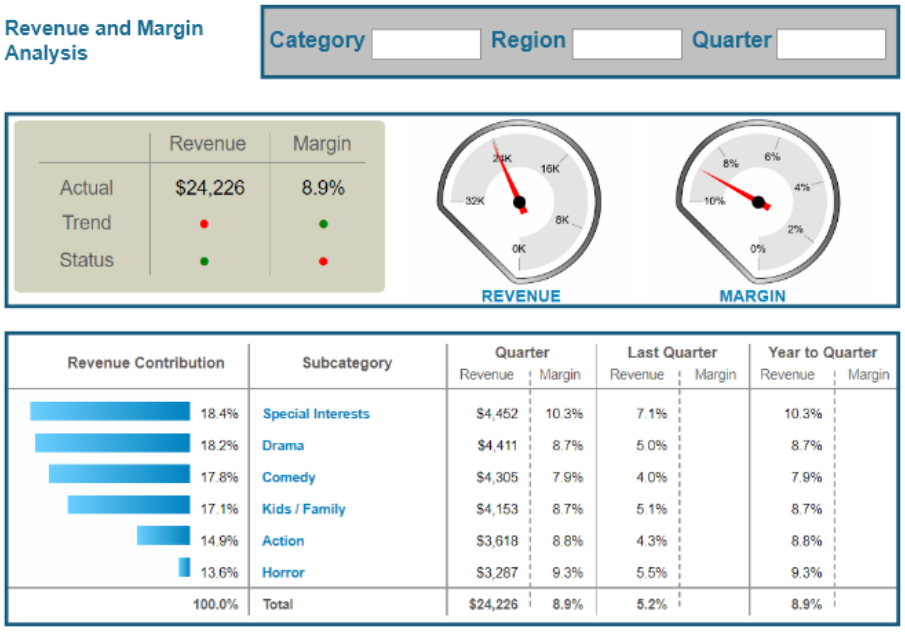
3 To complete your artboard's setup, scroll to the top of the artboard menu and change the orientation to **landscape**. Then, enable **Responsive Resize**.



With Responsive Resize, you can resize groups of objects while keeping their placement and scalability.

Design the artboard for the Cockpit document

To familiarize your team with Adobe XD and create a standard for BGH's storyboarding design process, reverse engineer the Cockpit document, one of the most highly utilized documents within BGH.



The completed artboard of the Cockpit document is shown above. The detailed steps to create this artboard in XD are listed below, or you can challenge yourself to create it on your own.

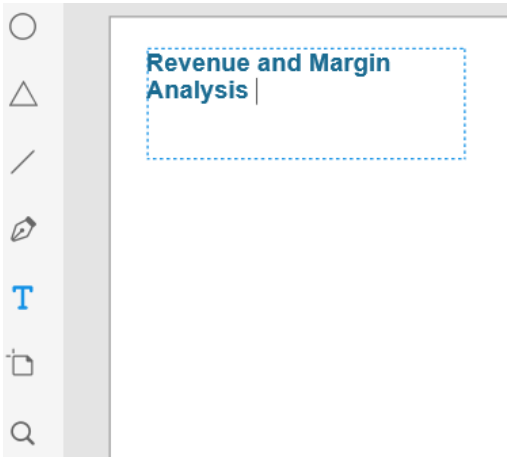
Insert and format the title

1 In Adobe XD, click the **Text** icon in the left toolbar to add a text box.

2 Draw the text box in the upper-left corner of the canvas, then type **Revenue and Margin Analysis**.

3 Highlight the text and change the font to **Arial**, **bold**, **size 28**.

4 To match the font color in the document, select the box next to **Fill**. Type **#1A6B91** for the Hex number and press **Enter**.



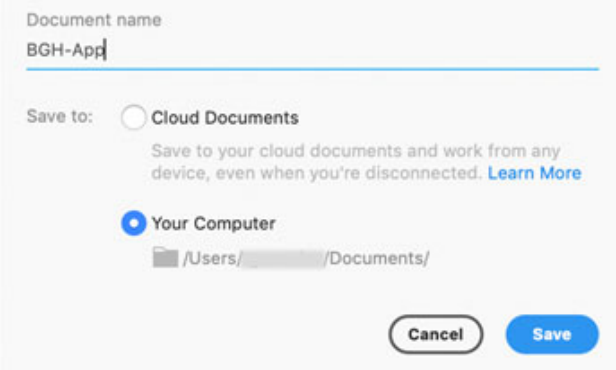
Save your design

Best Practice

5 Double-click the artboard title, **Android Mobile - 1**, and type **Revenue and Margin Analysis**. You should require your designers to name each artboard. Naming artboards help you keep track of which screen is which in an app design.

6 Click the **File** menu, and select **Save**.

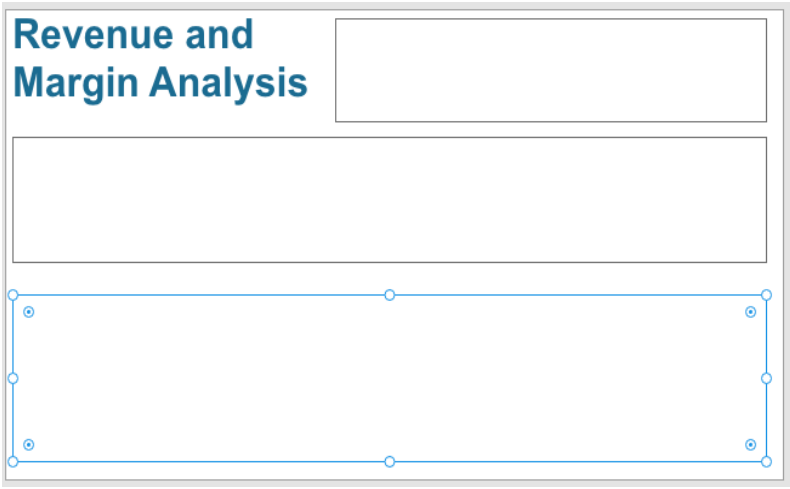
7 For the **Document name**, type **BGH-App**. Select **Your Computer** and click **Save**.



Add the artboard layout

1 Click the **Rectangle** icon and draw a rectangle across the top of the artboard, to the right of the title, to hold the selectors.

2 Draw two more rectangles, as shown below, to hold the visualizations.



Notice that as you draw the rectangles, XD's Smart Guides help you align the rectangles to other objects on your artboard.

Format the rectangles

3 Click the **Select** icon. Press **Shift** on your keyboard and click the three rectangles on your artboard.

4 Change the **Border** color to **#165B7C** and change the border size to **4**.

Add content to the artboard

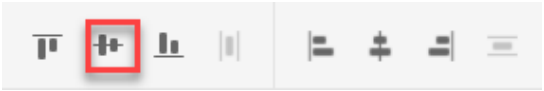
1 Click the **Selectors rectangle** (the top rectangle). Change the **Fill** color to **#C0C0C0**.

2 Add **three text boxes** and **three rectangles** as placeholders for the selectors.

3 Type **Category**, **Region**, and **Quarter** in the text boxes.

4 Change the text size to **24**.

5 To align your objects, select the **three text boxes** and **three rectangles** in the Selector box. In the Alignment Tools panel, click **Align Middle**.



Best Practice

The alignment tools are a more efficient method for adjusting objects compared to manually moving objects. Not only can you align all of your objects on your artboard, you can also evenly distribute them horizontally or vertically, as in this step.

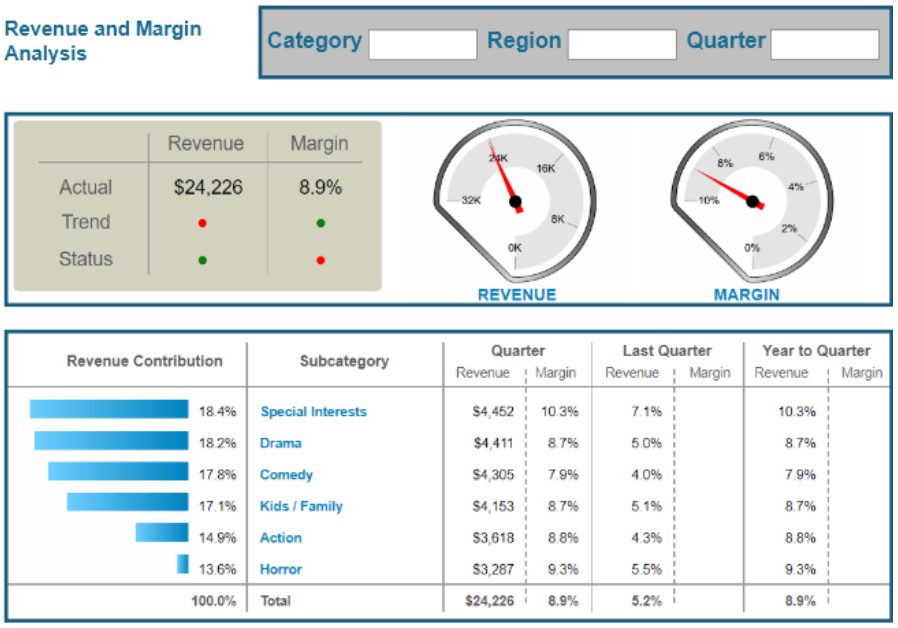
Complete the artboard with visualizations

6 In the exercise files provided by your instructor, open the **Storyboarding** folder. Arrange your screen so you can see both the folder and XD.

7 Click and drag **Analysis - 1.png** to the middle rectangle and **Analysis - 2.png** to the bottom rectangle.

8 Double-click the **Analysis - 1** image, then resize and position the image to fit in the rectangle. Repeat for **Analysis - 2**.

9 **Save** your design.



You have now completed the Revenue and Margin Analysis artboard. For additional practice, you can storyboard the Store Performance Management document located in Shared Reports/MicroStrategy Platform Capabilities/ MicroStrategy Report Services/Dashboard-style Documents.

Linking pages for a seamless workflow: App Navigation

Guide your users to the data they need, using one landing page that combines multiple BI applications. To provide a simple end-to-end navigational workflow, designers should add links to app pages using the Links Editor. The Links Editor can configure a link using one of the following link types:

- Navigate to this URL
- Perform an event on a Mobile device
- Run a specific report or document

Each link type is discussed in more detail during the creation of the app.

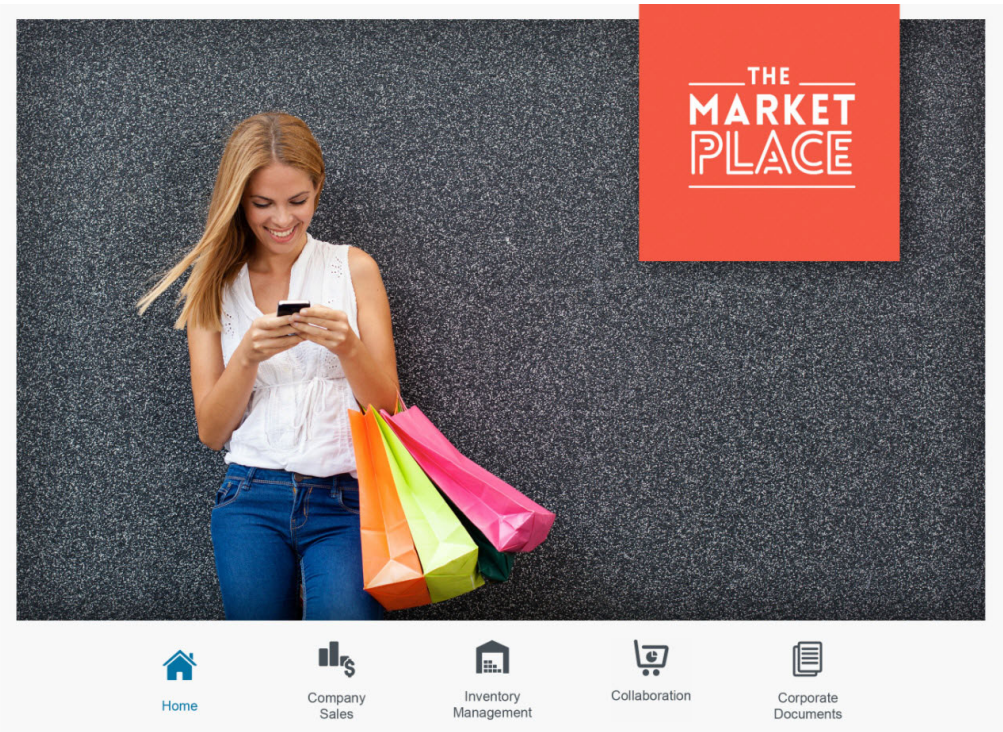
Navigational workflow: Landing pages

Best Practice

Intelligent Enterprise applications need to provide an end-to-end navigational workflow along with a look and feel of an optimized design. To help improve performance and establish the desired workflow, you should require app landing pages. The landing page should start with a simple and data-light document, with a clean look and easy access to links for various documents placed appropriately.

Not only does this start the user off with a sense of good performance, it also makes it faster for them to choose which way to go to reach the right set of data they are looking for.

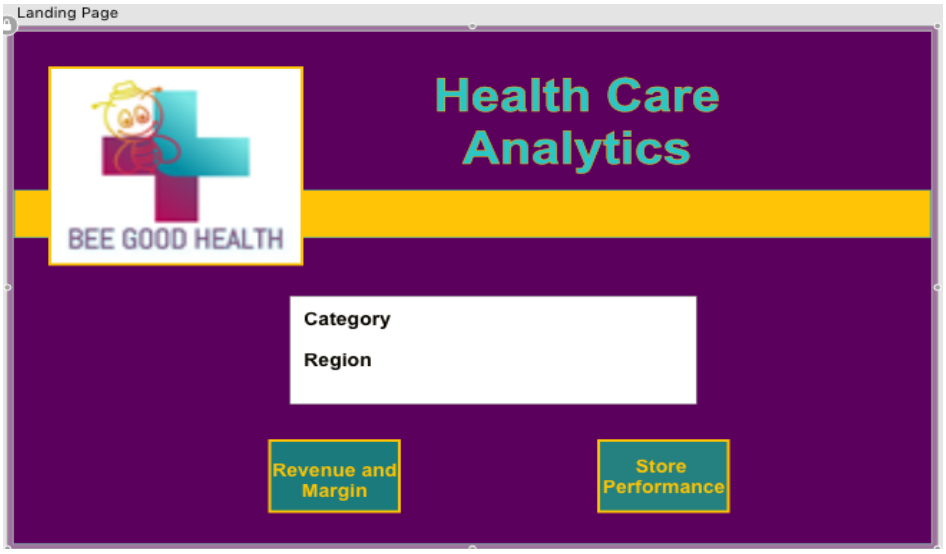
The example below provides an all-purpose landing page that guides users to the documents they need by tapping the buttons.



A simple landing page also provides end users with a quick application load time, helping to prevent user abandonment due to a frustratingly long wait for the app to open. While the user views the landing page, application data and document caches can begin downloading through pre-caching. This, again, reduces the time users spend waiting for a screen to render.

Exercise 2.4: Design the BGH app landing page

Using the pre-design guidelines you created in *Exercise 2.2: Create BGH's pre-design guidelines*, design an artboard for BGH's landing page. Below is a sample artboard of a landing page:



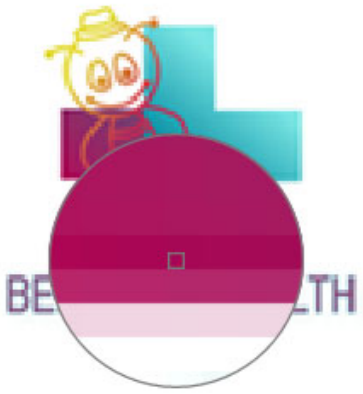
You can be creative with the design of the landing page, however, the artboard must include the following objects:

- The app's title, Health Care Analysis
- The BGH logo (supplied in the exercise files)
- Buttons for the Revenue and Margin Analysis document and Store Performance document
- Category and Region selectors

Create the landing page artboard

1. Return to **Adobe XD**.
2. Click the **Artboard** icon, then select **Android Mobile** as the device.
3. Change the orientation to **Landscape** and enable **Responsive Resize** to complete the setup.
4. Double-click the artboard name, **Android Mobile -1**, and type **BGH - Landing Page**.

5 Design the BGH landing page. Note that once you add the logo to your artboard, you can use the **Color Picker** to match BGH's corporate color palette based on the logo.



6 **Save** your work.

Prototype: Simulating user interaction

The fourth stage in the design thinking process is the prototype phase. Once the storyboard and wireframe have been created, your team should create a mobile app prototype. This provides a preliminary visual mockup that looks like the future result, and provides fundamental design and functionality. While wireframes and storyboards are a static representation of the app, the prototype simulates multiple states of the design. Typically, these do not contain any working code. The goal of a prototype is to simulate how users interact with the app as realistically as possible. This gives your app designers robust feedback, as users can test features like app navigability and see where users spend the most time. As the Mobile Architect, you should require your designers to create a prototype to avoid mistakes once the app is designed in MicroStrategy.

Exercise 2.5: Add interactivity with Prototype Mode

After designing your app screens, use Prototype Mode to connect the screens to each other to visualize how users interact with your app.

Link the BGH Landing page to the Revenue and Margin Analysis page

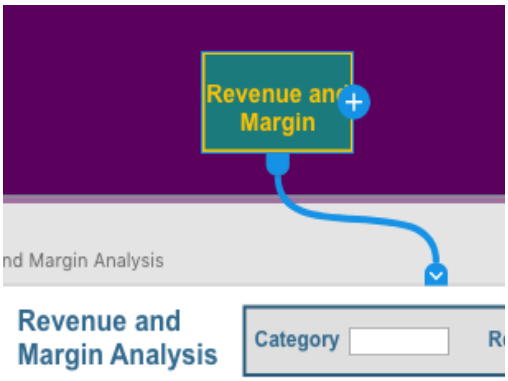
1 In Adobe XD, select the **Landing Page** artboard. Click **Prototype** in the top toolbar.

2 Click the **Home** icon to set the Landing Page as the home screen.

3 Select the **Revenue and Margin Analysis** button.

A blue tab is added. When you hover your mouse over the tab, the cursor changes to a connector.

4 Click and drag the **blue tab** to the **Revenue and Margin Analysis** artboard to link the button to the screen.

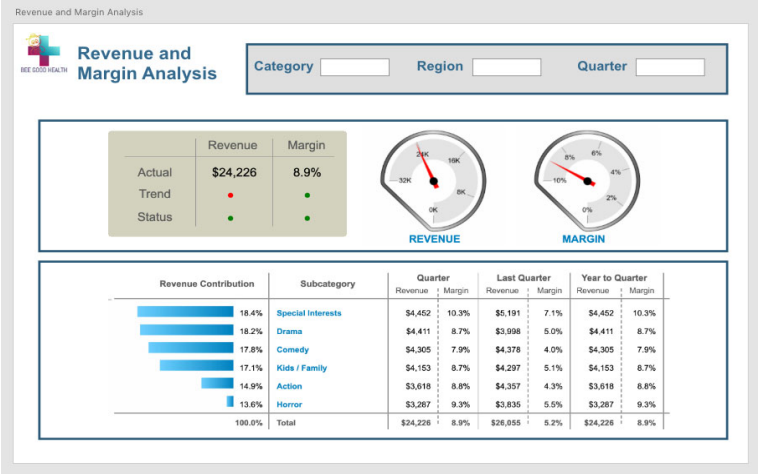


Add a navigation button to the Revenue and Margin Analysis artboard

1 From the Design tab, select the **Revenue and Margin Analysis** artboard.

2 Navigate to the Storyboard folder in the class exercise files, and locate the **BGH - logo.png** file.

3 Add the **BGH - logo** to the upper left corner of the Revenue and Margin Analysis artboard as shown below.



4 Click **Prototype** on the toolbar.

5 Connect the **BGH - logo** on the Revenue and Margin Analysis page to the Landing Page artboard.

6 **Save** your work.

If you created the Store Performance artboard as well, use the steps above to add the logo and connect it to the Landing Page artboard.

Open the Preview window to interact with your prototype

1 Click the **Desktop Preview** icon on the toolbar and test your prototype's interactivity.

Does the workflow make sense for end users? Is there any further interactivity you would want to add to the artboards?

3

Mobile Server Security and Configuration

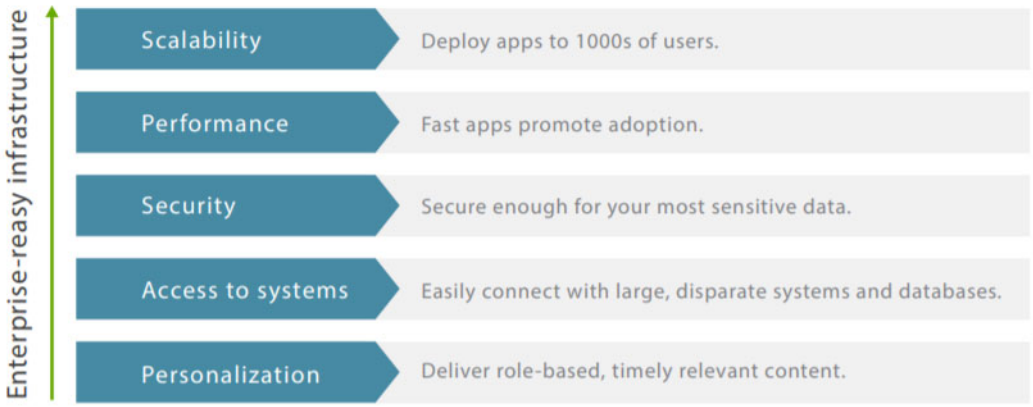
Before deploying enterprise applications, Intelligent Enterprises identify and implement strategies to run mobile solutions successfully without security, performance, or connectivity issues. As the Mobile Architect, you want to evaluate the Mobile Server, mobile app configurations, and security guidelines to ensure BGH is leveraging cutting-edge solutions that align with enterprise strategies.

In this chapter, we review:

- Mobile Server life cycle management, such as installation and upgrades. Regardless of the life cycle stage of the Mobile Server, the Mobile Architect continually optimizes and evaluates mobile architecture.
- Mobile security that enables the development of personalized and secured applications incorporating multi-factor, multi-layer authentication.
- Mobile configuration options, such as embedding a link in a custom app. The Mobile Architect evaluates and selects the appropriate configuration types for mobile app deployments to standardize configurations access across the enterprise.
- Configuring MicroStrategy Library and HyperMobile so business users and analysts can perform ad-hoc analytics on the go, supporting departmental analytics and mobility.

Enterprise-ready infrastructure: Mobile Server management

Organizations need enterprise-ready tools to ensure the success of their mobile applications. These tools need to be able to handle large, complex datasets stored across the enterprise, work when hundreds of thousands of users access the system, and dynamically adjust to unique user's business needs and preferences. They also need to ensure the security of an organization's sensitive information on broadly dispersed mobile devices.nThe ideal enterprise mobile deployment would be built on top of a truly enterprise-ready infrastructure that allows for:



As enterprises become more dependent on mobile technology and applications, it is imperative for the Mobile Architect to leverage consistent Mobile Server life cycle best practices for a coordinated implementation and continual upgrades.

Mobile Server life cycle

The Mobile Architect works with the System Administrator and Platform Administrator to follow server life cycle best practices, including the procurement, installation, maintenance, security, and upgrades of hardware, software, and network resources that the server needs to perform successfully.

Scalability

Scalability is at the core of a successful mobile app deployment. A true enterprise-grade mobile app platform has the flexibility to allow designers to build any app that they can imagine and the infrastructure to make sure that their apps can be easily deployed to any number of users across the enterprise.

The MicroStrategy Mobile Server can be scaled up through the use of clustering. A cluster is a group of two or more servers connected to behave like a single server. Each machine in the cluster runs the same services as other machines in the cluster, therefore any machine can stand in for any other machine in the cluster. Clustering provides many benefits, such as:

- Improved performance - Multiple machines provide greater processing power.
- Greater scalability - As your user base grows and report complexity increases, your resources can grow.
- Load balancing - Provides even distribution of MicroStrategy Web and MicroStrategy Mobile user sessions across the Web Server, Mobile Server, and/or Intelligence Servers.
- Failover support - If one node fails, another node in the cluster can pick up the workload. This prevents the loss of valuable time and information in the event of a failure.

For more information on clustering the MicroStrategy Mobile Server take the *Advanced Administration for Enterprise Mobility* course.

Increase Performance: Hardware capacity

Poor performance in a mobile app can be very frustrating, and lead to users abandoning the app and going back to the traditional means of accessing information. Mobile, more than any other interface, requires fast performance for adoption. Poor performance means a less valuable app experience for end users. Truly enterprise-grade mobile application platforms have the infrastructure to support mobile applications with lightning-fast performance.

An Intelligent Enterprise continually optimizes hardware capacity to maximize and take advantage of the MicroStrategy platform. Variables such as the following are factors that play an important role in Enterprise Mobility:

- CPU speed
- CPU type
- Operating system version
- Service upgrades
- File space
- Physical and swap memory

The Mobile Architect should work with the System Administrator to implement and continually upgrade enterprise hardware to make your deployment of MicroStrategy Mobile a successful one and continue to engage users. In addition to CPU hardware, the Mobile Architect should also ensure that any corporate mobile devices comply with the software requirements so users can properly interact with MicroStrategy Mobile on their devices.

Deployment and upgrades

An Intelligent Enterprise successfully coordinates their BI implementations, and the Mobile Server is a key part of this effort. To ensure that the enterprise has a smooth Mobile Server installation and subsequent version upgrades, you should work with the System Administrator and Platform Administrator to deploy and plan upgrades for the Mobile Server.

Installation and version upgrade

During the installation of the MicroStrategy analytics platform, the MicroStrategy Mobile Server is installed and configured. As the Mobile Architect, you oversee the installation and configuration processes of MicroStrategy Mobile.

For system requirements and specific instructions to install or upgrade MicroStrategy, see the *MicroStrategy Installation and Configuration Guide* and the *MicroStrategy Upgrade Guide*.

Best Practice

For upgrades, if you have created any mobile device configurations or saved any images from the MicroStrategy Photo Uploader widget, those configurations and images are deleted during the upgrade, so you should always manually back them up before the upgrade and restore them after the upgrade.

Upgrade components

The Mobile Architect is responsible for reviewing all components of the Mobile Server after an upgrade is complete. The Mobile Architect should:

1. Review and update Mobile Server and Intelligence Server connectivity, diagnostics, statistics, and security to facilitate the communication between mobile devices and the MicroStrategy platform.

2. Review and update Mobile configurations for device, connectivity, and home screen settings related to the Mobile Server upgrade to be applied in the mobile application.

Exercise 3.1: Download the MicroStrategy Mobile app

In this exercise, you search for and download the MicroStrategy Mobile app to your mobile device. Next, explore some of the capabilities of the sample apps located in the App Gallery.

Download MicroStrategy Mobile

1. On your device, download the **MicroStrategy Mobile** app from the Apple App Store or Google Play Store.
2. Take a few minutes to explore the sample apps in the App Gallery.

   By default, when the MicroStrategy Mobile app is downloaded, the application is connected to the MicroStrategy Demo Server, showing sample mobile apps. In the following exercise, you create the mobile configuration to link your mobile device to your environment with your enterprise data.

Access control: Mobile authentication and security

A secure Intelligent Enterprise effectively and efficiently controls who has access to specific resources, defines the conditions for access to those resources, ensures that the identities of all users trying to access them are accurate and true, and analyzes all these activities. In mobile environments, it is essential to govern, protect, and review assets, such as enterprise data.

Intelligent Enterprises leverage cutting-edge security resources, such as multi-factor, multi-layer authentication which go beyond simple passwords by combining something you have (such as a PKI certificate), something you know (such as a phone passcode), or something you are (such as Touch ID fingerprint verification).

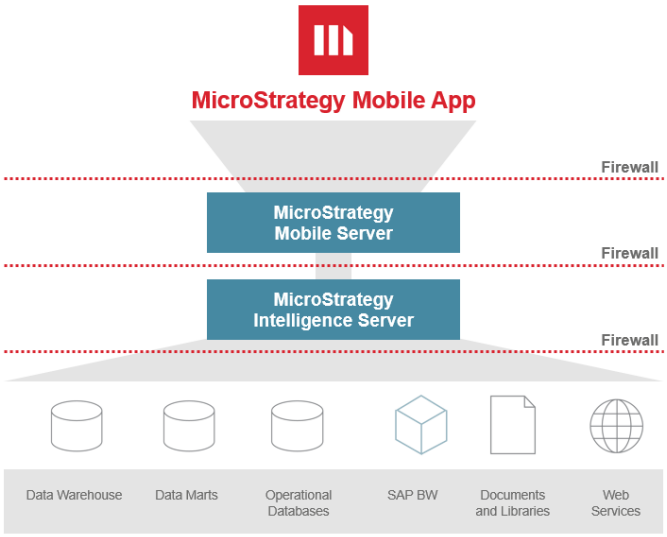
As the Mobile Architect, you are responsible for reviewing and strengthening the mobile security architecture to ensure your enterprise is tackling mobile specific risks, such as connectivity and data transmission. The sections below review how to establish security policies ranging from device access to enterprise application access.

Securing connectivity and communication

The core component of a MicroStrategy Mobile deployment is a secure Mobile Server. The Mobile Server facilitates communication between the Mobile clients and the Intelligence Server through a trusted connection.

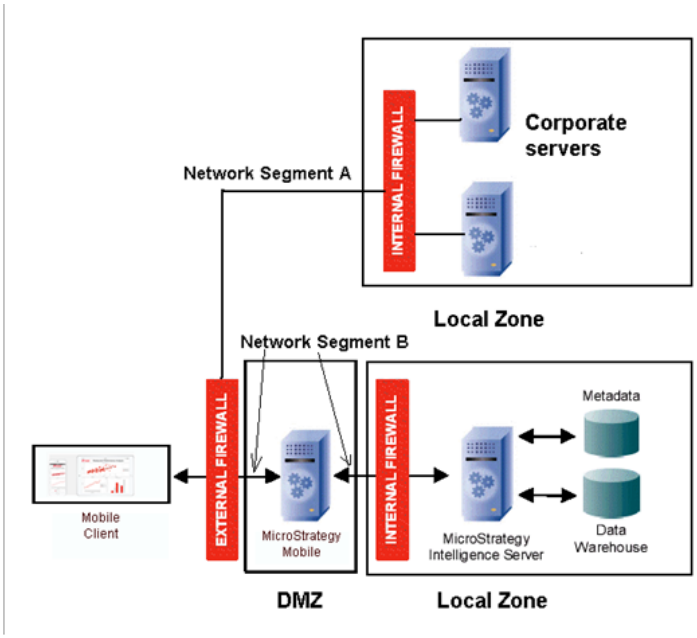
The Mobile Server coordinates data query requests with the Intelligence Server, taking results and embedding them into defined document files (along with configuration details, formatting, and images) that are distributed to mobile devices as specified. The Mobile Architect works with the Intelligence Center to coordinate a secure Mobile Server.

Securing communication across firewalls



Enterprises typically install the MicroStrategy platform on more than one server to distribute the workload. Secure communication across these servers is often governed by layers of firewalls constructed into Demilitarized Zones (DMZ). Using multiple firewalls, two distinct DMZs are created with one DMZ protecting the

Mobile and web servers. The second DMZ secures the infrastructure of the data sources and MicroStrategy Intelligence Server.



An effective DMZ is characterized not only by the presence of firewalls, but also an architectural component that accesses the database, which resides behind a firewall. The Intelligence Server is the core of MicroStrategy's Analytics Platform, and is the only component that has access to the database. It resides behind two firewalls.

Securing data transmission

Secure communication channels are important when it comes to data transfer. Data can be transferred by placing the Mobile Server behind a firewall and using a VPN (Virtual Private Network) connection to retrieve data for MicroStrategy Mobile apps, regardless of the transfer protocol or wireless network the device is connected to.

iOS and Android mobile devices integrate with several VPN technologies and protocols, including IPsec (Internet Protocol Security), L2TP (Layer 2 Tunneling Protocol), PPTP (Point-to-Point Tunneling Protocol), and SSH (Secure Shell). The implementation and setup of VPN is straightforward regardless of the corporate environment, and there are readily available extensions to existing corporate VPNs to support a compatible environment with Apple and Android devices. Setup can be automated, managed by an MDM (which we cover in a later chapter), or secured by a reverse proxy.

A reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client as if they originated from the web server itself.

The first line of defense: User authentication

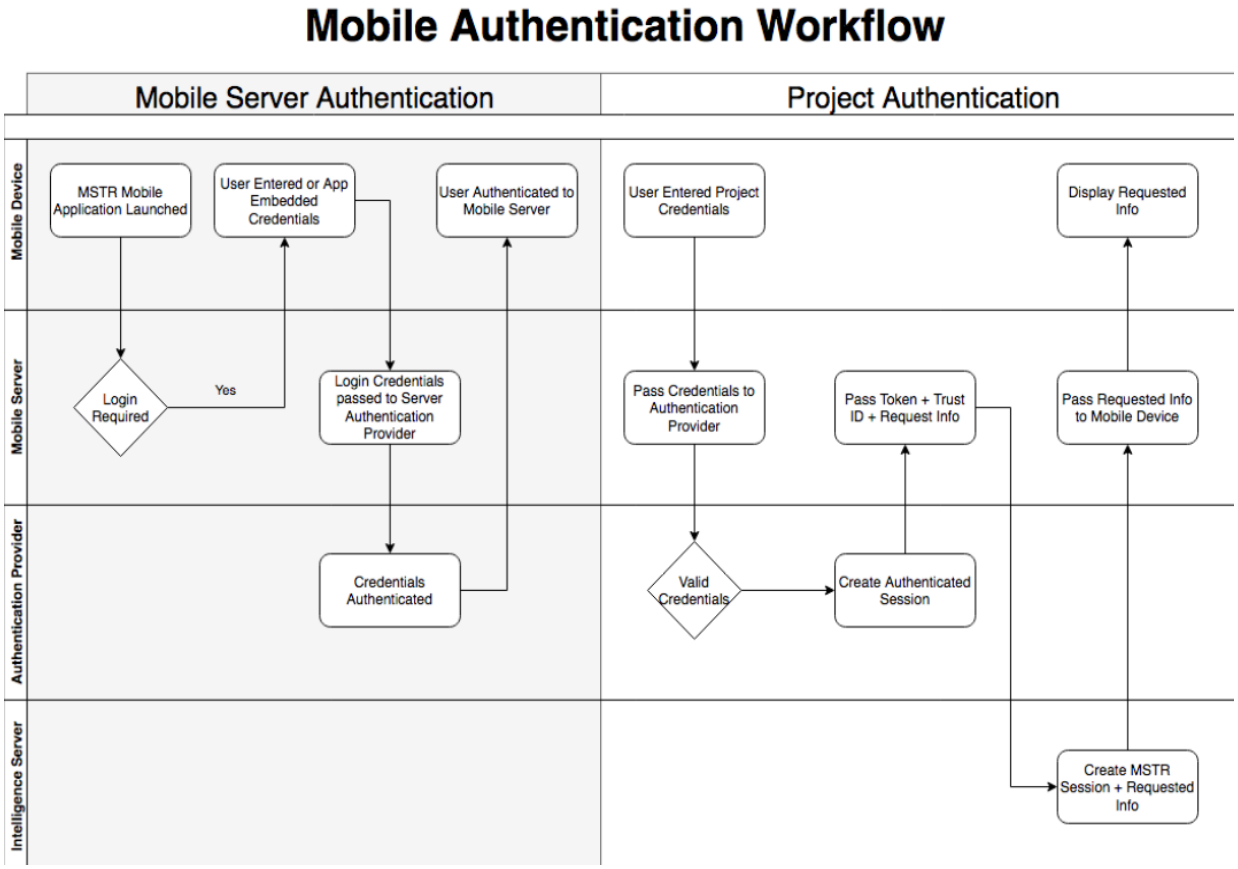
Intelligent Enterprises employ the defense in depth approach, using several layers of security throughout the IT system, including user authentication. The first line of defense in securing enterprise applications is user authentication. Choosing a user authentication experience requires balancing convenience versus security.

As the Mobile Architect, you should understand and implement the appropriate authentication methods to restrict app access to only the appropriate users, working with the enterprise's security platform and choice of identity management system. Both the out-of-box MicroStrategy Mobile app and custom apps that embed the MicroStrategy Mobile framework must log into MicroStrategy and be authenticated before they can access data.

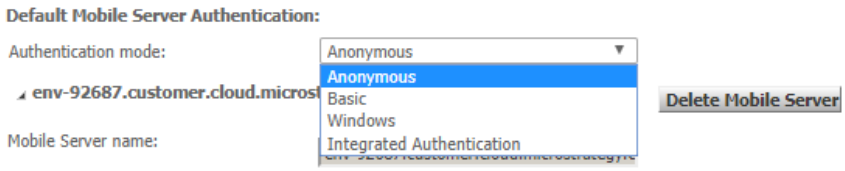
The Mobile Architect ensures that the appropriate authentication type is selected when creating a mobile configuration. There are two different types of authentication performed by the MicroStrategy mobile app: Server and Project.

- **Server authentication** - validates users when they log into the application server, for example, Tomcat or IIS.

- **Project authentication** - validates users when they log in to a project in the MicroStrategy metadata.



Mobile Server authentication



MicroStrategy Mobile supports the following server authentication modes:

- **Anonymous**: Anonymous authentication gives users access to the Mobile Server without prompting them for a user name or password. Anonymous may be required to enable other authentication modes, such as database warehouse or LDAP.
- **Basic**: When using Basic authentication, MicroStrategy Mobile app users are required to provide a valid login credentials to access content. You can use Basic authentication even with firewalls and proxy servers.

Best Practice

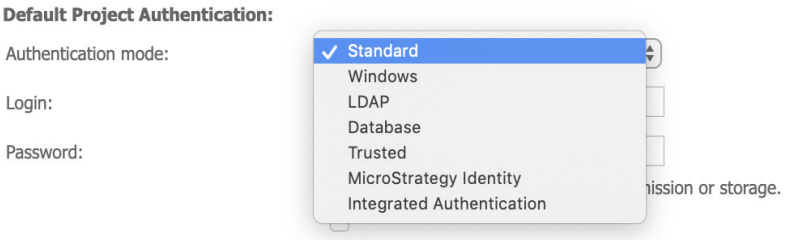
As Basic authentication transmits unencrypted password across the network, it should be used when the connection between the client and the server is secure such as when establishing a connection either over a dedicated line or when using Secure Sockets Layer (SSL) encryption

- **Windows**: When using Windows authentication, the MicroStrategy Mobile app users are not prompted for login credentials. Instead, the MicroStrategy users are linked to a Windows user account and are automatically authenticated by the Windows operating system.

  If using Windows or Basic authentication, you must disable Anonymous authentication. In addition, you need to provide login credentials in the username and password boxes in the mobile configuration.

- **Integrated Authentication**: The Mobile Server authenticates the user against the Kerberos server.

Project authentication



Listed below are the Project authentication types for MicroStrategy Mobile.

- **Standard**: Intelligence Server is the authentication authority. This is the default authentication mode. When standard authentication is used against the MicroStrategy Intelligence Server, users' group memberships are simple to handle because there is only one source for group membership information: the MicroStrategy metadata. The MicroStrategy user and group editors accurately reflect the user and group relationships that exist in the metadata.

Best Practice

- **Anonymous**: Users log in as “Guest” and do not need to provide a password. This authentication mode should only be used with public data or a demo app. For example, a public health app that allows all users to access readily available Centers for Medicare and Medicaid data to compare cancer rates among different states.

- **Database**: A database warehouse is the authentication authority. To set up warehouse authentication, the Mobile Server must be configured to anonymous authentication.

- **LDAP (lightweight directory access protocol)**: An LDAP server is the authentication authority. LDAP introduces another source in addition to the MicroStrategy metadata, the LDAP server, of user and group membership data. Since there are two user and group repositories, conflicts may arise in representation in the MicroStrategy interface and application of user privileges within a user session. By using the LDAP import process, synchronizing users and user groups reduces administrative efforts.

  If users cannot log into Intelligence Server with LDAP, work with the Platform Administrator and Services Architect to answer the following questions. These questions serve as tools to help focus and identify root causes of the problem.

  - Are the LDAP-SDK dynamic-link libraries installed on the Intelligence Server machine?
  - Do the LDAP-SDK dynamic-link libraries reside in the correct system path?
  - Does the certificate reside on the Intelligence Server machine in the correct system path?
  - Do the LDAP server-side logs show success messages?
  - Is the certificate available on the MicroStrategy Intelligence Server machine in the correct path specified in the Intelligence Server configuration?

- **Trusted**: If you select trusted authentication, the login and password fields are disabled. To use trusted authentication, you must supply your trusted authentication provider credentials in the Mobile Server authentication section above.

- **Integrated and Windows Authentication**: Integrated authentication encompasses several different third-party authentication methods, including:

  - Windows authentication: Windows is the authentication authority.
  - Integrated authentication: A domain controller using Kerberos authentication is the authentication authority. Kerberos supports both Linux and Windows architecture.
  - Third-party authentication: A third-party Single Sign-on tool, such as IBM Tivoli Access Manager, CA SiteMinder, or Oracle Access Manager, is the authentication authority.

    One example of a SSO challenge is with enterprises that employs a bot detection program, such as Distil. In one enterprise, the bot detection technology captured API service calls from the app to the Mobile Server, attempting to inject CAPTCHA where there was no user interface to

respond to the challenge-response test. This resulted in the SSO credentials to be rejected.

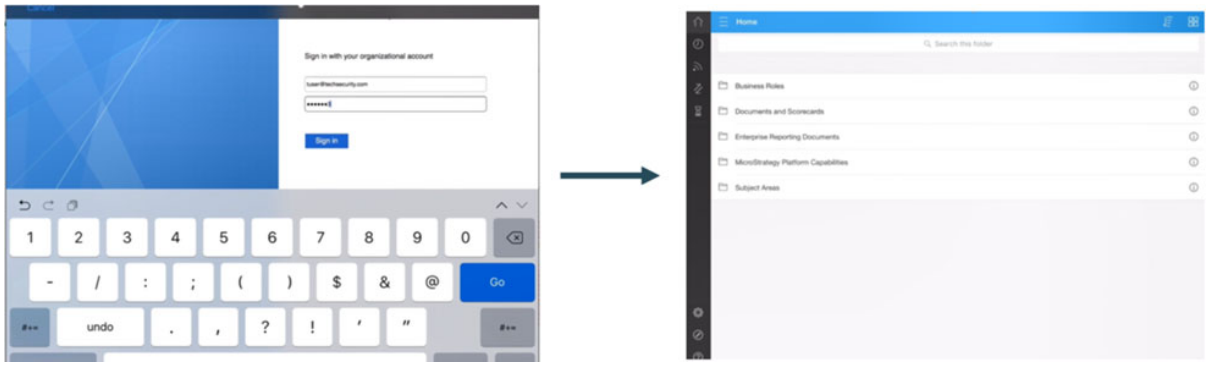
The bot detection technology was disabled for any interactions between the mobile app and the Mobile Server, allowing the app to pass the SSO credentials to the Mobile Server.

Evaluating the authentication types, which would you use for Bee Good Health? Why?

Best Practice

Seamless authentication: Single Sign-on

The BGH Intelligence Center has decided to implement a seamless authentication experience for enterprise app users by implementing SSO. As the Mobile Architect, you work with the other architects to integrate MicroStrategy Mobile with third party SSO tools including Tivoli, SiteMinder, Oblix, and Okta.



With SSO, users can access enterprise apps with a single authentication, such as their corporate credentials, entered when they initially connect to the app. SSO can help drive adoption of apps across the Intelligent Enterprise, enhancing user experience with fewer sign-on prompts and better security.

The following steps are for reference only, and are not intended to be performed in class.

How to plan a SSO solution

1. Analyze the current Single Sign-on software set up at the enterprise.
2. Determine dependencies to enable authentication with SAML, OAuth, or a vendor-specific implementation such as Windows.

SAML SSO transfers the user's identify from one place to another, for example, when a user logs into their intranet, they do not need to enter their username and password to access their email.

O-Auth does not deal with authentication but rather authorizes resources. For example, a user logs into one application with a set of credentials. These same credentials can then be used to log into related websites or apps.

3. Work with the Services Architect to develop the SSO integration solution.

4. Use the MicroStrategy Mobile Administrator page to create a trust relationship. A trusted relationship enables the authentication token to pass from the trust authentication provider to the Intelligence Server via MicroStrategy Mobile, allowing user permissions to be retrieved from a directory, such as LDAP, and grant access to the MicroStrategy application.

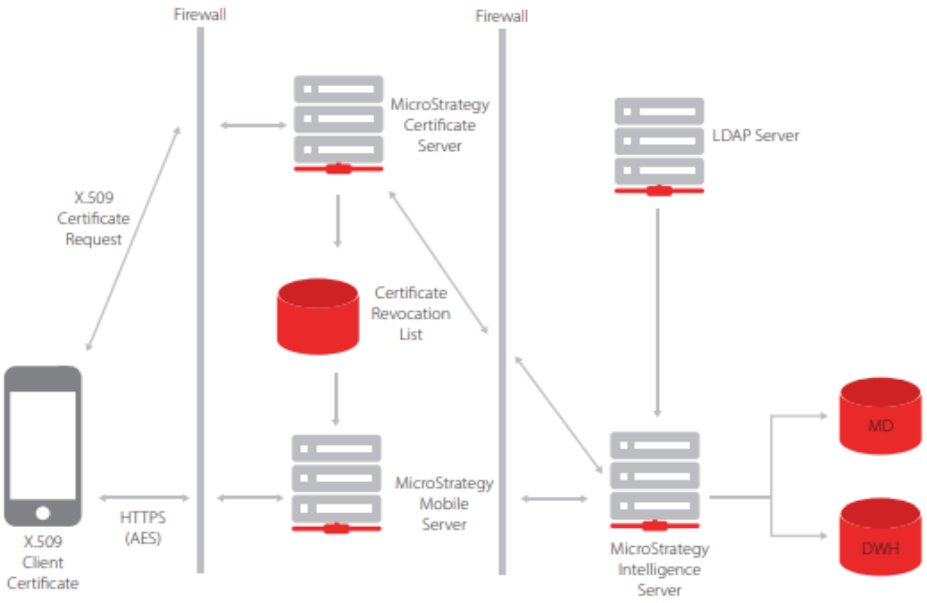
5. Deploy the SSO solution across the production environment by enabling Single Sign-on authentication for MicroStrategy Mobile.

Mutual (two-way) authentication

HTTPS Mutual authentication, also known as two-way authentication, is facilitated by the addition of a new server component called the MicroStrategy Certificate Server. The Certificate Server has the sole purpose of ensuring added security by providing mobile devices with specific certificates that are later used in authentication. As the Mobile Architect, you should ensure users are enrolled with the MicroStrategy Certificate Sever.

To gain access to the Mobile Server, the user must first enroll the device with the MicroStrategy Certificate Server. The user's credentials are presented to the Certificate Server and, upon validation, the Certificate Server issues an X.509 certificate that is sent to the device and stored. The image below shows how

MicroStrategy components interact to facilitate HTTPS communication with mutual authentication incorporating the Certificate server.



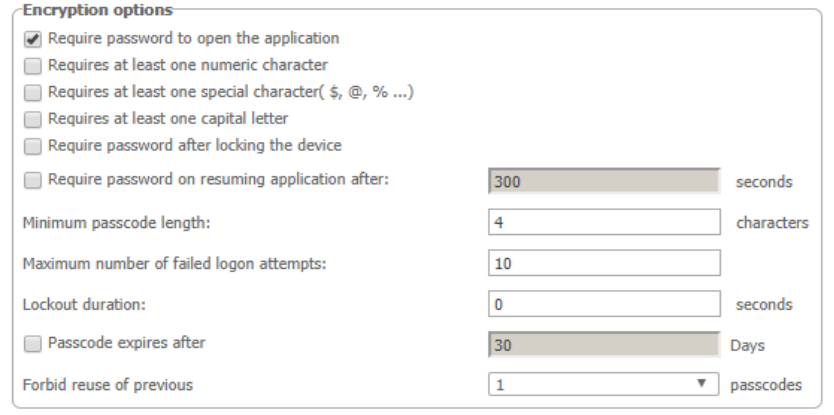
App-level security

The Mobile Architect should also require that app-level security is applied to enterprise apps. MicroStrategy Mobile also supports app-level passcode and Touch ID access for iOS. When configured, the app passcode or Touch ID (depending on the device and operating system) is required before a user can access information in the MicroStrategy Mobile app.

App level passcode: The user must define a passcode based upon configurable requirements such as:

- At least one numeric character
- At least one special character
- At least one capital letter
- Minimum password length
- Maximum number of failed logon attempts
- Lockout duration

The passcode encryption options are set in the Settings tab of the Mobile Configuration, which we discuss in the next section.



Establish connectivity: Mobile configuration

A Mobile configuration lets you establish connectivity between a mobile device and the MicroStrategy Mobile Server. It ensures that local settings on devices are correct, secure, and compliant with organizational policies. The Mobile Architect should work with the Platform Administrator to coordinate and standardize mobile configurations to support the mobile connectivity needs of the Intelligent Enterprise. For example, you should specify how often the configuration should check for new subscriptions in the Intelligence Server, or if device caches should be validated.

Exercise 3.2: Define a new Mobile configuration

Best Practice

In the previous chapter, you created a storyboard for the BGH mobile app. To assist in the app development and design process, it is best practice to view the app periodically on your device, so you can make adjustments throughout the development process and ensure that all the objects are visible and accessible on the device.

Open the Mobile Administration page

1. Open the Welcome to MicroStrategy landing page.
2. Under More Resources, hover over **MicroStrategy Mobile Administration**, and click **Configure**.

3 Login with the credentials you received in the MicroStrategy cloud email.

The Mobile Administration page opens.

4 In the left pane, under Mobile Server, select **Mobile Configuration**.

Define a new Mobile configuration

1 Click **Define New Configuration**, and select your device type from the drop-down list.

The mobile configuration page opens with the Settings tab displayed.

2 In the Configuration Name box type **BGH - *YourDeviceType***.

For example, if you have an iPhone, you would type BGH - iPhone.

Configure device settings

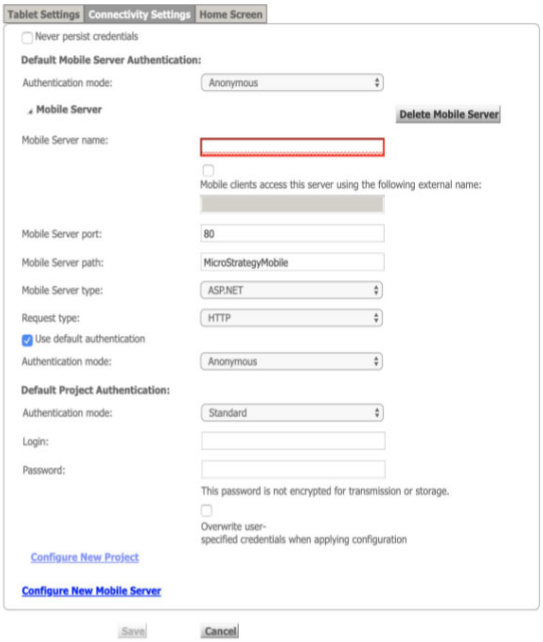
3 On the Settings tab, select the check box next to the following options:

- Automatically pre-load subscriptions
- Clear caches when application closes
- Enable push notification

Configure Connectivity settings

4 Select the **Connectivity Settings** tab.

5 Click **Configure New Mobile Server**.



6 Under **Default Mobile Server Authentication**, provide all the necessary information as described below:

- **Mobile Server name**, enter the following server name:

  env-XXXXX.customer.cloud.microstrategy.com

  where **XXXXX** is the environment number.

- **Mobile Server port**, enter **443**.
- **Mobile Server type**, select **J2EE** from the drop-down list.
- **Request Type**, select **HTTPS**.
- **Authentication mode**, select **Standard**.
- For **Login** and **Password**, enter your credentials you received in your MicroStrategy cloud email.

  In a Production environment, you do not embed user credentials. Users are prompted to login the first time they launch the app. The app can retain the credentials for future launches.

7 Under Default Project Authentication, click **Configure New Project**. The bottom of the page expands.

8 For **Project Name**, select **MicroStrategy Tutorial**.

Set the app's Home Screen

9 Select the **Home Screen** tab.

10 Select **Display the contents of a folder**, and click the **Browse** button.

11 Login with the credentials from your MicroStrategy cloud email.

12 Select **Shared Reports** from the drop-down list, and click **Current Folder**.

13 Click **Save**.

Connect to enterprise data: Configuration strategies

In the previous exercise, you defined a configuration profile for your specific device. However, your mobile app is still pointing to the MicroStrategy demo data.

The following are configuration options:

- Generate a specific URL link and distribute it to users in an email or on a web page.
- Embed the configuration file's content in the mobile app.

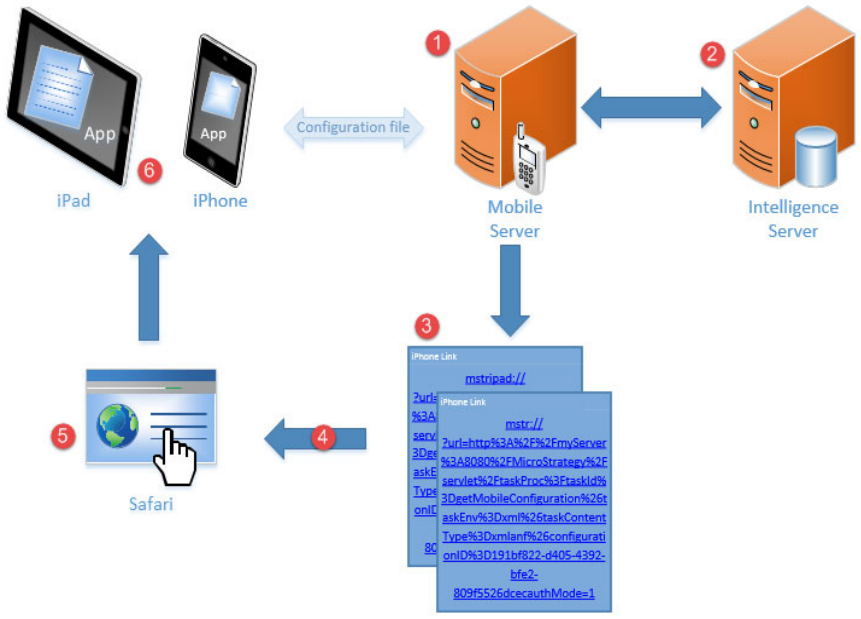
As the Mobile Architect, you should evaluate which configuration strategy should be employed for potential deployment situations. In the sections below, we examine both options and discuss the pros and considerations of each, so that you can make the best decision for your Intelligent Enterprise.

Tap to configure: Generate a configuration URL

A configuration URL can be used to trigger a dialog between the mobile app and the Mobile Server. Using the Mobile Administration page, the configuration URL link can be generated for each Mobile configuration. Once the URL is generated, it can be distributed to users by email or as a link on a web page.

When users tap the URL from their mobile device, the mobile app configures itself to the Mobile Server. Once the process is complete, the mobile app can access the projects available to the Mobile Server and remembers this configuration at each launch.

The process of configuring the Mobile app using a configuration URL is shown in the diagram below for both iOS and Android devices.



This configuration method can also be used to update an existing mobile application. This avoids developers having to embed a new configuration file and force an update of the mobile application.

Pros

- Users are not forced to update their app.
- Simple to do, no programming skills required.
- Link can be distributed quickly.
- Can easily revert to previous settings using the previous URL.

Best Practice

- Can be set to automatically check the Mobile Server for changes to the configuration, and update those changes to the app.
- Configurations can be deployed and updated with an Enterprise Mobility Management (EMM) solution using the App Config framework.

Considerations

- Less secure, as the link can be read.

Exercise 3.3: Generate a configuration URL

To configure your MicroStrategy Mobile app to your Mobile Server, generate a mobile configuration URL link, send it to an email accessible from your mobile device, and tap.

Generate a URL for a mobile configuration

1. On the Mobile Server, access the **Mobile Configuration** page.
2. Locate the **BGH - *YourDeviceType*** configuration name.
3. Click the **Generate URL** icon located in the **Actions** column as shown below.

4. On the Generate Configuration URL window, the **Server Name** field should be automatically populated.

   If this name is not automatically filled in, type the IP address or a domain name of the machine listed in the URL of the MicroStrategy Cloud landing page. Example: env-23638.customer.cloud.microstrategy.com.

5. Select the **Include port** check box, and type **443** in the text box.
6. In the **Request type** drop-down list, select **HTTPS**.
7. In the **Authentication Mode** drop-down list, select **Anonymous.**

   When the Mobile user taps the URL on the mobile device, the user may be prompted to provide a user name and password to log in to MicroStrategy Mobile Server using this authentication mode. For this class, this is the user ID and password provided by your instructor.

   Due to a requirement of the Android operating system, a short URL must be created when sending the configuration URL to users. To generate a short URL for Android, select **Use short URL**.

8. Click **Generate URL**.

   The URL is displayed in the box below the Generate URL button.

9. Copy the link to send in an email you have access to from your mobile device.

10 Click **Save** to save the URL settings if they were updated.

The Generate Configuration URL window closes, and the authentication mode and host are saved for the next time a URL is generated for this configuration.

Configure MicroStrategy Mobile

1 Open the email with the generated URL on your mobile device.

2 Tap the link or paste the URL in a browser on your device to configure the MicroStrategy app.

Your MicroStrategy Mobile app is now configured to your Mobile Server.

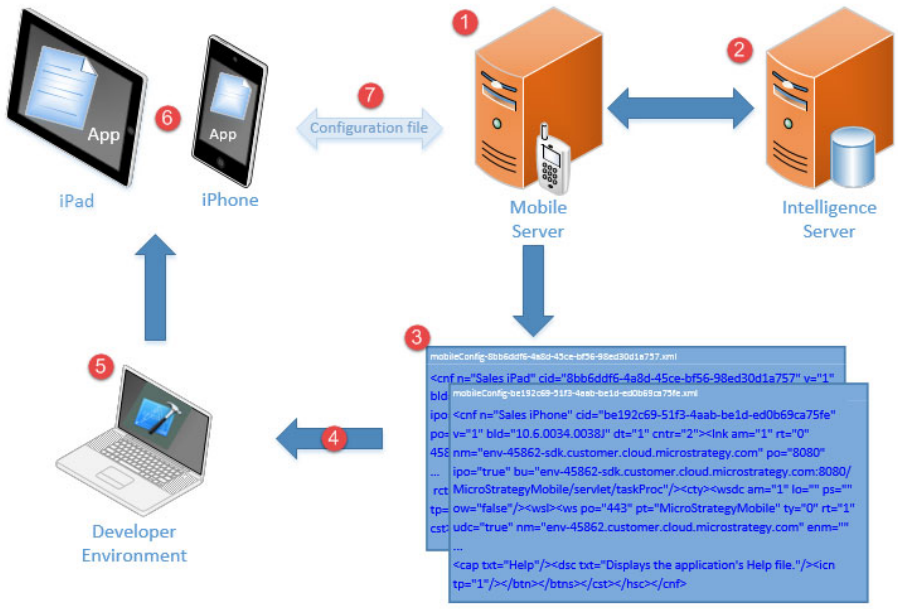
Pre-configure the app: Embed the configuration

The second method of configuring the MicroStrategy Mobile app removes any participation from the end user. When a mobile configuration is created, the settings are saved in an .xml file named mobileConfig-<Identifier>.xml, where <Identifier> is a Global unique Identifier (GUID) composed of 30 hexadecimal characters, as shown below:

mobileConfig-902d5e49-c11a-44b6-aa72-109fa83b5858.xml

The contents of the xml file are embedded in the mobile app, and becomes part of the mobile application. When the user downloads the app, it is already configured to your enterprise's Mobile Server. To change connectivity settings once the app is downloaded, either a new version of the mobile app can be posted to the app store, or an updated URL link can be given to users so they can update the app.

Below is an illustration of the process.



Pre-configuring the mobile app can be done through the use of MicroStrategy Mobile SDK. For iOS devices, the contents of the mobileConfig.xml file can be copied, and placed in the Preferences.xml file in the Xcode project for the appropriate device type. This method is covered in *SDK for Customizing iOS Applications*.

For Android devices, you must generate the appropriate JSON and text files through the getMobileConfiguration task to add the configuration to the MicroStrategy Mobile app. This allows the app to use the defined configuration you created, and not the default one. This method is covered in *SDK for Customizing Android Applications*.

Best Practice

As a best practice, do not embed credentials in the mobile configuration, except for testing purposes. In a production environment, users should have to enter their credentials once. After that, the application can remember them.

Pros

- Every user receives the change as the app is updated.
- Easier to force users to update the app.
- Simpler to support as everyone is on the same code and configuration.
- More secure as the configuration is harder to locate.

Considerations

- All users are using the same configuration, so it is harder to make on the fly adjustments for special circumstances
- Some programming skills are involved.
- Longer to implement and distribute.
- Harder to revert to previous settings, need to re-deploy the entire app. However, this can be avoided if the configuration is set to automatically check for updates.

***Poll**: True or false, a mobile configuration can contain connectivity information for multiple Mobile Servers.*

Dossiers on the go: Library Mobile

MicroStrategy Library is a web-based application that allows users to view, present, and analyze dossiers or documents added to their Library. The Library Mobile app empowers analysts to perform ad-hoc analysis while away from the office. This is especially important at BGH, since consultants are constantly traveling to hospitals and presenting analysis in dossiers.

To support departmental analytics and mobility, the Mobile Architect ensures that end users are able to access their Library from their mobile devices. You can also use Enterprise Mobility Management (EMM) software to deploy Library Mobile.

Setting Library Mobile home screen to a document or dossier

When users launch Library on their mobile device, the app opens to their entire Library. In MicroStrategy 2021, a document or dossier can be configured as the home screen for Library Mobile. This allows organizations to provide a more customized, branded experience for their users. For example, a Library landing page can be used to limit the scope to specific dossiers or documents, instead of users viewing all of the applications in the system.

Exercise 3.4: Download and configure Library Mobile

In this exercise, you download the MicroStrategy Library Mobile app to your device and configure the app to your Library. This same process can be used to configure Library Mobile for users in your organization.

Download the Library Mobile app

1 From the Apple App Store (iOS) or Google Play (Android), search for and download the **MicroStrategy Library app** on your mobile device.

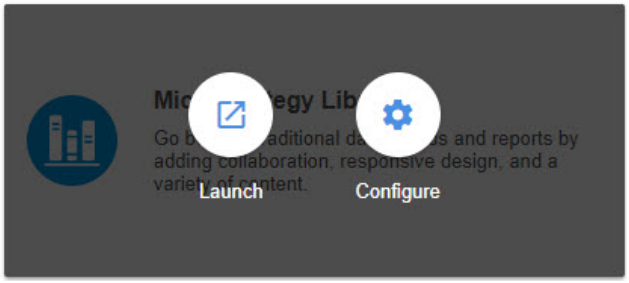
2 Take a few minutes to open the app and explore some of the sample dossiers.

Capture and send the configuration link

1 From the Welcome to MicroStrategy email, click **Access MicroStrategy Platform**.

2 Log in with the provided credentials.

3 Under More Resources, hover over **MicroStrategy Library** and click **Configure**.

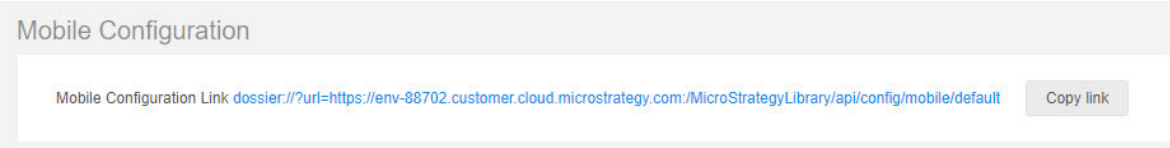


4 Log in with your credentials.

You are now on the Library Administration page.

5 Click **Library Server** on the left.

6 Under Mobile Configuration, click **Copy Link** to copy the mobile configuration link.

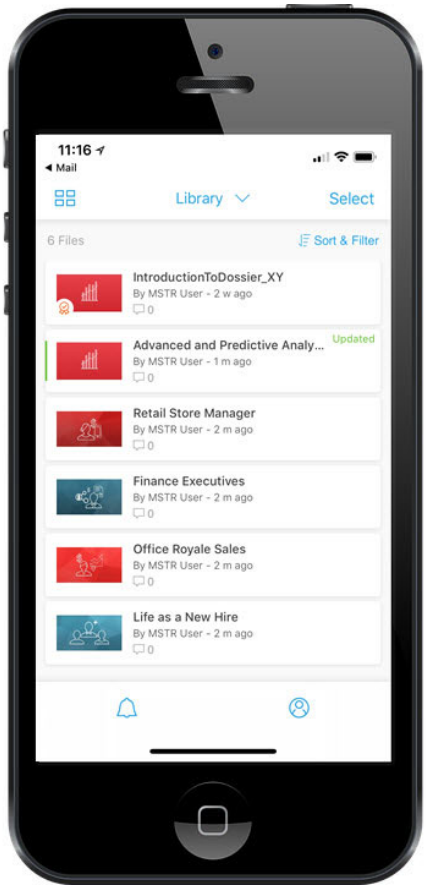


Alternatively, the mobile configuration link for Library is also available in Workstation. To access it right-click on your connected environment, and select Properties.

7 Paste the link in an email and send the email to an address you can access on your mobile device.

8 Open the email in your mobile device and tap the link.

9 Log in with your credentials. Your app is now configured to your Library.



4

Create an Enterprise Mobile Solution

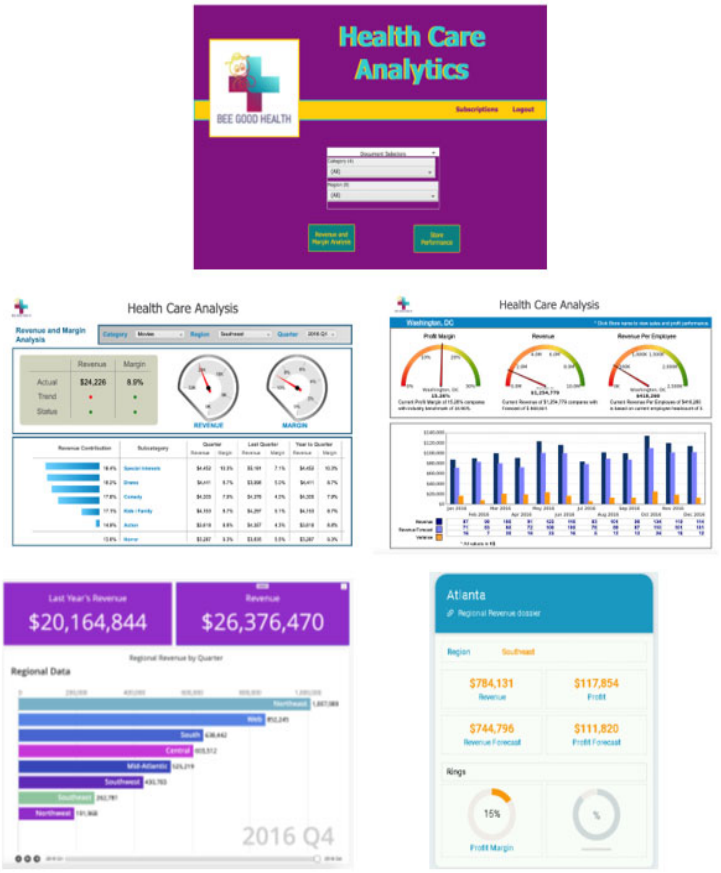
BGH's Enterprise Mobile solution

Now that you have completed the first few stages of the design thinking process and have your mobile configurations in place, it is time to build your enterprise mobile solution within MicroStrategy.

In this chapter, you create an enterprise mobile solution for BGH with the following specifications:

- A simple data light landing page that allows for easy navigation and the ability to filter based on the region and category attributes.
- Leverage pre-existing documents, and combine them using a multi-layout document.
- One application optimized for both phones and tablets by using Mobile views.
- A custom visualization that compares revenue vs last year's revenue per region.
- Dossiers available and optimized for Library Mobile.
- HyperMobile cards to view important KPIs on the go.

The below image shows the complete mobile solution that you create for BGH.



Exercise 4.1: Create the BGH landing page

In this exercise, you create the landing page for the BGH app. It serves as a starting point for the app with links to the needed documents, other pages such as the user's MicroStrategy Subscriptions, and filters to pass to the documents.

Access MicroStrategy Tutorial

1. In the Welcome to MicroStrategy email, click **Access MicroStrategy Platform**.
2. In the **User Name** and **Password** boxes, type (or copy and paste) the login credentials provided in the MicroStrategy email. Click **Login**. The landing page opens.
3. On the landing page, hover over **MicroStrategy Web** and click **Launch**.

4 Click **Go to MicroStrategy Web**.

Create the BGH landing page

1 In the Shared Reports folder, click Create, and select **New Folder**.

2 Name the folder **BGH** and click **OK**. This folder is used to hold all of the objects for the BGH mobile app.

3 Select the **BGH** folder to open it.

4 From Create, point to **New Document**, and select **01 Blank Dashboard**.

5 Right-click the new panel stack and select **Properties and Formatting**.

6 In the Properties and Formatting window, select **General** in the list of properties on the left and change the name to **Main Panel stack**.

7 Click **OK**.

8 Design your Landing Page according to your artboard you created in Adobe XD.

9 **Save** your document in your BGH folder as **BGH - Landing page**.

Add a filter panel

1 Click **Insert**, then select **Filter Panel**.

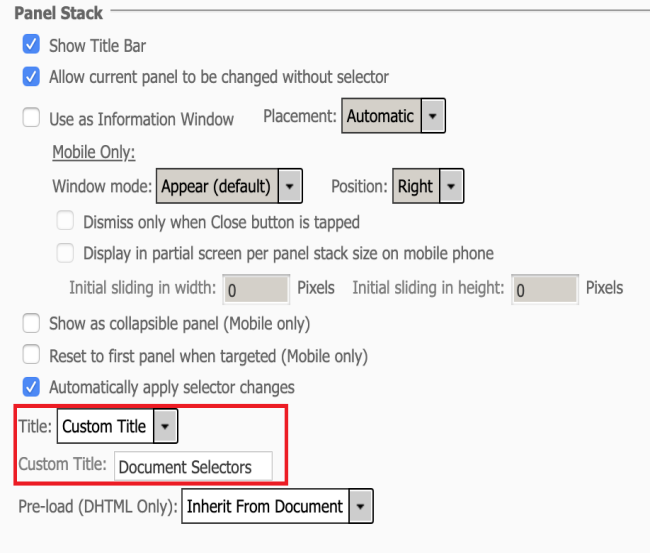
2 Use your cursor to draw the filter panel on your Landing Page.

3 Right-click the filter panel and select **Properties and Formatting**.

4 Select **General** from the list of properties on the left.

5 From the **Title** drop-down list, select **Custom Title**.

6 Type **Document Selectors** in the **Custom Title** text box.



7 Click **OK**.

8 Click the arrow next to Add Selector and select **Drop-down**.

9 To add a second selector, click **Insert**, point to **Selector**, then select **Drop-down**.

10 Draw the selector in the Filter Panel.

The selectors are configured in a later exercise.

11 **Save** your document.

Optimizing documents for mobile devices

Optimizing the design of a dashboard-style document for mobile devices requires the use of device-specific document features. MicroStrategy provides many tools to aid in the production of documents optimized for mobile such as:

- Pre-designed mobile document templates
- Widgets
- Mobile views
- Information Windows
- Responsive design using Fit Page or Fit Width

Pre-designed document templates

A document template allows you to start with a pre-defined structure when creating a new document. MicroStrategy provides several pre-designed templates for mobile documents that are correctly sized for the device's screen. The following templates are optimized for display on iOS and Android mobile devices:

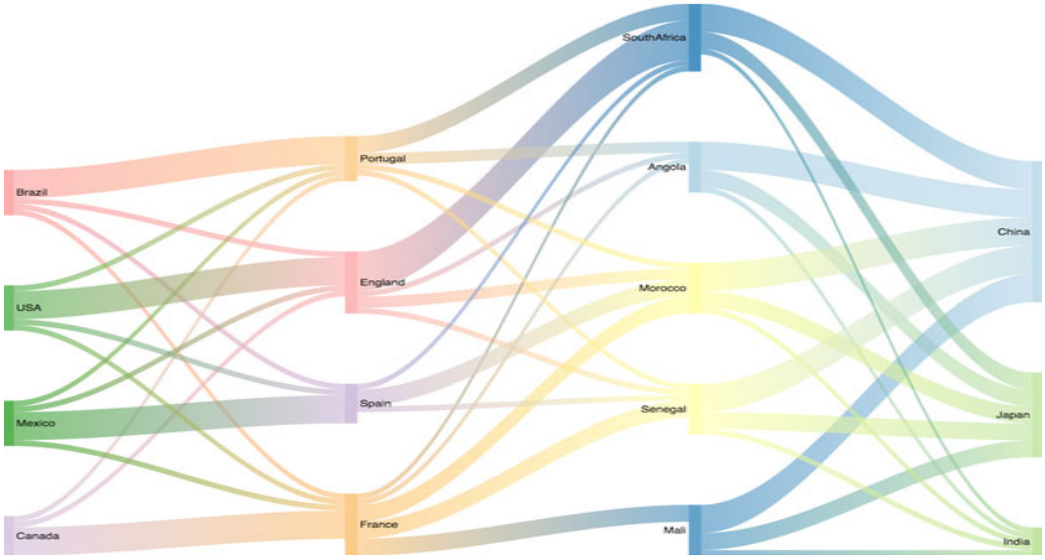
- iPhone Portrait - Designed to be viewed on an iPhone or Android smart phone held in a vertical position.
- iPhone Portrait Micro-Application - Same as the iPhone Portrait, except that the Optimize layout for micro application option is selected, preventing users from performing actions such as zooming in or out of the document. This allows you to better control the user's experience and interaction with the document.
- iPhone Landscape - Designed to be viewed on an iPhone or Android smart phone held in a horizontal position.
- iPhone Landscape Micro-Application - Same as the iPhone Landscape, except that the Optimize layout for micro application option is selected, preventing users from performing actions such as zooming in or out of the document.
- iPad Portrait - Designed to be viewed on an iPad or Android tablet held in a vertical position. Only one section of the document is displayed.
- iPad Landscape - Designed to be viewed on an iPad or Android tablet held in a horizontal position. Only one section of the document is displayed.
- Navigation for iPad - Designed to create a navigation document for iPads.
- Navigation for iPhone - Designed to create a navigation document for iPhones.

By default, some of these templates may be hidden from the document template list. They can be enabled from the document template folder in Developer.

Mobile Widgets

Widgets are sophisticated visualizations that use rich interactivity to enable users to understand their data more effectively. MicroStrategy provides a variety of widgets, such as the Data Cloud, Heat Map, Interactive Grid, or Map widget. Additional widgets can be imported as a custom visualization, and can be used on a document or dossier. You can define a Grid/Graph to display as a widget on a

document when executed on a mobile device. The image below displays a sample custom visualization.



Formatting documents: Different screen sizes and orientations

Mobile views

One approach to designing a mobile application is creating a unified design where one app can be used across multiple devices. You can quickly and easily determine how elements are displayed on documents for both iOS and Android devices. For example, you can resize a graph to take advantage of the extra horizontal space when held in landscape orientation, or rearrange the controls on the document to accommodate the extra vertical space when the mobile device is held in portrait orientation. When using Mobile views on your document, consider what devices the app is designed for and how the app looks and functions on each device.

Once you have added a Mobile view to a document, you can display a preview of each view in Design Mode or Editable Mode in Web. There you can determine whether a control is visible when that view is displayed on a mobile device.

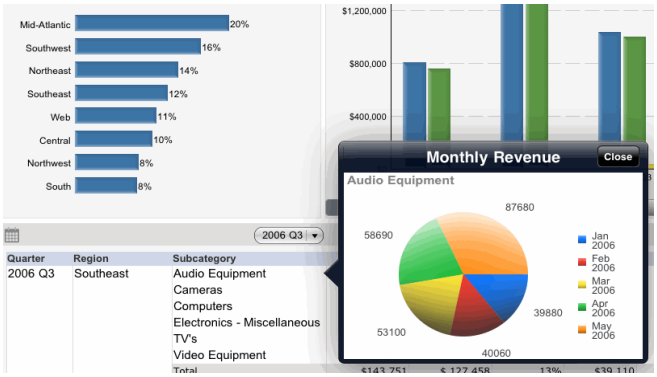
Using Mobile views with multi-layout documents

Documents can contain multiple layouts, with each layout containing its own individual document. Each layout in a document can be defined to display a document sized for an iPhone, iPad, or Android device independently of the other

layouts within the same document. When you create a Mobile View, it is automatically available to every layout in the document. For example, you have a multi-layout document with three layouts, if you create a Mobile View to determine how the document is shown on an iPhone, you must edit the controls in each layout to define how the layouts are displayed. You can use the Orientation option for Mobile Views in conjunction with the Supported Orientation option for document layouts to determine how a mobile device chooses the best Mobile View to use to display a document layout.

Information Windows

Information Windows allow users to view additional information about an attribute element by tapping the element in a grid or graph. The Information Window opens over the element, displaying the additional information.



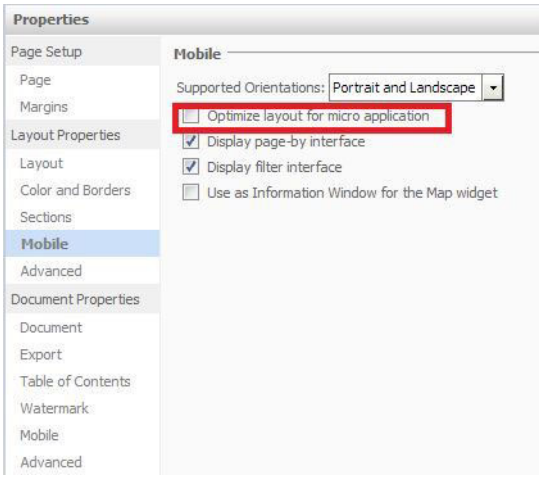
The following controls can be added to an Information Window:

- Images, such as a corporate logo or button
- Data fields
- Links to mobile device applications, such as a link to call a phone number
- Links to other documents or reports

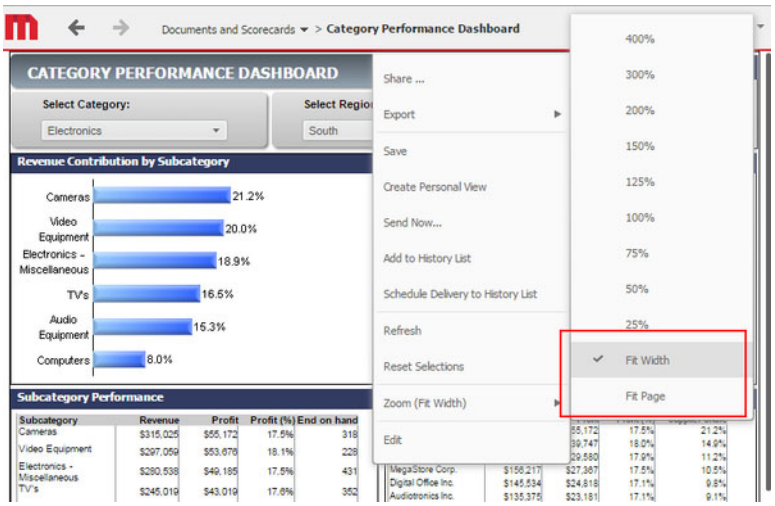
Optimize layouts for micro applications: Responsive design

Optimizing the layout for a micro-application allows you to better control the user's experience and interaction with the document. Selecting this option allows

you to size the layout for Android devices to fit the page or the width of the device's screen.



When a user views the document on an Android device, a fit-to-page layout is zoomed and displayed within the screen, so that they do not need to scroll vertically or horizontally to view any data. A fit-to-width layout is zoomed to the width of the screen, so that a user does not need to scroll horizontally to view any data.



Exercise 4.2: Create the BGH optimized mobile document

One of the requirements from the Application Architect is for the documents used in the mobile app to be contained within a single document. This can be accomplished through the use of a multi-layout document, as well as using

Mobile views for different devices. Using a multi-layout document saves the designers time since the document formatting and Mobile views only have to be defined once.

Another requirement from the Application Architect is the ability to leverage existing documents in the creation of the new mobile application. To meet all of the Application Architect's requirements as well as those of the stakeholders, you do the following in this exercise:

- Create a new document using a mobile document template.
- Optimize the new document for both tablets and phones using Mobile views.
- Create a multi-layout document that contains the two documents the end-users need.
- Re-brand the application document to BGH's guidelines.

Create a new document with Mobile views enabled

1 From the Shared Reports folder, open your **BGH** folder.

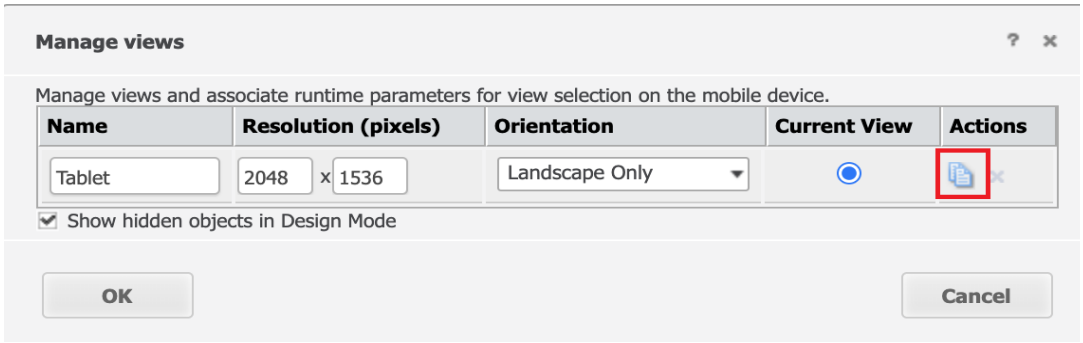
1 Click **Create,** point to **New Document**, and select **iPad Retina Landscape**.

2 Click the **Tools** menu, and select **Manage views**.

The document currently has one view configured that displays in Landscape orientation. Since BGH's app is going to be viewed on both tablets and phones, create a view for each device type.

3 Change the name of the first view to **Tablet**.

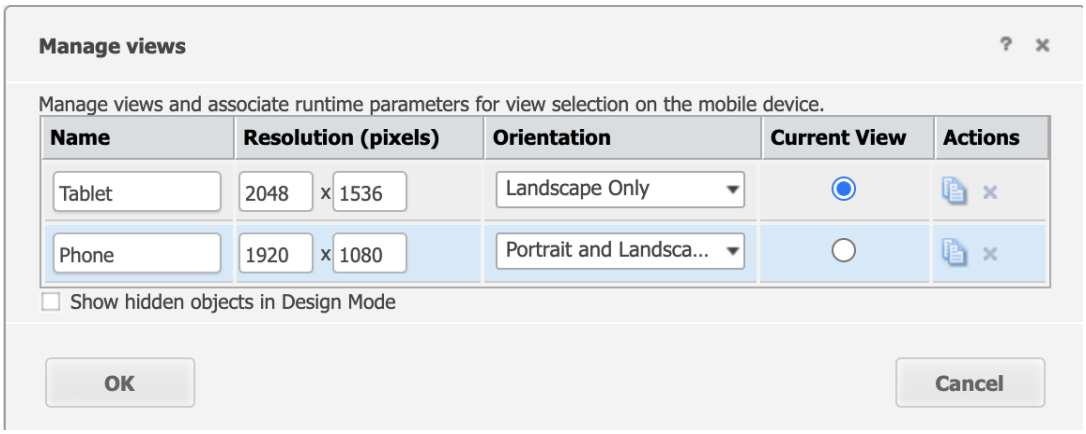
4 Click the **Duplicate** icon under Actions to create a second view.



5 Change the name of the second view to **Phone**, and the resolution to **1920 x 1080**.

6 For the Phone view, set the orientation to **Portrait and Landscape**.

7 Clear the check box next to **Show hidden objects in Design Mode**. Your Manage Views window should look like the one below.



8 Make sure Tablet is selected as the Current View, and click **OK**.

9 Save your document in your BGH folder as **BGH - Mobile app**.

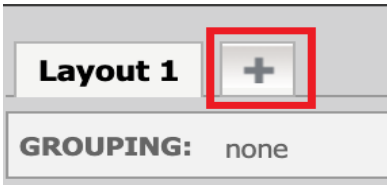
10 Click **Run newly saved document**.

Another benefit of placing all of the documents in one unified multi-layout document is that the multiple views only have to be configured once. Select each layout and open the Manage views window. Notice that the views are the same for each layout.

Add and format the Revenue and Margin Analysis layout

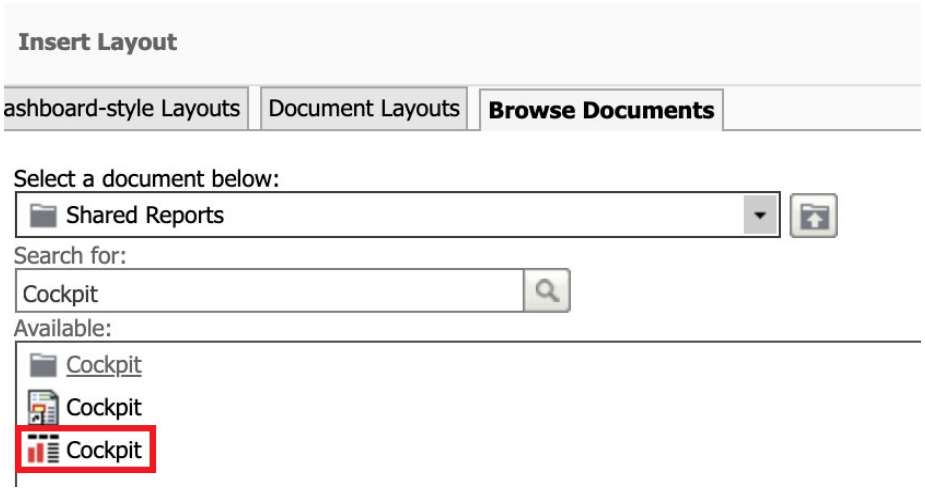
1 Select **Edit** from the menu to return to Design mode.

2 Select the **+** sign next to the Layout 1 tab to add a second layout.



3 Select the **Browse Documents** tab, type **Cockpit** in the Search for text box, and press **Enter**.

4 Select the **Cockpit** document and click **OK**.



The Cockpit document has now been added to your BGH mobile app document. Here we can make modifications to the document before deploying it to mobile devices.

5 Right-click the Cockpit layout tab, and select **Rename**.

6 Change the name to **Revenue and Margin Analysis**, and click **OK**.

Re-brand the document to BGH's guidelines

You want to add BGH's logo and app title at the top of each layout. This can be placed in the page header of the document.

7 From the **Tools** menu, select **Document Properties**.

8 Select **Sections** from the list of Properties on the left.

9 Select the **Page header (shared)** check box, and click **OK**.

10 Expand the page header by clicking the **+** next to it.

11 From the **Insert** menu, select **Image**. Click **Yes**, to make it visible in all views.

12 Use your cursor to draw the image in the left corner of the page header. The Properties and Formatting window opens.

13 Click **Browse**, select the **BGH logo.png** image from the Storyboard folder in your exercise files, and click **Open**.

14 A preview of your image displays.

15 Select **Color and Lines** from the list of Properties on the left.

16 Change the Borders to **None**, and click **OK**.

17 Insert a text box in the page header, and make it visible in all views.

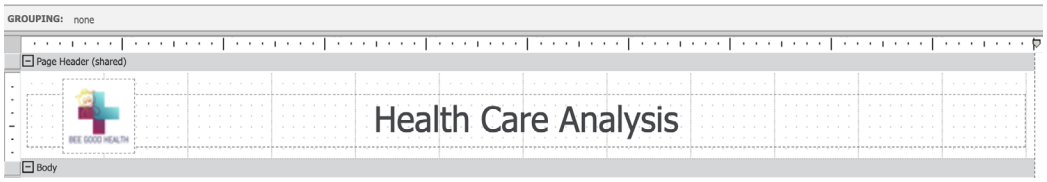
18 Type **Health Care Analysis** in the text box.

19 Format the text as follows:

- Text size: **24**
- Alignment: **Center**
- Fill Color: **No Fill**
- Border: **None**

20 **Save** you document.

Your page header should look similar to the image below:



Add the Store Performance layout

1 Click the **+** next to the Revenue and Margin Analysis tab to add a third layout.

2 From the Browse Documents tab, search for **Store Performance Management**.

3 From the list of documents, select **Store Performance Management Dashboard (For a specific Region)**, and click **OK**.

4 This document uses a prompted report as its dataset. With the default regions selected, click **Edit in Design Mode** in the bottom left.

Using a prompted report as a dataset not only meets one of the requirements of the Regional Sales Manager to be able to filter by region, it can also be used to narrow down the amount of data that the mobile app needs to load.

5 Right-click **Layout 1**, and select **Delete**. Click **OK**.

Optimize for Responsive design

Select the zoom for each view and layout.

6 Using the Document Home toolbar, change your Zoom to **Fit Width**.

7 Select the other layout, and set the Zoom to **Fit Width**.

8 Click the **Tools** menu, and select **Manage views**.

9 Change the **Current View** to **Phone**, and click **OK**.

10 Using the Document Home toolbar, change your Zoom to **Fit Page**.

11 Select the other layout, and set the Zoom to **Fit Page**.

12 **Save** your document.

13 Open your new multi-layout document in Presentation mode. Use the tabs at the top of the document to navigate between the layouts. Notice the page header is visible on both layouts.

Connecting document pages: Links Editor

To provide a simple end-to-end navigational workflow, designers should add links to app pages using the Links Editor. A link is a connection in a report, document, or dossier (the source) to another report, document, dossier, or web page (the target). A link can be created from a text field, an image, or a button.

Links can automatically pass parameters to selectors or answer any prompts that are in the target. For example, if a user is viewing a document containing regional sales, he can click a particular region to execute another document that displays sales for the stores in that region.

Link type: Navigate to this URL

The Navigate to this URL option allows the link to be configured to an existing web page or to a custom path using a mobile URL API link. The URL Application

Programming Interface (API) allows you to directly request MicroStrategy to perform actions by using specific arguments in the request, embedded in the address or link used to access MicroStrategy. For re-branded mobile apps that have a modified URL scheme, some features such as opening the home page or logging out of the app need to use a URL API.

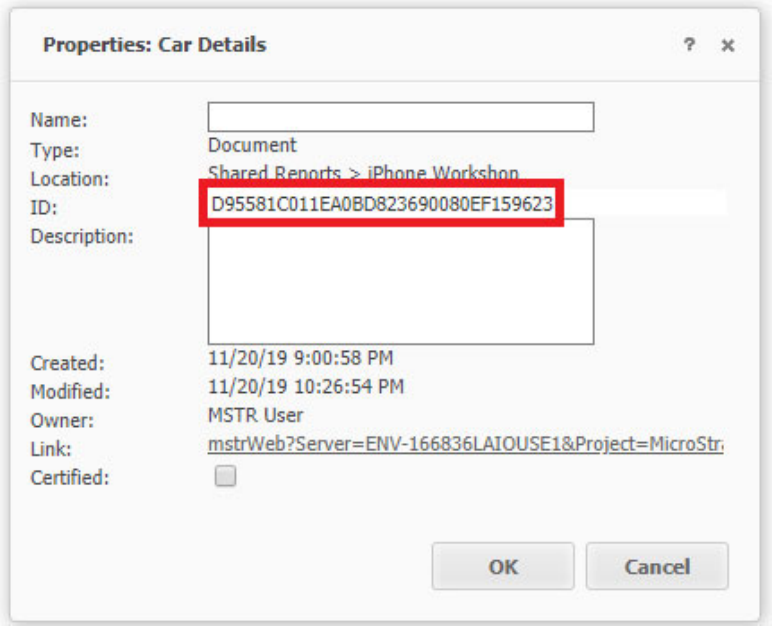
Using a URL API for mobile link allows you to use links to execute both MicroStrategy and non-MicroStrategy actions from inside of your MicroStrategy Mobile application. Below are examples of the functionality you can achieve by using a URL APL link on a mobile device:

- Allow users to execute a report or browse the contents of a folder.
- Enable drilling from a document to a report or other documents. You can also embed a document drilling link in a report grid within a document.
- Allow objects such as images, text boxes, and shapes to be used for application navigation.
- Link to a specific MicroStrategy page from any external website by passing appropriate parameters such as report name or folder name, as well as optionally including credentials for authentication by a specific MicroStrategy Intelligence Server.
- Pass filtering values to another document.
- Navigate within a document and between documents through a customized workflow that lets users display specific panels on a panel stack or maintain selections in selectors between documents.

Every request to MicroStrategy Web corresponds to at least one event or action. URLs typically include only one event, but in some instances, you may need a single URL to execute multiple events. To build a URL API link, start with **mstrweb?** and add the parameter and value for the event you want to execute. Parameter/value sets are separated by an ampersand **(&)**. The table below displays the two parameters needed to execute a specific document.

Parameter	Value
evt	2048001
documentID	ID of the report services document to execute

To locate the document ID for a document, right-click the document and select Properties.



In addition to specifying a document to execute, you can also specify the document layout to be displayed. The following parameter-value pair is used to target the document layout that should be displayed using the mobile URL API link. This parameter is appended to the URL APL link for executing a document shown above and is separated by an ampersand (&).

&layoutIndex=

The value used indicates which layout you want to display. The first layout in a document is 0, the second layout is 1, and so on. An example of a complete mobile URL API link to open a specific layout of a document is shown below:

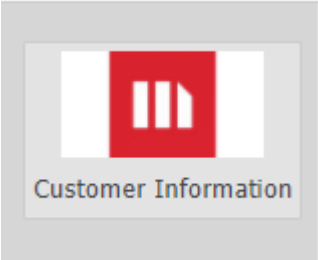
mstrweb?evt=2048001&documentID=
D95581C011EA0BD823690080EF159623&layoutIndex=0

The following high-level steps are for reference only, and not intended to be an in-class exercise.

Add a URL API link to a button to pass filtering values to the target

1 Insert a **Panel Stack** into your document. Adding a panel stack allows the button to be targeted by a selector.

2 Add a button to the panel stack.



3 Configure the button to navigate to a specific URL API link that passes filter info to the target.

For example, the link below passes filtering information to the target document.

```
mstrweb

?evt=4001

&src=mstrWeb.4001

&reportID=03981D054654EB41C053889A967429CE

&reportViewMode=1

&elementsPromptAnswers=

8D679D4B11D3E4981000E787EC6DE8A4;8D679D4B11D3E4981000

E787EC6DE8A4:1%5eNortheast
```

- evt - event/action to be performed, e.g. 4001 for running a report
- src - web page component to perform the event
- reportID - report object ID to be executed
- reportViewMode - the mode in which the report is displayed, e.g. 1 for Grid mode

The parameters to pass an answer to an element list prompt:

- elementsPromptAnswers=AttrID;AttrID:ElemID%5eElemName
- AttrID - object ID of the prompted attribute
- ElemID - element ID of the attribute element used as prompt answer

  - ElemName - description of the attribute element used as prompt answer

For more information on using URL APIs, refer to the URL API Reference document provided within the exercise files for class, and take the *Advanced SDK for Customizing Branding* course.

Exercise 4.3: Configure mobile URL API links

In this exercise you configure three mobile URL API links that perform the following tasks:

- Link to the second layout of the BGH - Mobile app document
- View MicroStrategy subscriptions for the user
- Log out of the app

Configure the Store Performance button

1. From your BGH folder, right-click the **BGH - Mobile app** document, and select **Properties**.
2. Copy the **Document ID** and paste it into Notepad to be used later.
3. Click **OK** to close the Properties window.
4. Open the BGH - Landing page in Design mode.
1. Right-click the **Store Performance** button, and select **Properties and Formatting**.
2. In the list of categories on the left, click **Button**.
3. Click **Configure actions on this button**.
4. Select **Navigate to this URL.**
5. Create your mobile URL API link using the example below and enter it in the text box.

   Replace the document ID in the following mobile URL API link with the one from your document that you copied in a previous step.

mstrweb?evt=2048001&documentID=*425D58A211EA49CBA77D0080EF35A0B1*&layoutIndex=1

Mobile URL API links only work on mobile devices. To configure the link to work in Web and Mobile enter it as mstrWeb.

6 Click **OK**, and **Close**.

7 **Save** your document.

Configure the subscription and logout mobile URL API links

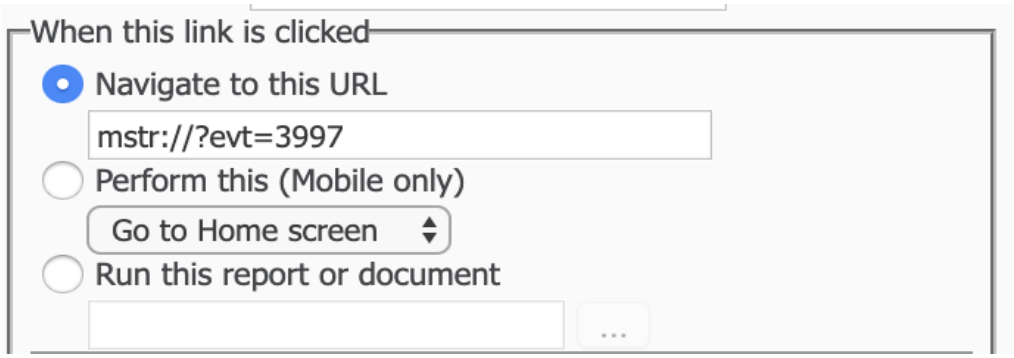
1 From the **Insert** menu, select **Text.**

2 Draw a text box on your Landing Page and type **Subscriptions** inside.

3 Add another text box below Subscriptions, and type **Logout** in it.

4 Right-click the **Subscriptions** text box and select **Edit Links.**

5 Select **Navigate to this URL** and type **mstr://?evt=3997.**



6 Click **OK**.

7 Right-click the **Logout** text box and select **Edit Links.**

8 Select **Navigate to this URL** and type **mstr://?evt=4000.**

9 Click **OK**.

10 **Save** your document.

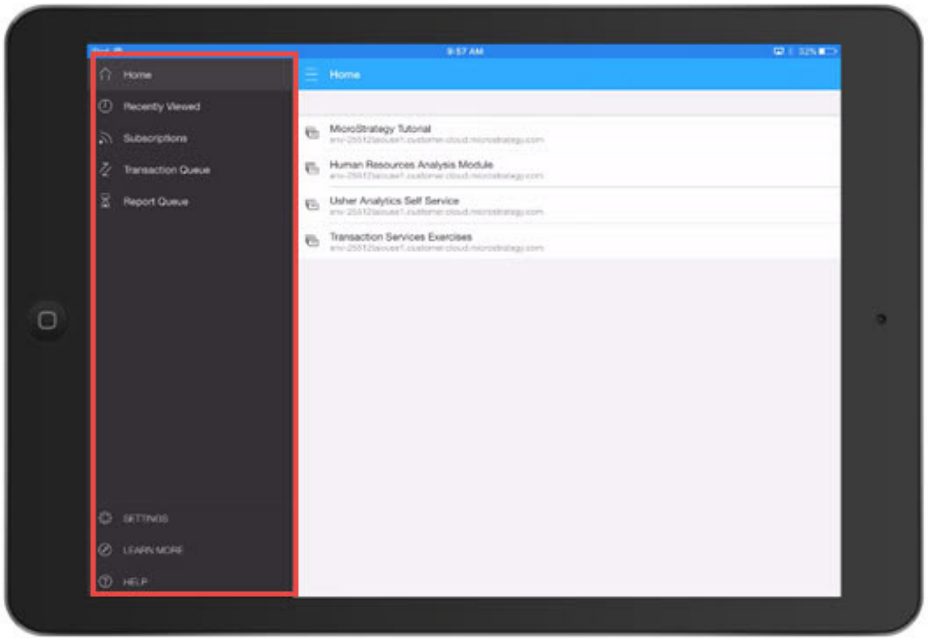
Reminder: Mobile URL API links only open on mobile devices.

Link type: Perform this (Mobile only)

These out-of-the-box MicroStrategy links allow you to link to a predefined MicroStrategy screen on a mobile device. Available options include:

- Go to Home screen (MicroStrategy home screen)
- Go to Settings screen
- Go to Status screen
- Go to Report list
- Go to Shared library
- Go to Help

By default, when you open an out-of-the-box MicroStrategy mobile app, the first screen is a standard Home screen that allows you to select your project or folder to view. On the left side of the screen, you can view the Navigational panel used to access the options listed above (you may need to swipe left to view the Navigation panel on a phone).



However, if your app is configured to deploy a document instead of the standard home screen, you cannot access this navigation pane. This is a perfect example of when to use **Perform this (mobile only)** link type, which allows you to access any of the pages.

Link type: Run a specific report or document

This option allows you to execute another MicroStrategy document, dossier, or report, which is the most common link type. You can also pass selector values from the source document to the target document or report, as long as both the source and the target contain the same selector or dataset object. A selector allows each user to interact with a document to display only the subset of data they are interested in or only specific attribute elements or metrics.

This means that either both documents must contain a selector with the same name (such as Region Selector), or both documents must contain a selector that uses the same source object, such as Region. When you create a link that passes selector values, you can choose to match the selector values either by the selector name or the source object.

Exercise 4.4: Pass attributes from the Landing Page to the app document

To complete the navigation between your landing page and your app document, in this exercise you:

- Create a data light dataset for the landing page
- Insert a new panel stack on your landing page for the navigation buttons
- Configure the selectors to target the new panel stack
- Configure the logo on the BGH - mobile app document as a link to the BGH - landing page
- Test your new app on your mobile device

Create the landing page dataset

To pass selector values from the source document to the target document or report, both the source and the target must contain the same selector or dataset object. Create a new dataset with only the needed objects to keep app's landing page simple and light.

1. From the Shared Reports folder, click **Create**, point to **New Report**, and select **Blank Report**.
2. In the All Objects panel on the left, select **Attributes**, then **Geography**.

3 Double-click **Region** to add it to the report.

4 Go up one level, and select the **Products** folder.

5 Double-click **Category** to add it to the report.

6 Save the report in your **BGH** folder as **Landing page dataset**.

7 Close the report.

Configure the buttons panel stack

1 Open the **BGH - Landing page** document in Design mode.

2 Insert a new panel stack on top of your Main Panel stack by selecting **Panel Stack** from the **Insert** menu, and drawing it on your document.

Adding a panel stack allows the buttons to be targeted by a selector.

3 Right-click the new panel stack, and select **Properties and Formatting**.

4 Select General from the list of properties on the left, and change the name to **Buttons Panel Stack**.

Best Practice

As a best practice, you should name your panel stacks when a document contains multiple panel stacks, so you can quickly target the correct one.

5 In the Panel Stack area, clear the **Show Title Bar** check box.

6 Select **Color and Lines** from the list of Properties on the left.

7 Set the **Fill color** to **No Fill** and the **Borders** to **None**.

8 Continue formatting the panel to match the style of your Landing Page, then click **OK**.

9 Move the Revenue and Margin Analysis and Store Performance buttons on the Buttons Panel Stack. Reposition the panel stack if needed.

10 **Save** your document.

Configure the attribute selectors

Add the Landing page dataset

1 In the Dataset Objects panel, click **Add a Dataset**.

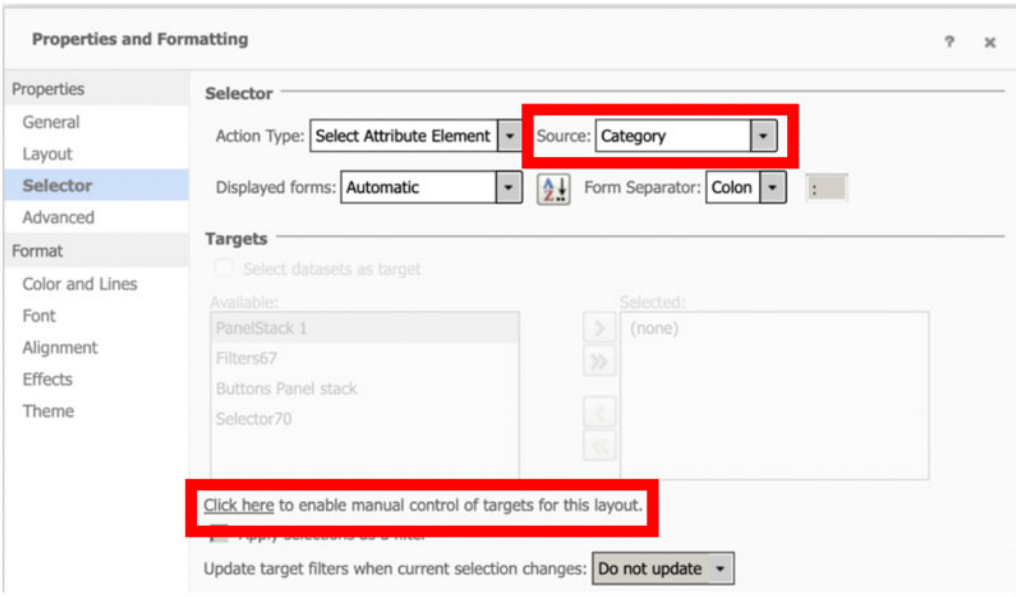
2 From the Shared Reports folder, select your **BGH folder**.

3 Select **Landing page document**, and click **OK**.

4 Right-click the first drop-down selector, and select **Properties and Formatting.**

5 In the list of categories on the left, click **Selector** and select **Category** from the **Source** drop-down list.

6 To set the target for the selector, select **Click here to enable manual control of targets for this layout**.



7 Click **OK** on the warning stating you need to manually maintain targets for all selectors in this layout.

8 Select **Buttons Panel stack** from the Available list, and use the right arrow to move it to the Selected list. Click **OK**.

9 Use the steps above to configure the second selector using **Region** as the source and the **Buttons Panel stack** as the target.

10 Click **Save**.

Pass the selector values to the Revenue and Margin Analysis linked document

Use the Links Editor to configure the Revenue and Margin Analysis button to pass the values of the Category and Region selectors to the target document.

1 Right-click the **Revenue and Margin Analysis** button, and select **Properties and Formatting**.

2 In the list of categories on the left, click **Button**.

3 On the bottom of the Button window, click **Configure actions on this button**.

4 Select **Run this report or document,** then click the **Select Target** icon to browse to the target document.

5 In the Select Target window, navigate from the Shared Reports folder to your BGH folder.

6 Select the **BGH - Mobile app** document, and click **OK**.

7 From the **Pass all selector values** drop-down list, select **Match selectors by source attribute**.

8 Click **OK**, and **Close**.

9 **Save** your document.

Configure the BGH logo to link to the BGH - Landing Page

To complete the navigation between the documents, configure the BGH logo to allow users to return to the app's landing page.

1. Open the **BGH - Mobile app** document in Design mode.
2. Right-click the logo, and select **Edit Links**.
3. On the Links Editor, select **Run this report or document**, then click the **Select Target** button to browse.
4. From the Shared Reports folder, navigate to your BGH folder, and select **BGH - Landing page**.
5. Click **OK** and **OK**.
6. **Save** your document.

Add your document to Library and test your app in Library Mobile

Both dossiers and documents can be added and viewed in MicroStrategy Library. Add your app landing page to Library, so you can view and interact with it in Library Mobile.

1. Right-click the **BGH - Landing page** document, and select **Share**.
2. From the Share window, select **Library link** and click **Launch.** You are now viewing your document in Library.
3. Click the blue **Add to Library** button at the top of the page to add the BGH-Landing page document to your Library.
4. On your mobile device, open the **MicroStrategy Library** app.
5. Locate and open the **BGH - Landing page** document.

Test your app in the MicroStrategy Mobile app

1. Open the **MicroStrategy Mobile** app on your mobile device. Your app opens to the Shared Reports folder.

2 Open the BGH folder, and select **BGH - Landing page**.

3 Select **Central** from the **Region** selector and **Books** from the **Category** selector.

4 Click **Revenue and Margin Analysis**. Notice when the document runs, the selectors at the top of the document match the selector values from the landing page.

5 Click the **BGH logo** to return to the landing page.

6 Click **Store Performance**. Notice the document runs and displays layout 2.

The Store Performance Management Dashboard uses a prompted report where the user can select region attribute elements.

Enhancing data analysis: Customizing visualizations

To support departmental decision making, designers should use appropriate visualizations to display data in an actionable way. MicroStrategy comes with many out-of the-box visualizations; however, there may be times when your team needs to create a unique visualization to give better insight into enterprise data.

Your team can leverage the Visualization SDK to extend existing widgets and visualizations. They can also build completely new widgets, visualizations, interactivity, and workflows to suit enterprise needs. As the Mobile Architect, you are responsible for ensuring best practices are met and appropriate visualizations are used when a designer adds a custom visualization to an app page.

A visualization plug-in incorporates XML configuration files and a JavaScript file that renders the visualization. It may also include some other files, such as images that are used by the visualization. You can refer to the MicroStrategy Community visualization gallery to find various custom visualizations that other MicroStrategy members have created.

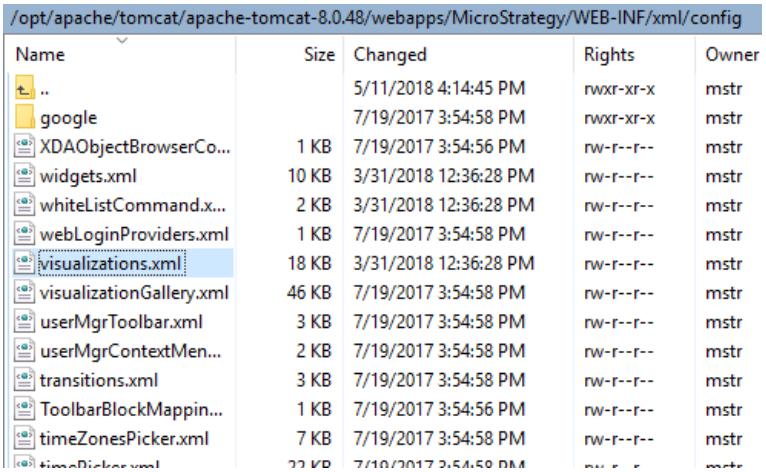
The steps below show you how to add a visualization created by your team.

These steps are for reference only, they are not intended to be performed in class.

Add a custom visualization

1 Copy the visualization plug-in to the **plugins** folder in the MicroStrategy Web installation directory.

2 In the visualization plug-in folder in the MicroStrategy Web installation directory, navigate to **..\WEB-INF\xml\config\visualizations.xml** for the visualization.



3 Open **visualizations.xml** and duplicate the entire **<visualization-list name= "ajax">** tag, including all contents.

4 In the duplicated XML, make the following changes:

- In the <visualization-list> node:

  a Change the value of the name attribute from "ajax" to "**device**".

  where **device** is one mobile device name from the table below.

Device	device value	view-mode value
iPhone	iphone	70
iPad	ipad	71
Android Phone	android	72
Android Tablet	androidTablet	73
Web	ajax	51

b Change the value of the show-in-web attribute from **true** to **false**.

c Change the value of the view-mode attribute from **51** to **view-mode value**.

where **view-mode value** is the correct value for the device based on the table above.

- In the <visualization> node:

a Delete the **is-mojo="true"** attribute.

5 Save the modified visualization plug-in.

To view a custom HTML5 visualization in a document on a mobile device

1 Copy the modified visualization plug-in to the plugins folder in the MicroStrategy Mobile installation directory.

2 Restart the Web server.

3 When you add the visualization to the document in MicroStrategy Web, make sure to select the **device** option in the widget editor based on the device you are designing for, such as a tablet, and save the document.

Exercise 4.5: Deploy a custom visualization to MicroStrategy Web

BGH uses the Bar Chart Race Visualization for Time Lapse, to display revenue values for the previous quarters to analyze their client hospitals' Centers of Excellence. The Bar Chart Race visualization does not come out-of-box for MicroStrategy, so you need to add a custom visualization to MicroStrategy Web.

Since the plug-in for the Bar Chart Race Visualization for Time Lapse is available on MicroStrategy Community, you can upload it using the Web Administrator page.

Download the Bar Chart Race plug-in from MicroStrategy Community

In addition to discussions, idea exchanges, and FAQs, MicroStrategy Community includes a data visualization gallery where you can download and customize visualizations.

1. To access the visualization gallery, visit:

   https://community.microstrategy.com/s/gallery

   in your web browser.

2. Scroll through the gallery until you locate the Bar Chart Race Visualization for Time Lapse.

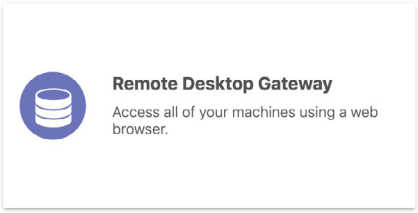
3. Select the **Bar Chart Race Visualization for Time Lapse**. The page opens, explaining the uses and requirements of the visualization.

4. Under **Files**, click **D3BarChartRace.zip** to download the file to your computer.

5. Close the Community tab.

Add the visualization plugin to MicroStrategy Web

1. From the Welcome to MicroStrategy landing page under More Resources, hover over **Remote Desktop Gateway**, and click **Launch**.



2. Enter your credentials from your Welcome to MicroStrategy Cloud email, and click **Login**.

3 Under All Connections, select **Developer Instance RDP**.

4 On your local computer, open the folder where you have the **D3BarChartRace.zip** file.

5 Drag the zip file onto the remote desktop of your MicroStrategy Cloud environment. The file downloads to the Guacamole Filesystem.

6 Open File Explorer on your remote desktop, expand **This PC** on the left, and select the **Guacamole Filesystem on Guacamole RDP**.

7 Right-click the **D3BarChartRace.zip** file, and select **Extract All**.

8 Click **Browse**, select **Desktop**, click **Select Folder**, then **Extract**.

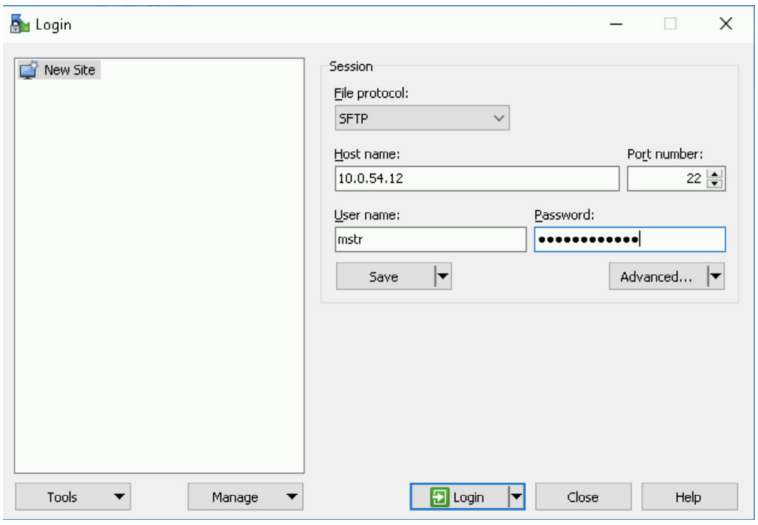
Your custom visualization is placed in a folder on the desktop and is ready to be placed in the plugins folder.

Connect to the Linux server

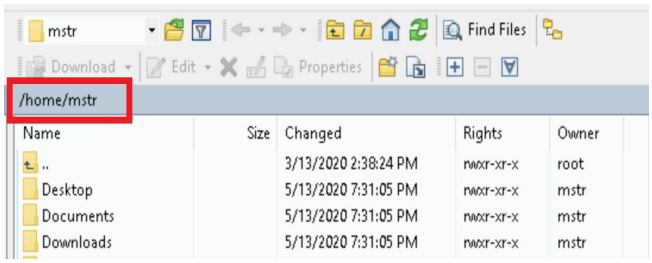
9 Double-click the **WinSCP** shortcut on the desktop. This application allows you to browse and upload files to the Linux server where MicroStrategy is installed.

10 In the **Host Name** text box, type the **Intelligence Server IP Address**, located in the Essential Connections portion of the Welcome to MicroStrategy landing page.

11 Enter your login credentials from your Welcome to MicroStrategy Cloud email in the **User Name** and **Passwords** text boxes, and click **Login**.

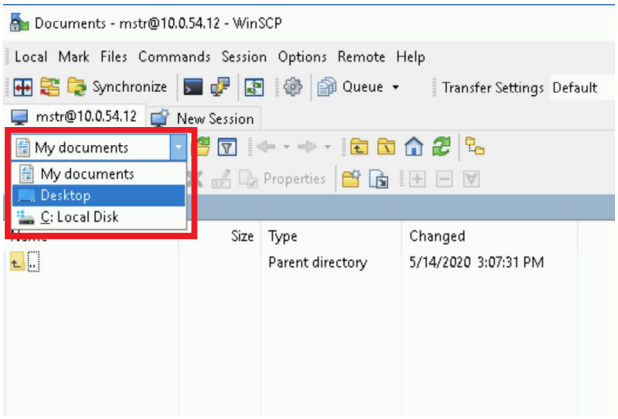


12 On the right side of WinSCP, double-click **/home/mstr**, enter the path below, and click **OK**.



/opt/apache/tomcat/apache-tomcat-9.0.30/webapps/MicroStrategy/plugins

13 On the left side of the WinSCP window, select **Desktop** from the drop-down list.



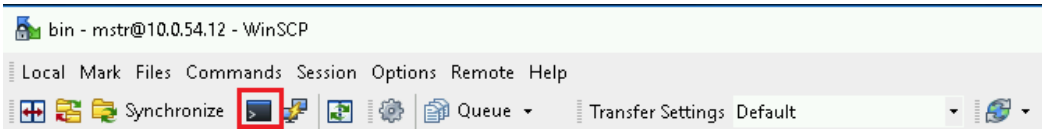
14 Once the desktop loads, select the **D3BarChartRace** folder, click **Upload,** then click **OK**.

This uploads the Bar Chart Race Visualization for Time Lapse to the MicroStrategy Web plugins folder. For the change to take effect, the tomcat web server must be restarted.

Restart tomcat on the Linux server

15 On the right side of WinSCP, navigate to **/opt/apache/tomcat/apache-tomcat-9.0.30/bin**.

16 Click the **Open terminal** icon on the toolbar shown below, to open a new command prompt window.



17 In the **Enter command** text box, type: **service mstr tomcatrestart**, and click **Execute.**

Your connection to the remote desktop and MicroStrategy Web is lost temporarily.

18 Wait a couple minutes for tomcat to restart, then click **Reconnect**.

Making custom visualizations successful

Best Practice

The Mobile Architect should work with the Services Architect to ensure all best practices are met for any customs plug-ins deployed on the Mobile Server. Best practices include:

- Avoid using symbols in the names of any files or folders stored within the resource plug-in to ensure the file is readable.
- For production environments, use minified versions of JavaScript libraries to further reduce load time. Minified resources have all unnecessary characters removed from source code without changing functionality, reducing file size. The impact is more noticeable in larger resources.
- Use the file:// prefix for local relative URLs. This allows the same relative URL to work in MicroStrategy Web and Mobile.
- Don't put any JavaScript comments in the externalLibraries object. It breaks parsing in MicroStrategy Mobile.

Interactive data discovery: Dossiers

Dossiers are another important facet in Enterprise Mobility. In addition to creating mobile applications with documents, you can also interact with dossiers on-the-go with MicroStrategy Mobile (on iPad) and Library Mobile. Dossiers are interactive and intuitive business intelligence dashboards with a responsive and modern interface. As the Mobile Architect, you are responsible for ensuring that dossiers are appropriately designed for mobile consumption. You can also deploy documents to the MicroStrategy Library Mobile app.

Exercise 4.6: Create a dossier for BGH

In this exercise, you add the Bar Chart Race Visualization for Time Lapse to a dossier, and make the dossier available in the BGH mobile app and Library. Since one of the requirements of the BGH mobile app is to have a visualization that compares current revenue vs last year's revenue, the D3 Bar Chart, along with KPI visualizations, are used to fulfill that requirement.

In this exercise, you:

- Create a new dossier and add existing objects as its dataset
- Use the Bar Chart Race Visualization for Time Lapse to view the regional revenue by quarter
- Add two KPI visualizations to allow users to quickly view the current revenue and last year's revenue
- Share your dossier to Library

Create a new dossier and add dataset objects

1 From the Shared Reports folder in MicroStrategy Web, open the BGH folder.

2 Click **Create**, and select **New Dossier**.

3 From the Datasets panel, click **Existing Objects**.

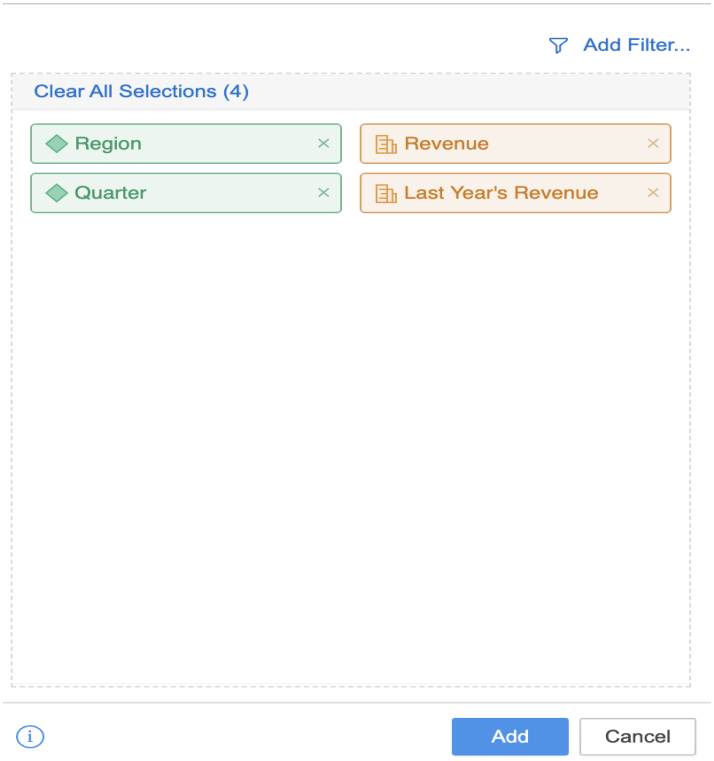
4 Expand Geography, and double-click **Region**.

5 Expand Time, and double-click **Quarter**.

6 From the drop-down list at the top, select **Metrics**.

7 Expand the Sales Metrics folder, and double-click **Revenue**.

8 Expand Transformation Sales Metrics, and select **Last Year's Revenue**.



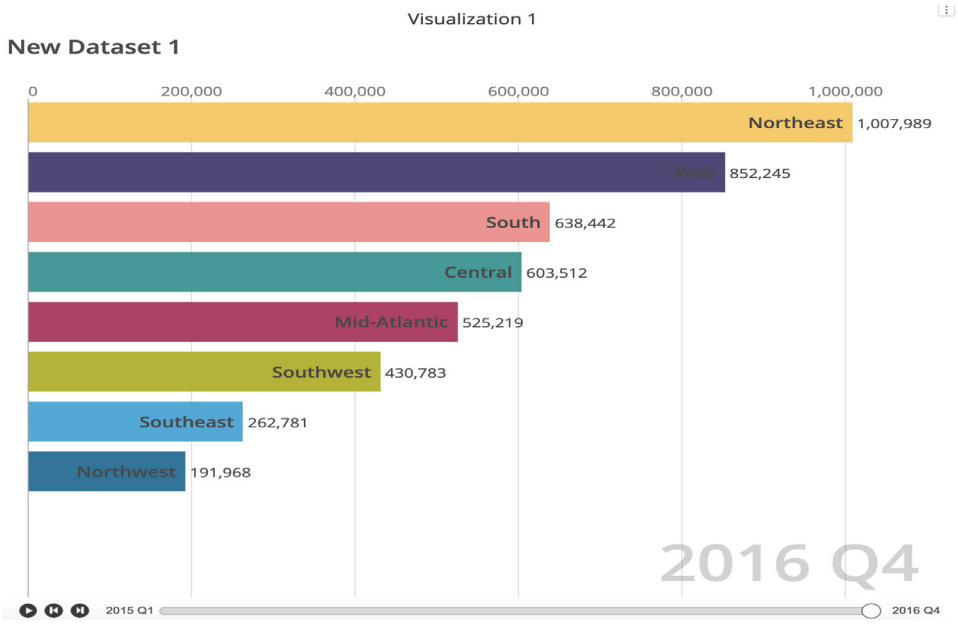
9 Click **Add**. The objects are now added as the dossier's dataset.

Use the Bar Chart Race Visualization for Time Lapse

1 From the Visualization Gallery under Custom, select the **D3BarChartRace** visualization.

2 Click and drag **Region** to the **Name Attribute** drop zone, **Quarter** to the **Trend Attribute** drop zone, and **Revenue** to the **Value Metric** drop zone.

Your D3 Bar Chart Race visualization should look like the image below.



Format the D3 Bar Chart Race visualization

3 Select the **Format** panel, and choose **Configuration** from the drop-down list.

4 Change the color of the **Label Font** to **white** to make the text visible against the color of the bars.

5 Double-click the visualization's title and type **Regional Revenue by Quarter**.

6 From the Format menu, change the dossier **Palette** to **Hummingbird**.

7 Save the dossier in the BGH folder as **BGH - Regional Revenue**.

8 Click **Run Newly Saved Dossier**.

Add the revenue KPI visualizations

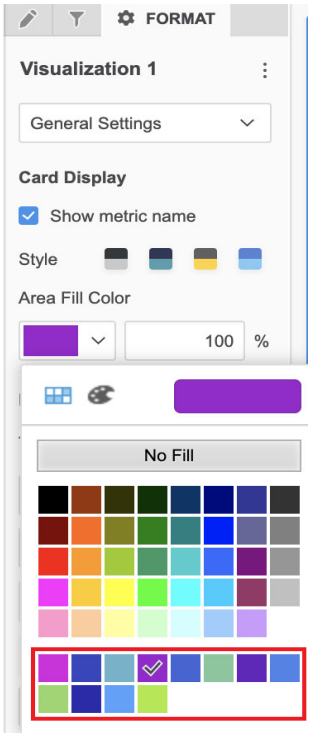
1 Add a new visualization to your dossier, and drag it above the D3 Bar Chart Race visualization.

2 Select the KPI visualization from the gallery.

3 Drag **Last Year's Revenue** to the **Metric** drop zone on the Editor panel.

4 Select the **Format** panel and **General Settings** from the drop-down list at the top.

5 Under Card Display, change the **Area Fill Color** to a color from the Hummingbird palette as shown below:



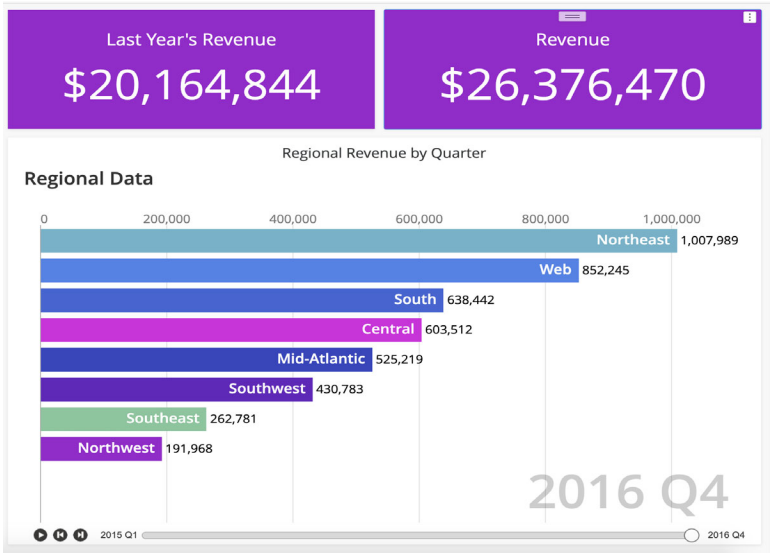
6 Select **Title and Container** from the drop-down list, and clear the **Show Title Bar** check box.

7 Click the menu icon in the upper right of the KPI visualization, and select **Duplicate**.

8 Click and drag the new visualization so the two KPIs are side-by-side.

9 Select the new visualization, and replace the Last Year's Revenue metric with the Revenue metric.

Your dossier should look similar to the one below.



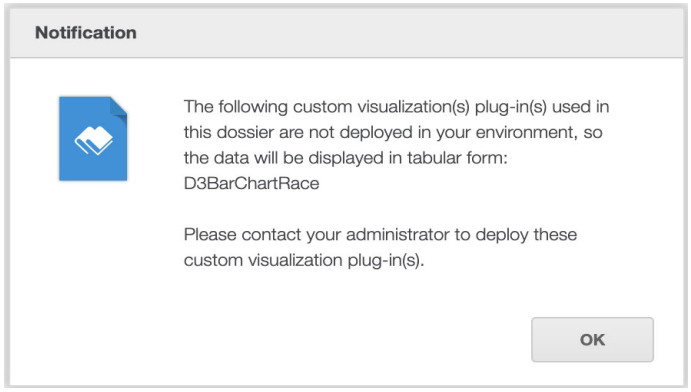
10 Right-click **Chapter 1** in the Contents panel, select **Rename**, and type **Regional Revenue**.

11 Click **Save**.

Share the dossier to Library

1 From the **Share** menu, select **Get a link to MicroStrategy Library**.

2 Select **Library Link**, then click **Launch**. You receive the following notification because the custom visualization has not been added to Library.



3 Click **OK**, and close the Library tab.

Add the D3BarChartRace visualization to the Library plugins folder

The custom visualization has only been installed for MicroStrategy Web. To view the custom visualization in Library, the visualization must be installed in the Library plugins folder as well.

1 From the Welcome to MicroStrategy landing page under More Resources, hover over **Remote Desktop Gateway**, and click **Launch**.

2 Enter your credentials from your Welcome to MicroStrategy Cloud email, and click **Login**.

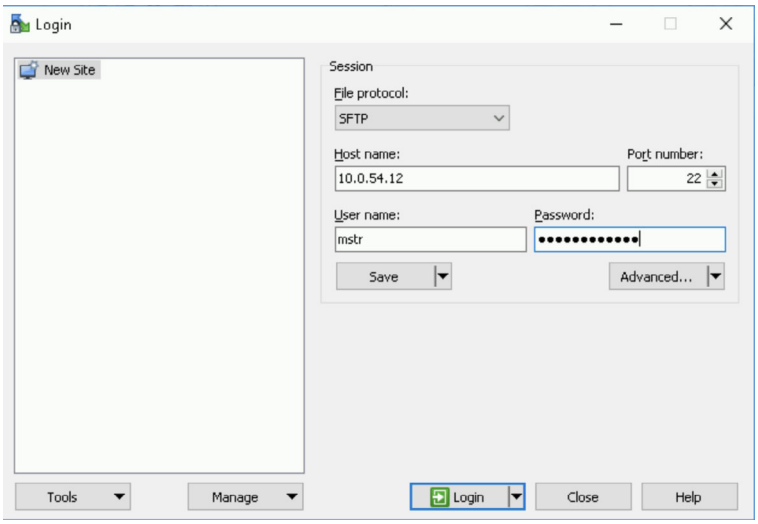
3 Under All Connections, select **Developer Instance RDP**.

Connect to the Linux server

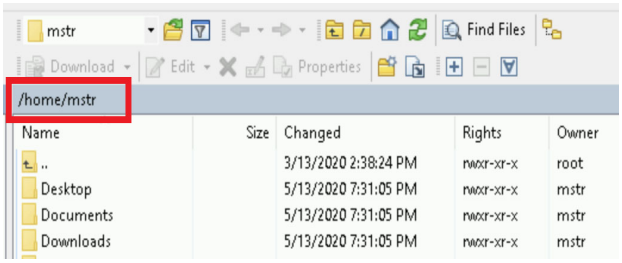
4 Double-click the **WinSCP** shortcut on the desktop.

5 In the **Host Name** text box, type the **Intelligence Server IP Address**, located in the Essential Connections portion of the Welcome to MicroStrategy landing page.

6 Enter your login credentials from your Welcome to MicroStrategy Cloud email in the **User Name** and **Passwords** text boxes, and click **Login**.

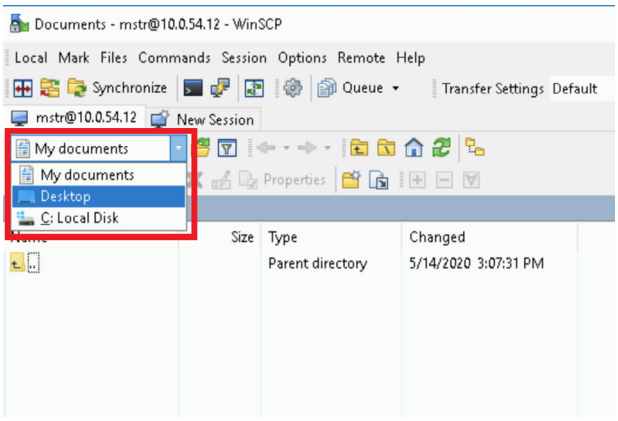


7 On the right side of WinSCP, double-click **/home/mstr**, enter the path below, and click **OK**.



/opt/apache/tomcat/apache-tomcat-9.0.30/webapps/MicroStrategyLibrary/plugins

8 On the left side of the WinSCP window, select **Desktop** from the drop-down list.



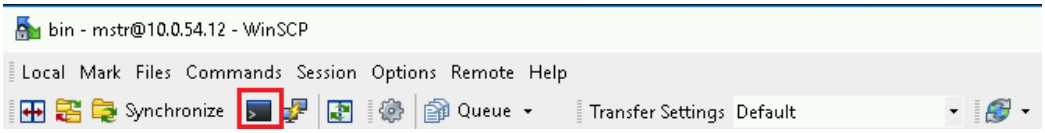
9 Once the desktop loads, select the **D3BarChartRace** folder, click **Upload,** then click **OK**.

This uploads the Bar Chart Race Visualization for Time Lapse to the Library plugins folder. For the change to take effect, the tomcat web server must be restarted.

Restart tomcat on the Linux server

10 On the right side of WinSCP, navigate to **/opt/apache/tomcat/apache-tomcat-9.0.30/bin**.

11 Click the **Open terminal** icon on the toolbar shown below, to open a new command prompt window.



12 In the **Enter command** text box, type: **service mstr tomcatrestart**, and click **Execute.**

Your connection to the remote desktop and MicroStrategy Web is lost temporarily.

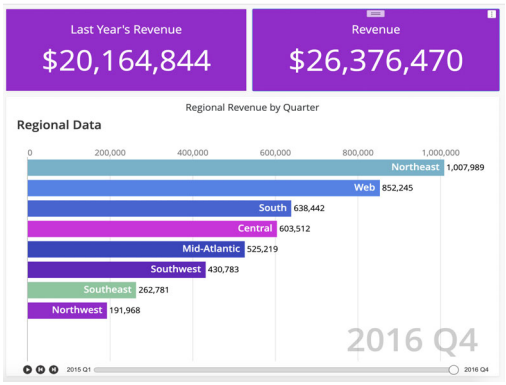
13 Wait a couple minutes for tomcat to restart, then click **Reconnect**.

Share the dossier to Library and view it on your mobile device

1 Return to MicroStrategy Web, and open your BGH - Regional Revenue dossier.

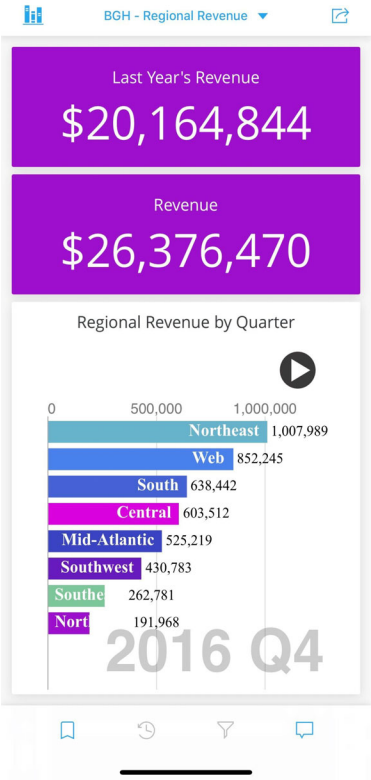
2 From the **Share** menu, select **Get a link to MicroStrategy Library**.

3 Select **Library Link**, then click **Launch**. You can now view your dossier, with your custom visualization, in Library.



4 Click **Add to Library** in the upper right to add your dossier to your Library for future viewings.

5 Using Library Mobile, open the BGH - Regional Revenue dossier. Notice the KPIs are stacked, and not displayed side-by-side as designed. This is addressed in the following exercise.



Optimizing dossiers for Library Mobile

To ensure your enterprises' dossier pages are optimized for mobile use, when designing dossiers in Web, you should require that authors leverage Responsive Preview and the Responsive View Editor.

With a single click, Responsive Preview mode allows authors to see how the dossier displays on a mobile device, to determine if any changes need to be made. For example, using Responsive Preview mode can show you that you need to group a pie chart and bar chart showing revenue together so mobile users can see contextual information on one screen.

Exercise 4.7: Optimize the BGH - Regional Revenue dossier

In this exercise, you optimize the BGH - Regional Revenue dossier for mobile viewing by using the Responsive View Editor to group the two KPI visualizations.

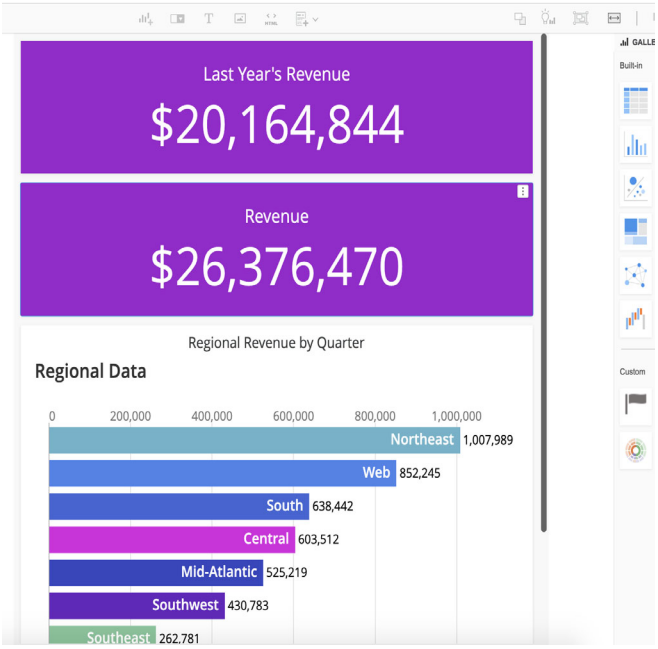
Then, you open the dossier again in Library Mobile to view the changes.

Optimize the dossier for mobile viewing

1 Open the BGH - Regional Revenue dossier.

2 To view the dossier as it displays on a mobile device, click the **Responsive Preview** icon.

Notice that the KPI visualizations are stacked. To improve mobile user experience, and keep the entire custom Bar Chart visible without scrolling, group the KPIs to display side-by-side.



3 Click the **Full View** icon to return to Edit mode.

4 Select **Responsive View Editor**.

5 Select the two KPI visualizations. They are both highlighted in blue.

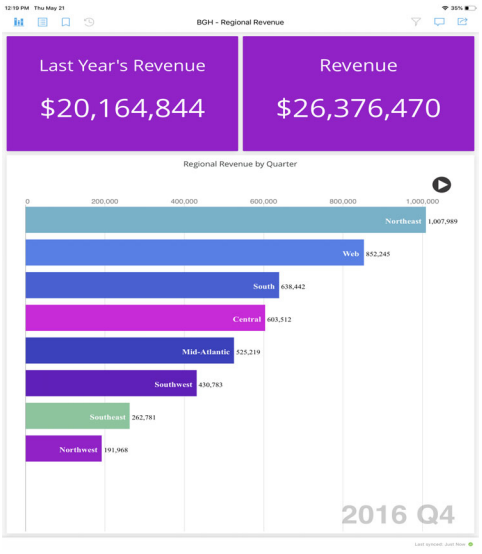
6 Click **Group** to group the visualizations, then click **Save** to save your changes. You are returned to Edit mode.

7 Click the **Responsive Preview** icon again to view the page in mobile preview. Now, the two visualizations are grouped together at the top of the dossier.

8 **Save** and **Close** the dossier.

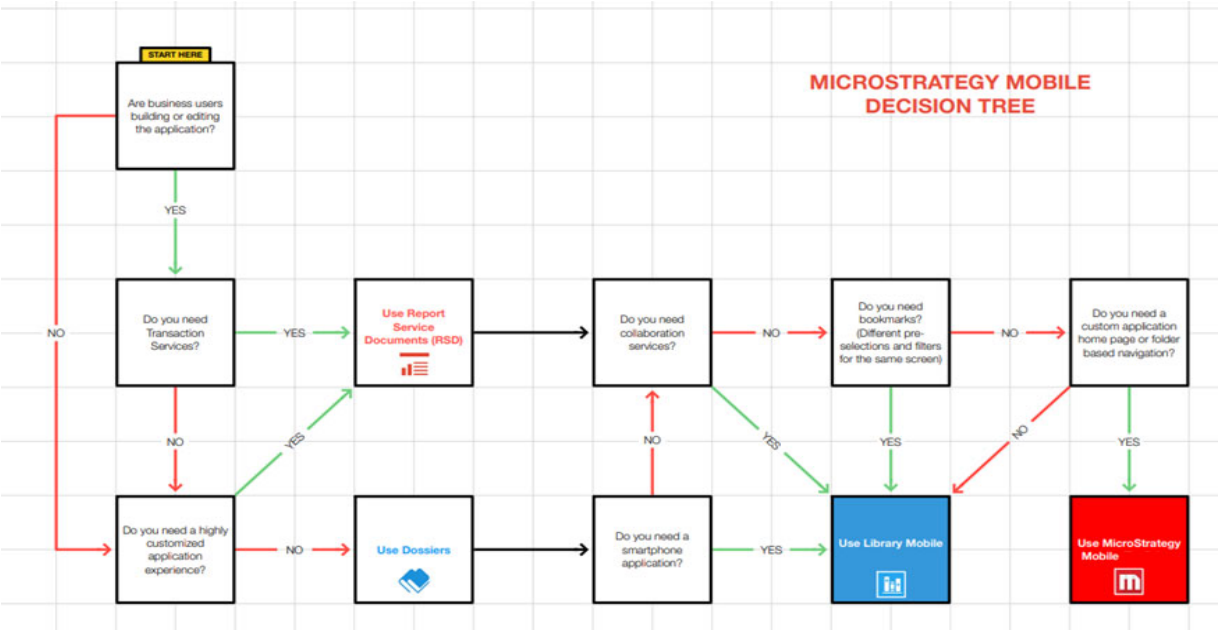
View the dossier in Library Mobile

1 Open the BGH - Regional Revenue dossier in Library Mobile. Notice the KPIs are displayed side-by-side as designed, and the full bar chart can be seen without scrolling.



Determining end user goals: Dossier or document

During the initial design phases of an enterprise application, your designers should determine if a dossier or a document fits end user goals. As the Mobile Architect, you should establish the follow a decision tree to help your designers determine which medium is best for the app.

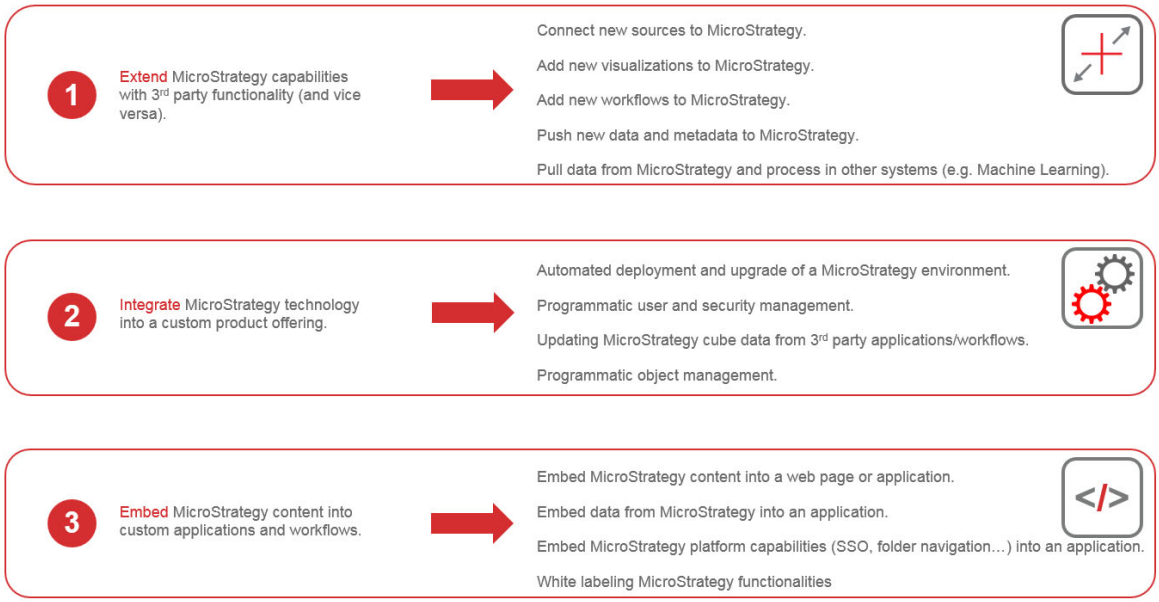


Create customized apps: Mobile SDK

Through the use of Mobile SDK and Library SDK, enterprises can customize apps built using the MicroStrategy Mobile platform. Customizations are most commonly used to apply corporate branding, such as colors and icons, to the software and to design specific user workflows to support corporate needs. The Mobile Architect should understand what can be achieved through SDK to elevate enterprise apps to the level of an Intelligent Enterprise and when to engage the Services Architect.

Using Mobile SDK, enterprise applications can be re-branded, pre-configured, and customized to perform actions such as disabling software encryption,

enabling Single Sign-on, and use mobile device features such as integration with Google Maps. The three key scenarios for using SDK are:



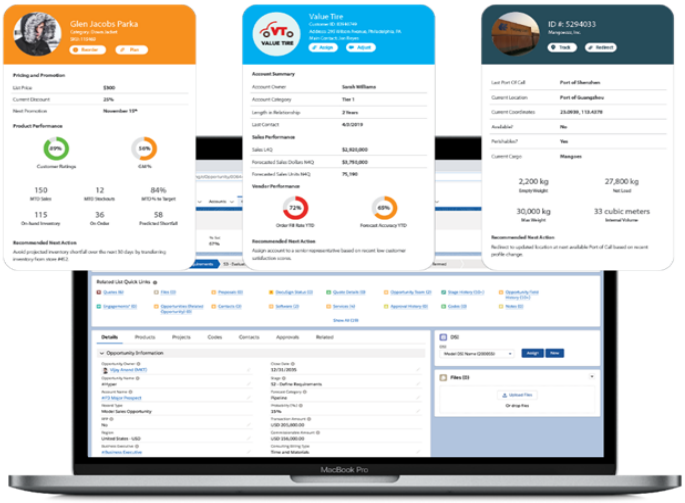
To learn more about MicroStrategy Mobile SDK and Library SDK, take the following courses:

- *SDK for Customizing iOS Applications*
- *SDK for Customizing Android Applications*

HyperIntelligence for Mobile: HyperMobile

HyperIntelligence is the biggest breakthrough in the world of analytics since Mobile, providing Zero-Click Intelligence to every user throughout the organization. Developers can use Workstation to quickly build and deploy cards that are designed for use with the MicroStrategy HyperIntelligence browser

extension on both Google Chrome and Microsoft Edge, the MicroStrategy HyperMobile app, and MicroStrategy HyperOffice.



HyperIntelligence for Mobile is an application that allows you to view and interact with cards on both iOS and Android devices. With features such as iOS Spotlight Search, calendar-based notifications, and a new Calendar tab, you can search for a specific card in your environment's database, enable your device to send push notifications with relevant cards, or view card related to your up coming calendar events.

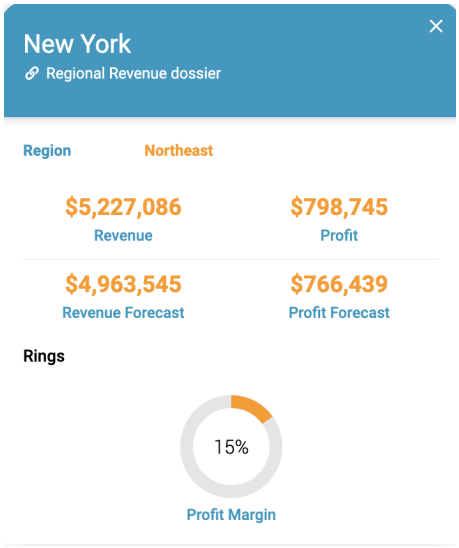
Exercise 4.8: Build and deploy a HyperMobile card

To complete your enterprise mobile solution for BGH, you create a HyperMobile card that displays important KPIs for each of BGH's locations. This allows the regional sales manager as well as the store managers to quickly view up-to-date data on the go.

In this exercise, you:

- Connect to your environment using Workstation
- Build a card in Workstation using the Card Editor
- Certify your card
- Download the HyperMobile app and connect to your environment
- Deploy your card to the app

The image below is an example of your complete card.

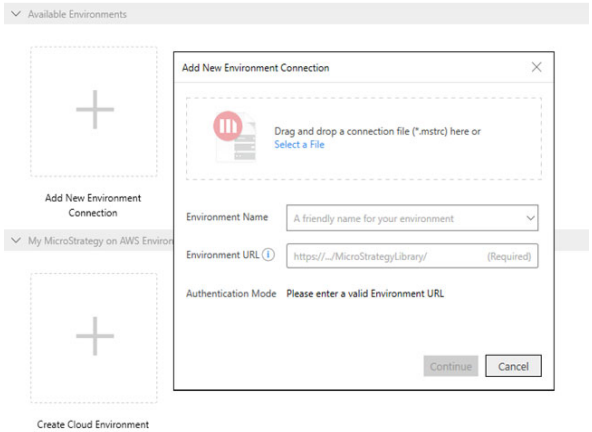


Connect Workstation to your MicroStrategy environment

1 On the remote desktop of your MicroStrategy environment, double-click the **MicroStrategy Workstation** shortcut.

2 In the Navigation panel on the left, click **Environments**.

3 Under Environments, click **Add New Environment Connection**.



4 In the Add New Environment Connection window, enter the following:

- **Environment Name**: MobileArchitect
- **Environment URL:** Type the following URL of the Library Web Server associated with your environment: **https://env-*XXXXXX*.customer.cloud.microstrategy.com/MicroStrategyLibrary**

  Replace **XXXXXX** with your environment number
- **Authentication Mode:** Standard

5 Click **Continue**.

6 In the Connect to Environment window, enter your login credentials. Then select **Remember Me** and click **Connect**.

7 The applications, also known as MicroStrategy projects, available in this connection are displayed. Select **MicroStrategy Tutorial**.

8 Select the **Remember Selected Applications** check box, and click **OK**.

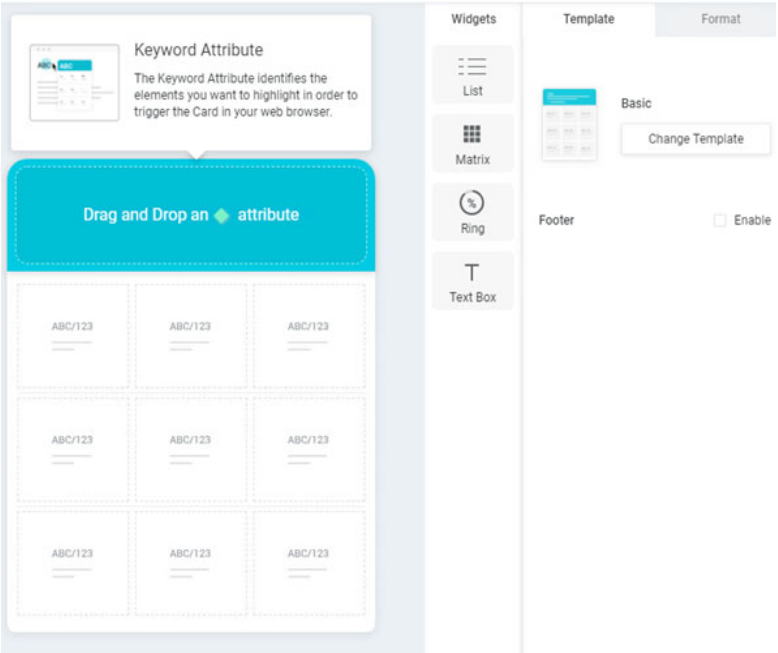
Under Available Environments, your environment should now show a Connected status.

Build a new HyperMobile card for BGH

1 Click **Datasets** in the Navigation panel.

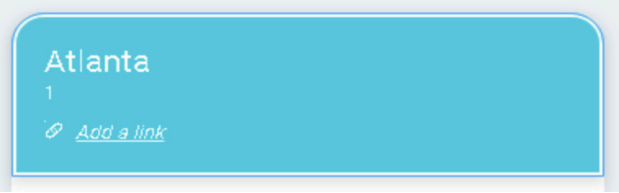
2 Right-click **Intelligent Cube - KPI List**, and select **New Card from Intelligent Cube - KPI List**. The Card Editor opens.

The Card Editor provides flexible options for designing cards. You can mix and match various footers, headers, widgets, and visualizations, and customize formatting to create the best card for your users.



3 The Keyword Attribute is placed in the header and is used to define the card. Think of it as the topic of the card, which can be anything from a product or employee to an insurance claim or policy ID.

Click and drag **Call Center** from the dataset to the card's header. Your header should look like the image below:



4 The header currently displays the description and ID attribute forms of the Call Center attribute, but BGH only wants to display the description. Hover over the header, and select the **Manage Attribute Forms** icon in the upper right.

5 Clear the check box next to **ID** under Subtitles.

6 To complete the header, BGH wants the Regional Revenue dossier linked to the card for easy access to higher level KPIs. Leaving your New Card window

open, navigate to the main Workstation window, and select **Dossiers** from the Navigation panel.

7 Using the search box in the upper right, search for **BGH - Regional Revenue**.

8 Double-click the **BGH - Regional Revenue** dossier to open it.

9 From the **Share** menu, select **Get Link**, and click **Copy**. Click **Close**.

10 Close the BGH - Regional Revenue dossier.

11 Return to the New Card window, and click **Add a link** in the card's header.

12 In the **Text to display** text box, type **BGH - Regional Revenue dossier**.

13 Select the **Link address** text box, and press **Ctrl+V** to paste the dossier link copied in a previous step. Click **Apply**.

14 Click **Save** in the upper left. Expand the **Mobile Architect** environment, and navigate to **MicroStrategy Tutorial -> Public Objects -> Reports -> BGH**.

15 In the Save As text box at the top, type **BGH - Call Center KPIs**, and click **Save**.

Add additional widgets and data to the card

1 On the right of the Card Editor under Widgets, click and drag the **List** widget directly below the header, as shown in the image below.

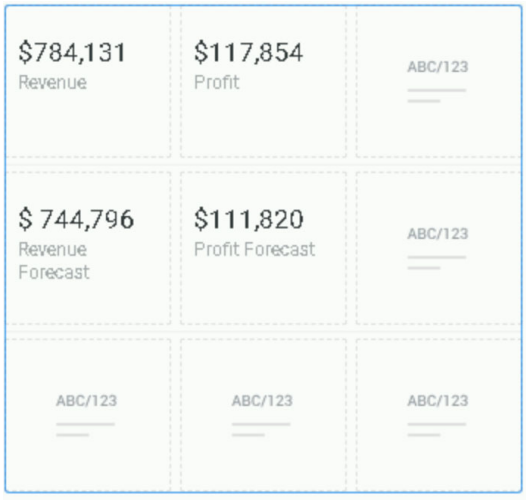
2 Click and drag the **Region** attribute from the dataset to the top row of the List widget.

3 Delete the unused rows, by clicking the **trash can** icon next to each row.

4 Select the **Matrix** widget below the List.

5 Drag and drop the following metrics to the card in the order shown in the image below:

- Revenue
- Profit
- Revenue Forecast
- Profit Forecast



6 Using the **Format** tab on the right, change the Matrix's size to a **2x2** and select the most condensed option under Height.

7 Click and drag the **Ring** widget to the bottom of the card.

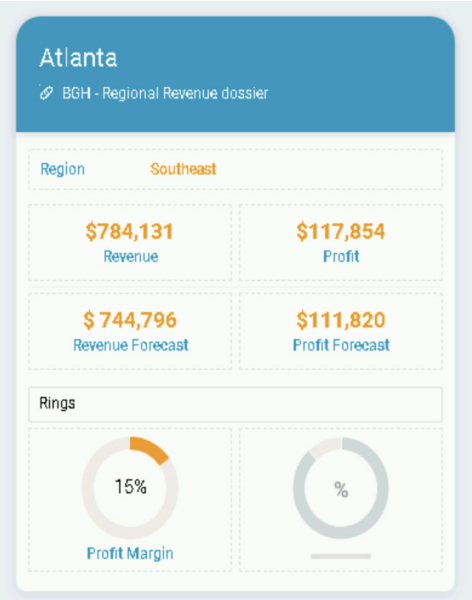
8 Drag the **Profit Margin** metric to the first ring, and save your card.

Format and certify your card

To make the card consistent with BGH's other mobile applications, format the card to be compliant with BGH's corporate colors.

1. Select the card's header, using the **Format** tab change the **Header** color to **#1798C1**.
2. Select the **Link** widget. Change the **Value** color to **#FF9900** and the text weight to **Bold**.
3. Change the **Label** color to **#1798C1** and the text weight to **Medium**.
4. Select the **Matrix** widget. Change the **Value** color to **#FF9900** and the text weight to **Bold**.
5. Change the **Label** color to **#1798C1** and the text weight to **Medium**.
6. Select the **Rings** widget, and expand **Profit Margin** on the Format tab.
7. Change the **Ring** color to **#FF9900**.
8. Change the **Label** color to **#1798C1** and the text weight to **Medium**.

   Your completed card should look like the image below.



9. **Save** and **close** your card.

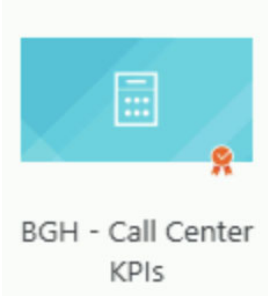
Certify the card

Cards must be certified before users can interact with them in their web browser, Outlook application, and mobile device.

10 Click **Cards** in the Navigation panel on the left.

11 Right-click your **BGH - Call Center KPIs** card, and click **Certify**.

The Certification icon is added, signifying the card contains a governed dataset and has been approved for use across the organization.



Deploy and view your card in the HyperMobile app

1 Click **Environments** in the Navigation panel on the left.

2 Right-click the **Mobile Architect** environment, and select **Properties**.

3 Under **HyperIntelligence for Mobile Configuration URL**, select the check box next to **Show tiny URL**. You use this URL to configure the HyperMobile app to your environment.

Either the full URL or the tiny URL can be used to configure the HyperMobile app. You can also use the Mobile Configuration URL above the HyperMobile link to configure the Library Mobile app.

Download and configure the HyperMobile app

4 From the Apple App Store or Google Play Store on your device, search for and download the **MicroStrategy HyperMobile** app.

5 Once the download is complete, open a browser on your mobile device and type the tiny URL displayed in HyperIntelligence for Mobile Configuration URL in Workstation.

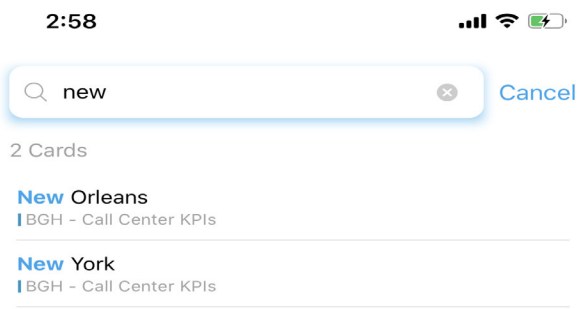
Alternately the configuration link can be either emailed to users or posted on a secure site that they can reach from their mobile device. The user can then tap the link to configure the HyperMobile app on their device.

6 Click **Open** on the alert verifying that you want to open the page in Hyper.

7 Enter your same MicroStrategy login credentials, and click **Login**.

8 Tap **Cards**, and use the toggle to ensure the **BGH - Call Center KPIs** card is on.

9 Tap **Home**, and search for **New**. As you are typing, the cards are being filtered and displayed in the list as seen below.



10 Tap **New Orleans** in the list to view the card.

11 Tap the **Share** icon in the upper right, to send or save the card as an image.

12 Tap the **BGH - Regional Revenue dossier** link to view the data at the regional level.

Summary

In this chapter you created and deployed a complete enterprise mobile solution for Bee Good Health, to include a mobile application viewable in both MicroStrategy Mobile and the MicroStrategy Library Mobile applications and an HyperIntelligence Card viewable in the HyperMobile app. All of the stakeholder are pleased with the solution and excited to start their mobile data analysis.

5

Publish and Manage Mobile Applications

Now that you have standardized the mobile app creation process, set up mobile configurations, and created the BGH mobile app, you can publish your app across the enterprise. With strong enterprise deployment standards, the Mobile Architect can provide effective and scalable app solutions.

In this chapter, we cover:

- Preparing apps for enterprise deployment, such as using TestFlight for beta testing and simulating different mobile use cases to test performance to ensure they meet corporate standards before publication.
- Publishing applications to the App Store and Google Play.
- How to leverage Enterprise Mobility Management solutions in the Intelligent Enterprise for controlled app deployment and security management.
- Upgrading mobile apps to the latest versions and operating systems to leverage new features from MicroStrategy, and avoid common update issues that take away from the user experience.

Ensuring quality and functionality: Mobile application testing

The Mobile Architect is responsible for creating Mobile application testing guidelines to ensure app quality and functionality. There are several tools available for testing. For example, you can use Apache JMeter to load-test functional behavior and measure performance.

To test across multiple devices and operating systems, you can leverage the AWS Device Farm to test and interact with many Android and iOS devices at once, or reproduce issues on a device in real time. View video, screenshots, logs, and performance data to pinpoint and fix issues, as well as increase quality before deploying your app.

Reliable decision making: Testing data integrity

Best Practice

As reliable data impacts decision making and app accuracy, reviewing data within the app to ensure data integrity is critical. By creating a data review process, the Mobile Architect ensures the accuracy and consistency of the app. Testing should be performed on a regular basis, since important data changes over time.

For example, if the app employs Transaction Services, you should test if entered values are successfully saved in the database.

A sample data validation review process is listed below:

1. Verify that you can create, modify, and delete any data in tables.
2. Verify that sets of radio buttons represent fixed sets of values.
3. Verify that a blank value can be retrieved from the database.
4. Verify that when a particular set of data is saved to the database, each value gets saved fully, and string truncation and numeric value rounding do not occur.
5. Verify that the default values are saved in the database, if the user input is not specified.
6. Verify compatibility with old data, old hardware, versions of operating systems, and interfaces with other software.

Ensure functionality and usability: Quality Assurance tests

Apps require testing from every angle to ensure quality and usability, and as the Mobile Architect, you should ensure your team is testing all app functionality before publishing. Functional testing is a Quality Assurance (QA) process that tests basic user interactions with the app such as launching the app and confirming that all buttons, links, information windows, and other functionalities work as desired.

To run functionality tests:

1. Conduct system testing to confirm that the app works as a whole from a user perspective.
2. Conduct unit testing, such as opening an information window, to confirm that individual functions are working correctly.

Best Practice

Unit tests should be done frequently, while function tests should only be performed with a major release.

Intelligent Enterprise standards: Service Level Agreements

The Mobile Architect should verify the performance of the application to ensure the performance is within the Service Level Agreements (SLAs). This ensures the app meets the Intelligent Enterprise standards and corporate guidelines for data analysis and enterprise mobility.

For example, an SLA might establish that when loading the time-off request page of the app, the results must be returned within three seconds. Using a synthetic test environment, the Mobile Architect and team tests this page over both wi-fi and 3G networks. To calculate the load time, the average of the tests is taken.

A sample SLA test is outlined below:

1. Determine the testing client. Either use emulators that imitate a mobile platform or OS environment from the point of view of the application running within the OS installed on a computer, or test using real devices.
2. Create a georealistic testing environment. You should try and create as close to a realistic testing environment as possible.

3 Determine the testing scope:

   a Normal Load: Average number of users visit your website

   b Heavy Load: The maximum number of users visit your website

   c What is your target in this test?

4 Build the testing environment.

5 Run through scenarios to test the app to see if it meets your organization's SLAs.

Exercise 5.1: Test the Hospital Quality app

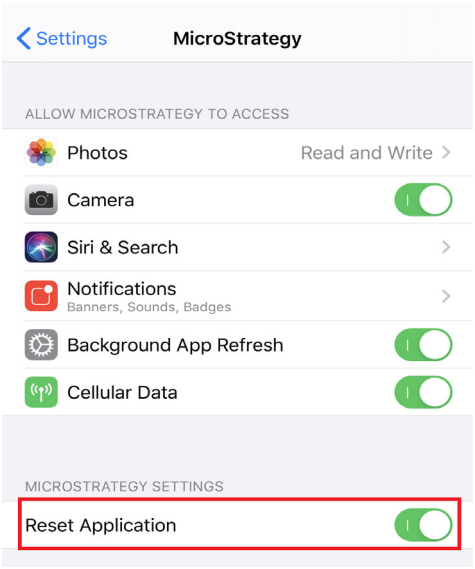
In this exercise, you test the Hospital Quality app to ensure that all design and functionality standards have been met before you deploy the app to your enterprise.

Test the app for design and functionality standards

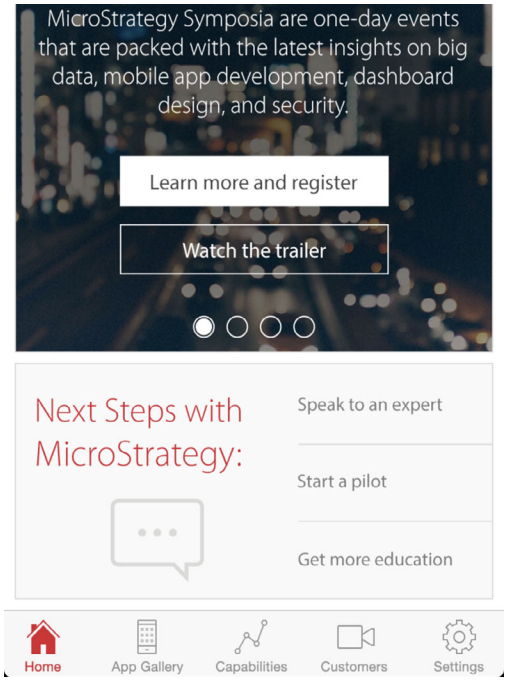
To view the App Gallery on the MicroStrategy Mobile app, reset the app to point to the Demo Server.

1 Open the Setting on your mobile device.

2 Within the Apps sections, locate the MicroStrategy app and select **Reset Application** as shown below.



3 Launch the app on your mobile device.



4 From the MicroStrategy Mobile homepage, click **App Gallery**.

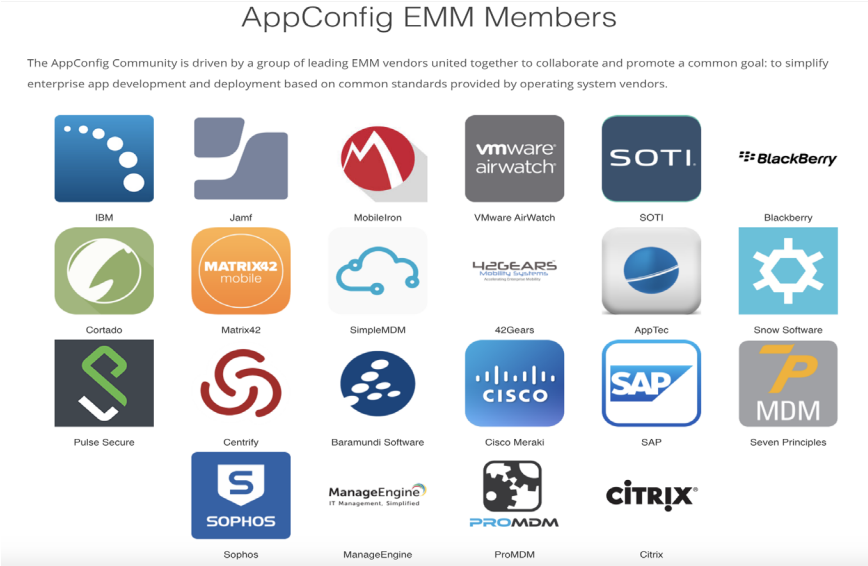
5 Tap **All Apps**.

6 Select the **Hospital Quality** app. The home page opens.

7 Navigate throughout the app. What did the designer do well? What would you do to improve the app?

Integrating with AppConfig-compliant EMM providers

As the Mobile Architect, you should understand how MicroStrategy Mobile works with EMM solutions, as well as the benefits and considerations of different EMMs.



MicroStrategy Mobile offers native SDK integration with any EMM provider that supports AppConfig guidelines, including the following leading EMM providers: AirWatch, Blackberry Dynamics, and MobileIron. App Configuration (AppConfig) for the Enterprise delivers a standard approach to configuring and securing enterprise applications. AppConfig allows enterprises to leverage their existing investments in EMM systems, VPNs, and identity management solutions.

As the MicroStrategy Mobile app is AppConfig compliant, most enterprises choose to use SDK for either iOS or Android. However, there are certain EMM-specific features that are only available with an integrated app.

Developers can use app wrapping through an EMM, which applies custom security policies. However, as the EMM vendor owns the wrapper code, MicroStrategy does not have access to code and cannot debug or fix any issues that might occur when using a wrapper.

Regulating devices and apps: MDM and MAM

Today's enterprise employees are increasingly expected to be productive wherever they may be, on whatever device they choose, which raises multiple management and security issues. An Intelligent Enterprise should tackle those

issues and provide a way to ensure employees can do their work while preventing them from violating corporate policies. Working with your colleagues in the Intelligence Center, plan and develop EMM solutions for different scenarios.

Mobile Device Management

Mobile Device Management (MDM) is a device life cycle management technology that enables your IT team to deploy, configure, manage, support, and secure mobile devices through MDM profiles installed on the devices. MDM software provides asset inventory, over-the-air configuration of email, apps, and Wi-Fi, remote troubleshooting, and remote lock and wipe capabilities to secure the device and the enterprise data on it.

MDM focuses on controlling the entire device and requires that users enroll their device and install a service agent. Users with their own personal devices can opt-into MDM, allowing access to corporate services, such as email and calendar. When a user enrolls their device to the MDM, they are typically provided with information about what the MDM can and cannot do to their device. Users who aren't comfortable with their organization having any access to their device can opt-out at any time.

Although this solution is comprehensive, there are some potential issues to consider. Employees typically carry two devices, one at the enterprise level and another with personal data. Another drawback is that policy is configured at the device level. MDM may not be the right choice for enterprise policies that are only applied on selective applications or to application-specific data as opposed to the device.

The Mobile Architect is responsible for creating an MDM strategy. For example, you may decide that as an MDM policy, an employee's personal device is wiped after an employee loses it or leaves the company.

Best Practice

MDM best practices include:

- If there is a BYOD policy, you should support multiple device platforms.
- Periodically analyze reports to gain insight to make sure MDM addresses how users are using their mobile devices.
- The process should be simple for users and scalable across IT support.
- Provide a backup and recovery service.

Mobile Application Management

Many employees are expected to use their personal devices in the workplace. BGH is no exception as the BYOD corporate culture has become a challenge to security managers and administrators requiring strict fraud prevention programs. You want to streamline operationally efficient security policies in the BYOD environment.

To give employees the freedom to use their personal devices while controlling access to internally developed apps or a group of apps, Mobile Application Management (MAM) provides granular controls on the app level that enable the Mobile Architect to manage and secure app data.

Using MAM solutions, the Mobile Architect can distribute, secure, and track mobile applications directly from the MAM console, which can be integrated into your mobile environment.

Exposing iOS Library SDK APIs to support third-party EMM SDK integration

In many enterprises, employees use their personal mobile device to access company applications, such as the Library Mobile app. The management of mobile apps deployed to non-company-owned devices can be complex since it requires control over the employee's device. The integration of third-party EMM providers, such as Microsoft Intune, allows enterprises to manage the Library Mobile application itself without requiring full control over the device.

What scenarios would you use MAM? MDM?

Distributing apps to end users: Deployment methods

MicroStrategy Mobile is a highly customizable app development platform that gives organizations the flexibility to deploy their mobile app in a variety of ways. As the Mobile Architect, you are responsible for creating the guidelines for deploying mobile applications, whether to an app store, internally to enterprise users, or manually. Before deploying your app, review the following deployment prerequisites.

App deployment prerequisites

Before deploying an app to your enterprise, the Mobile Architect is responsible for creating and ensuring all iOS and Android deployment prerequisites have been met. This ensures the application deployment process runs smoothly on iOS and Android devices.

Sample prerequisites are below:

1. Enroll the organization in the Apple Enterprise Developer or Google Developer Program. This is required for deploying all iOS applications to the App Store, MDM, or internal Store.
2. Download the MicroStrategy Mobile SDK for each mobile operating system (iOS or Android).
3. Use Xcode or Android Studio to create or customize the files you need to build the application. For iOS applications, you must use a Mac running the version of Xcode supported by your version of the MicroStrategy Mobile SDK.
4. Create a distributable archive of the application.

Best Practice

Best practices for enterprise deployment prerequisites include**:**

- To maintain control of your enterprise's MicroStrategy Mobile implementation cycle, it is recommended that the Mobile Architect uses the Enterprise Deployment process to install the MicroStrategy Mobile applications on your users' devices.
- You need a location on your network that users can access through their web browsers, using either the HTTP or HTTPS protocol. The distributable archive must be saved to this location.

User testing: Beta deployment

To test for issues, bugs, and overall user experience, the Mobile Architect should implement a beta deployment policy for all mobile applications. For example, you can require that all iOS apps must be tested through TestFlight and Android apps using .apk files by at least five users for one week. Then those users submit feedback, your team reviews the feedback, and follows up with the test users for clarifications before deploying the app to an enterprise-wide audience.

To plan a beta deployment policy, consider the following:

1. Define the range of platforms that are going to be included in beta tests; the variations include device iterations and operating systems. Do you want to require that apps are tested on only Android? Tested only on phones or tablets?
2. Ensure chosen beta testers have access to the selected platforms.
3. Decide what features are to be tested and whether users should test the app in its entirety or certain documents?
4. Determine the expected outcome of the app.
5. Determine the distribution software for the beta app, such as TestFlight and Google Play Developer Console.
6. Select a mechanism for users to provide feedback.
7. Require that app developers also study implicit feedback, such as usage analytics and crash reports.

Enterprise Deployments

As done in a previous chapter, you manually downloaded the out-of-box MicroStrategy Mobile app and configured it using the Mobile Configuration URL link. Alternatively, the app can be pre-configured or customized before being made available to end users. The Mobile Architect leads and creates guidelines for the app distribution process. A pre-configured, customized enterprise mobile application can be distributed using the following methods:

- Internal website
- App Store
- Enterprise Mobility Management system

Below are key app distribution workflows the Mobile Architect should leverage and some considerations in choosing a distribution method.

Deployment method: Internal website

The Mobile Architect can configure a web page that contains installation links to the archive file (.apk or .ipa) to allow users to install enterprise applications on their device. The installation link can then be shared to users via a QR code which the user scans with their device, or included in an internal port or website.

Pros	Cons
• An EMM solution isn't necessary, however EMMs can be leveraged in this process.	• No control over installs and upgrades as user must manually download the app.
• Limit download access through secure email or web page.	• iOS apps internally deployed over a web server require enterprise developer account provisioning profiles to be installed on the devices.
• Less invasive for BYOD users as they do not need to enroll their device in an MDM system.	

Deployment method: App Store

Using a public app store such as Apple's App Store or Google Play, enterprises can distribute their customized apps to the public. The Mobile Architect is responsible for the app store submission and policies. For example, you should implement a framework for submitting apps to both Google Play and Apple's App Store.

Pros	Cons
• Easily distribute external apps as the store is available to the public.	• No control over installs and upgrades, users must manually download the app.
	• Requires a wait time for publishing approval from Android or Apple.

Guiding the submission process: Apple's App Store

Once performance testing is completed, the Mobile Architect is responsible for overseeing the publishing of enterprise applications on the Apple App Store. The steps below outline how to publish an app to the Apple App Store.

1. Prepare your developer account. Sign up for a developer account and check that your developer account details are accurate. If you're going to sell products, set up your merchant account.

2. Choose your build. Each app can have multiple versions, and each version can have multiple builds. To publish your app on the App Store, choose which build to submit to review.

3. Set pricing and availability. You must set a price for your app and can select specific territories for your app. Your app is available in all App Store territories by default. You also have the option to publish your app as a pre-order.

4. Submit your app for review. You submit your app for review to start the App Review process and to make your app available on the App Store. Before you submit an app, enter all the required metadata and choose if you want to release your app manually or automatically, or if you want to release your app in phases.

5. View your app status and resolve review issues. After you submit the app, the app status changes to Waiting for Review. If there are any issues with your app, read and reply to App Review communications. After your app is approved, it can take up to 24 hours to go live on the App Store.

6. Request promo codes. After your app is approved, you can request promo codes to distribute to users before you make your app available on the App Store. You can distribute the promo codes by email or other means, and the user enters the promo code when purchasing the app.

Guiding the submission process: Google Play

As with Apple App store submissions, the Mobile Architect is responsible for ensuring standards are met and apps are correctly published on the Google Play store. The steps below outline how to publish an app to the Google Play store.

1. Understand the Developer Program policies. The Developer Program policies are designed to ensure that the Play Store remains a trusted resource for Android users. Review the policies thoroughly as there are consequences for violations.

2 Prepare your developer account. Sign up for a developer account and check that your developer account details are accurate. If you're going to sell products, set up your merchant account.

3 Plan for localization**.** If you plan on including localized copies of your app at launch, start planning early and follow the Localization checklist.

4 Plan for simultaneous releases. Releasing your app on multiple platforms and devices maximizes your promotion activities and the number of installs, so include it in your development plans upfront. If you can't launch your app on all platforms at once, ask for users' contact details and let them know when your app is ready

5 Test against the quality guidelines. Quality guidelines for all apps plus specific criteria for tablet, TV, and Auto apps provide testing templates. You use these to confirm that your apps offer the basic UI design, features, and functionality expected by Android users.

6 Build a release-ready APK. When you're ready to make your app available to users, either for testing or as a final product, prepare the APK with basic code cleanup and optimization, building and signing with your release key, and final testing.

7 Plan your app's Play Store listing**.** Prepare the descriptions, promotional graphics, screenshots, and videos you'll add to your app's Play Store page. Make sure you include a link to your privacy policy if required. Localize your store listing in all the languages your app supports and for the countries you're targeting.

8 Define your app's device compatibility. Let the Play Store know which Android versions and device screen sizes your app is designed to work on.

9 Set up your app's price and countries of distribution. Once you've determined your monetization model, setup your app as free or paid and select the countries in which it is going to be distributed.

10 Opt-in to the right distribution options. From the Pricing & Distribution page, opt-in to specific devices and programs, such as Android Wear, Android TV, and Designed for Families. Google Play can then review your app and, once approved, make it more discoverable for users.

11 Determine your app's content rating. Providing an appropriate rating for your app is a requirement of the Developer Program Policies and it also ensures your app gets seen by the right age-based audiences.

12 Final checks and publishing. First, go back and double check you've done everything on this list. Now you're ready to publish your app to the production channel. If you're releasing an app update, use staged roll-outs to release your update to progressively more users. This allows you to halt the update if you find an issue, so you can limit the number of users it affects.

Deployment method: Enterprise Mobility Management

Enterprises can use an EMM system to manage the deployment of mobile applications to their end users. Managed apps can contain sensitive information and deploying through an EMM provides more control than apps downloaded manually by the user. A Mobile Device Management server can remove managed apps and their data or remove apps when the MDM profile is removed.

For managed app distribution, an MDM solution can push apps to both managed (BYOD) and supervised (corporate) devices. The EMM administrator uploads the app binary (.apk or .ipa) to the EMM system, defines the users and groups to receive the app, then sends installation requests to MDM-enrolled devices from the administrative console. The EMM administrator can select as many devices as needed and push several mobile apps to them all at once. The EMM system can notify the EMM administrator of the users who received the application and those who did not.

Apple offers enterprises Volume Purchase Program (VPP) which allows EMM administrators to deploy and configure applications hosted on the Apple App Store directly on the users device. For example, an EMM administrator can push paid applications such as Keynote to the user's device. As a benefit, end users do not have to use their personal Apple ID on a supervised (corporate) device to have paid apps used by the enterprise.

Google provides similar functionality to Apple's VPP program via their Managed Google Play store. Additional information for the Managed Google Play store can be found at the Google Play support site.

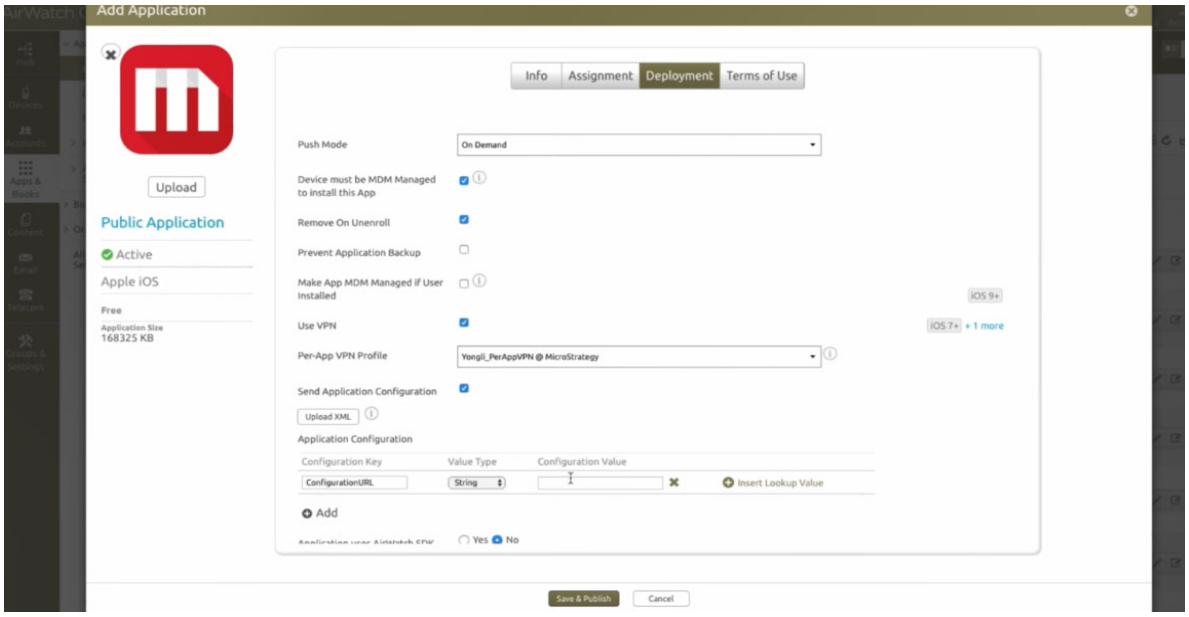
The following are important considerations for MDM deployments:

Pros	Cons
• Provides control over app installs and upgrades, ensuring users have the enterprise apps they need.	• Devices must be managed or supervised within the EMM system, which may not be preferred for BYOD users.

Pros	Cons
• Apps can be pushed to the appropriate user groups in bulk.	• An EMM solution is necessary to manage the MDM.
• All apps are managed in one console, standardizing app distribution.	
• Includes reporting tools for identifying users who received, or did not receive, apps.	

Demo: EMM deployment

In this video, the Mobile Architect uses AirWatch to deploy MicroStrategy Mobile without any customizations to the iOS App.



Governing the mobile workforce: Enterprise Mobility Management

Intelligent Enterprises can use Enterprise Mobility Management systems to govern various aspects of workforce mobile devices, applications, and services; such as updates, access settings, device restrictions and password protocols. EMM goes beyond traditional device management to include the management and configuration of enterprise apps and content.

As each business is unique, so is the implementation of an EMM solution. While some businesses require strict control over the device, application, and content, others may only need control over application and content. The Mobile Architect is responsible for overseeing and setting the standards within the enterprise's EMM solution, such as:

- Determining security restrictions and access controls applied via EMM, such as requiring a device passcode or disabling copy and paste to prevent data leakage.
- Regulating MicroStrategy predefined configuration keys that are defined within the EMM policy to push to user devices.
- Establishing app distribution policies through EMM, for example if the app is pushed to the device or if users download the app from an internal app store.

Best Practice

- Staying up to date on EMM developments by checking with EMM vendors regularly and subscribing to the AppConfig newsletter to watch for critical updates and other important EMM information.

Securing devices and apps with EMM

Data on a mobile device is vulnerable, as a device can easily be lost or stolen. When working with an EMM solution, the Mobile Architect determines which security policies should be applied at both the app and device level to ensure the governance of EMM complies with corporate security standards. For example, should employees be allowed to download any application? Should they be allowed to use voice commands, such as Siri? Among other important security measures, EMM solutions allow the enterprise to remote lock and wipe devices to protect the data.

Best Practice

The Mobile Architect should consider the following best practices when implementing security via an EMM solution:

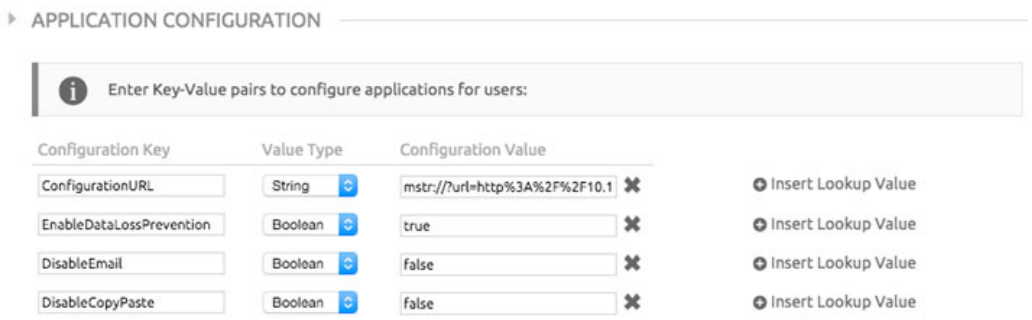
- Provide a method to wipe the device in the event it is reported lost or stolen. For example, on iOS devices you can enable Managed Lost Mode. When Managed Lost Mode is enabled, the user can't unlock the device until an administrator switches the mode off.
- Enforce a strong, complex device password and a low maximum number of failed login attempts to prevent non-approved users from logging into the device.
- Enforce encryption of the device, including removable storage and mobile backups.
- To prevent data leakage, block users from taking screen shots.

- Restrict users from using cloud backup on corporate apps, once corporate data is in the cloud the organization no longer has control of it. If a user leaves the company, sensitive data may still be in the user's personal cloud account.

Custom security settings

In addition to the mobile configuration specified by ConfigurationURL, the Mobile Architect can require the EMM administrator to restrict the availability of some app features to prevent data leakage or data loss within the MicroStrategy Mobile app. These restrictions are passed to MicroStrategy Mobile by adding key-value pairs to the Application Configuration section on the Add Assignment screen. The following restrictions can be added as BOOLEAN key-value pairs:

```
EnableDataLossPrevention

DisableEmail

DisableOpenIn

DisablePrint

DisableCopyPaste

DisableCameraAccess

DisableLocationServices

DisableSaveToPhotos
```



The EnableDataLossPrevention key is the main switch for all of the app restrictions. Other options, such as DisableEmail or DisableCopyPaste, take effect only when EnableDataLossPrevention is set to true.

When EnableDataLossPrevention is set to true, the following features are automatically disabled:

- Open PDF in third-party applications
- Open email URL in third-party applications

- Open telephone, SMS or unknown URL in another application
- Open EPub files in third-party applications
- To verify the security settings in MicroStrategy Mobile, go to Settings -> Logging and set the level to Messages. When you view the log, you can see the security settings.

Standardizing mobile configuration and security: AppConfig

Using AppConfig-compliant vendors for EMM deployments helps Intelligent Enterprises by delivering a standard approach to configuring and securing apps. Guidelines set by the AppConfig Community ensure that EMM systems configure and secure mobile applications by leveraging app security and configuration frameworks available in the OS.

The AppConfig framework provides the following workflows for configuration, security, and access:

- **App configuration** - Configuration information such as Intelligence Server connectivity, project information, home screen configuration, and general app settings to eliminate the need to educate end users about first time setup.
- **Security policies and access control** - Restrict apps to run only on approved devices and enforce security policies such as required encryption and data loss prevention at the app level.
- **App Tunnel** - Selectively enable approved apps to use an app tunnel to connect to backend and corporate networks.
- **Single sign-on (SSO)** - MicroStrategy supports single sign-on through SAML and HTML Form.

Apps like MicroStrategy Mobile that are AppConfig-compliant allow EMM administrators to remotely configure applications directly on the user's device after the application has been installed. The AppConfig model allows enterprises to explicitly control the data and security of the application without requiring any additional steps from the user and eliminates the need for any proprietary SDK or app wrapping to ensure the app is secure.

The benefits to using AppConfig are:

- Provides an EMM vendor-neutral solution and helps to deploy enterprise-ready apps faster.
- Eliminates the need for proprietary SDK or app wrapping.

- Helps enterprises to use existing VPN solutions and leverage existing EMM investments.
- Apps can be restricted to run only on approved devices and enforce security policies such as required encryption and data loss prevention at the app level.
- Can easily migrate from one EMM vendor to another.
- Takes advantage of updates MicroStrategy publishes on a quarterly basis.
- Up-to-date with Apple and MicroStrategy security updates.

Consider the following when choosing an AppConfig-compliant vendors:

- Limited feature support through the EMM vendor.
- Typically requires a more expensive licensing structure needed to support AppConfig by the vendor.
- Geared towards MDM instead of application and content management.
- Users need a security profile, which users might not want to install on personal devices.
- Needs to have a VPN app tunnel that pairs with the AppConfig app.

Communicate with the Intelligence Server: App configuration

The Mobile Architect defines the sets of configuration keys that the MicroStrategy Mobile app accepts from the EMM server. These configuration keys are placed in the EMM administration console, normally stored as part of a profile assigned to the app for deployment. The EMM provider can also update the configurations at any point to an existing application, without requiring the app itself to be reinstalled.

The Mobile Architect should determine what profile assignments should be created to deploy the application to different groups of devices by applying different configuration settings to each. For example, managers can be assigned to receive an app that allows them to approve expenses and time off.

One of the configuration keys that can be pushed to the MicroStrategy Mobile app is the ConfigurationURL, which is a link to a full set of configuration settings such as Intelligence Server and Mobile Server connectivity, project information, home screen configuration, and general app settings.

For example, to set up app configuration on the AirWatch console:

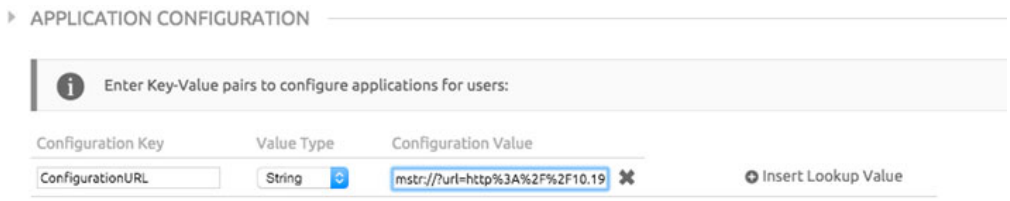
1 Upload MicroStrategy Mobile to the AirWatch console.

2 Click **Save & Assign** to assign the application to devices.

3 On the Add Assignment window, select the device group to specify which devices MicroStrategy Mobile is going to be delivered to; and set the delivery time to control when it is delivered.

4 To initialize the application with a configuration URL, set a key-value pair in the Advanced, Application Configuration section. Add a ConfigurationURL key with a String value type.

5 Add the Configuration Value, which is the mobile configuration link created in the Mobile Administrator page.



Securely connect apps to corporate networks: App Tunnel

When using an AppConfig vendor, an application may require access to web services residing behind a corporate firewall, which requires a secure App Tunnel. An App Tunnel selectively enables approved apps to use an app tunnel to connect to back end and corporate networks. For example, AirWatch can distribute VPN profiles to its managed devices and allows MicroStrategy Mobile to set up VPNs following the profiles.

The following steps are provided for reference only, and are not intended to be performed in class.

To enable per-app VPN

1 Create a device profile with AppProxy VPN enabled, and assign it to your device. For example, on the AirWatch console, you configure and distribute the VPN profile in **Devices**, then **Profiles**, then **iOS**, then **VPN**, where you specify the domains and host names in the profile to auto-trigger the VPN.

2 Configure the VPN in the device profile. For example, on the AirWatch console, go to **VPN**, then **Configure**, and set the **Connection Type** to **AirWatch Tunnel**.

3 Publish the device profile to your device groups.

After you have assigned the device profile to your device, you can see all the app-layer VPN settings in **Settings**, then **General**, then **Profiles & Device Management,** and then **Mobile Device Management**.

4 Configure the Per-App VPN Profile for MicroStrategy Mobile and allow it to use the VPN by choosing its device profile in Per-App VPN Profile. Click **MicroStrategy Mobile** in Apps and Books, then select **List View**. Click **Assign**, and edit the assignment.

5 Install AirWatch Tunnel to the device and open AirWatch Tunnel.

These settings take effect after MicroStrategy Mobile is pushed to the device. When MicroStrategy Mobile is launched, it automatically connects to the VPN server, and the VPN icon is shown on the left side of the status bar.

Upgrading and updating Mobile SDK

MicroStrategy releases regular updates to MicroStrategy Mobile SDKs for new features and functionality to MicroStrategy Mobile applications. The Mobile Architect should require that enterprise apps are up-to-date with the latest version to take advantage of new capabilities.

iOS

If your customizations are based on the MicroStrategy Mobile project, they are not affected by the new file structure and SDK framework. Once the base application is working, you can migrate your customizations and adjust them if necessary to the new version of MicroStrategy Mobile.

Android

Start by making sure the base MicroStrategy Mobile application runs without issues. Follow the Android Studio recommendations for updating components regularly to keep your IDE current.

Once the base application runs without problems, start migrating your customizations to the new version. It is recommended to migrate changes increasingly to narrow problems only to the latest change instead of doing a bulk migration and making the task of debugging much more complex.

Evaluating changes: Regression testing

Regression testing makes sure that where a small change is made, the overall flow of the Mobile Application still works as expected once the development of the feature is completed. The Mobile Architect should require that designers thoroughly test their apps when any underlying component changes, to proactively address any issues; this prevents users from experiencing those issues and then discontinuing app use.

Steps to run regression testing are outlined below.

1. Run the impacted application.
2. Run through all functional aspects of the application to test for issues due to object changes.
3. If an error occurs, work with the Application Administrator to fix the underlying issue.

Best Practice

Best practices to perform regression testing include:

- Review schema and application level changes with the Analytics Architect to understand what has changed, and to analyze any potential impacts to your mobile applications.

- Ensure SLAs are met after changes have been applied to the Mobile applications.
- Test all components and ensure functionality works as intended.
- Perform tests on all supported devices and OS versions your app is going to be used on.

Storing and tracking revisions: Source code control

The source code for the customized MicroStrategy Mobile applications and the custom applications are assets for the Intelligent mobile team. This work should be kept easily manageable during a migration to a new version.

You should invest in a source code management system that is able to compare files to highlight differences in a version upgrade.

Back up source code

At the end of a development cycle, where you published customized MicroStrategy apps and custom apps, a backup process must be set in place to trace the source code evolution at this specific point in time. As the Mobile Architect, work with the System Administrator to archive the source code and label it with the correct version number. Make certain you archive the following:

- Source code for each app
- Design documents of enhancements present in these apps

Upgrade check list

Establish a standardized check list for your team to follow during an upgrade. The following elements should be part of that list.

- With a new version of the base MicroStrategy Mobile app, you must first load it into your development environment and successfully compile a version pointing to the demo data. This standard test ensures that any architecture and libraries changes are addressed.
- Point the app to your data environment. Having MicroStrategy Tutorial appear is enough to confirm accurate connectivity.
- Using a file compare approach, adjust the files within the new MicroStrategy Mobile app to match a change you made in the previous version. Some

tweaking might be necessary to make the best of a new feature or replace a deprecated API call, for example.

- Make certain all the changes behave as before within the app. A thorough QA cycle is important at this point.
- If your users need the app immediately, you can publish it as the QA cycle ensures the application behaves as before.
- Your team can now add new functionality. At this point, you have a solid base to work with to add new customizations. This becomes a regular internal development cycle as the new MicroStrategy Mobile version has been integrated.

6

Monitoring and Troubleshooting

An Intelligent Enterprise consistently monitors performance across the Intelligence program to maximize efficiency and utilization of all BI applications. As the Mobile Architect, you are responsible for ensuring your team is tracking and monitoring data such as mobile app usage statistics, crash logs, and Mobile Server reports.

In addition to monitoring logs, your team is responsible for troubleshooting mobile application issues. As the Mobile Architect, you provide troubleshooting technical leadership and support. You can use these logs and statistics during meetings with the Intelligence Center to report on progress within Enterprise Mobility.

In this chapter, we review:

- Monitoring important design-related key performance indicators (KPIs) such as percentage of error rate and average load time.
- Setting up diagnostics, statistics, and reporting to track performance.
- Guiding your team in troubleshooting app issues to support business users.

Track user engagement: Key performance indicators

By monitoring user engagement KPIs, your team can evaluate if the app is adding value, and if user engagement is increasing or decreasing over time. KPIs also identify which device and operating system combination is being used most frequently to properly allocate resources to developing the user experience.

To track the success of enterprise mobility, installations, user session counts, user session durations, mobile application deletions, device types, OS and version, and device models, data should be incorporated into a dossier analyzing these KPIs.

What KPIs would you track? Why?

Exercise 6.1: Build and analyze the Mobile Usage dossier

To streamline monitoring all your enterprise applications in the Intelligent Enterprise, you want to set up a dossier that tracks key performance indicators. In this exercise, you view and edit the Mobile Application Usage dossier and analyze the results.

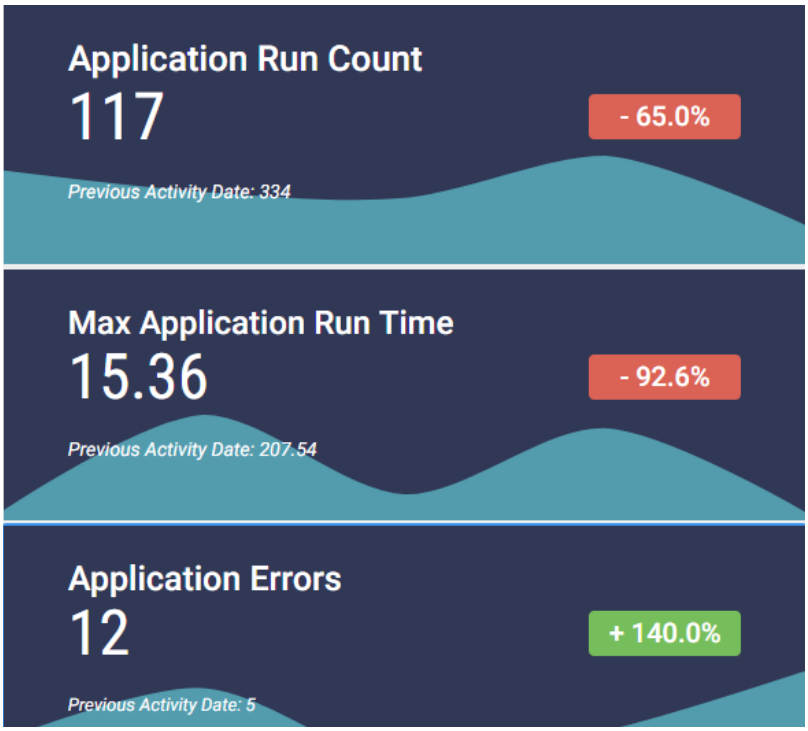
Add the Mobile Application Usage dossier to MicroStrategy Web

1 From the Shared Reports folder, click **My Reports** on the menu on the left.

1 Click **Create**, then select **Upload MicroStrategy File**.

2 Add the **MobileApplicationUsage.mstr** file your instructor sent you in the beginning of class. Then click **View Dossier**.

Analyze the KPIs page



1 The Application Run Count visualization displays the number of times all enterprise apps were run by enterprise users. What does the drop off in execution the past few days show?

2 The Max Application Run Time visualization displays the longest run time for an enterprise app. There's a large dip in run time, what does this mean for the apps?

3 The Application Errors visualization displays the number of app errors. What does this increase show?

Add the Application Error Rate visualization

There is one more visualization you would like to add to fully monitor the enterprise apps: Application Error Rate. This visualization shows the percentage of errors over time.

1 Click the **Menu** icon on the Application Error visualization and select **Duplicate**.

2 Replace the Application Error metric with the **Application Error Rate** metric in the new visualization.

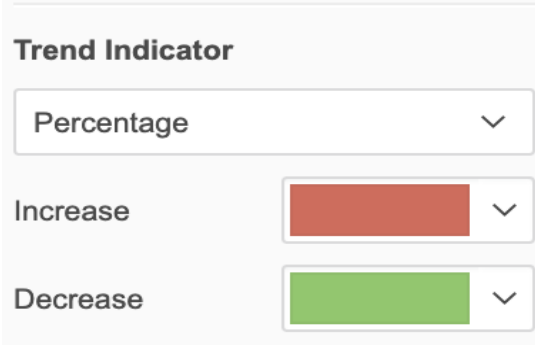
Invert the KPI threshold colors

KPI widgets automatically display positive growth as green and negative growth as red. However, with the Mobile Application Usage dossier, positive Application Errors and Application Error Rate are unfavorable. You want to invert the threshold colors of the two KPIs.

1 Select the **Application Error** KPI, then click the **Format** panel.

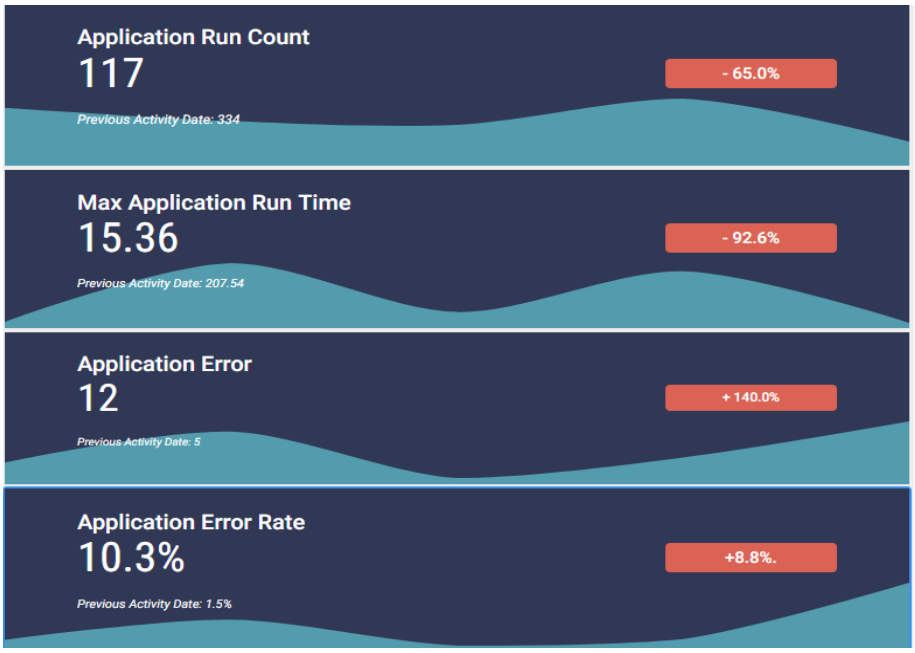
2 In the first drop-down list, select **General Settings**.

3 At the bottom of the panel, set the **Increase** color to **Red** and the **Decrease** to **Green**.



4 Repeat steps 1 through 3 for the **Application Error Rate** visualization.

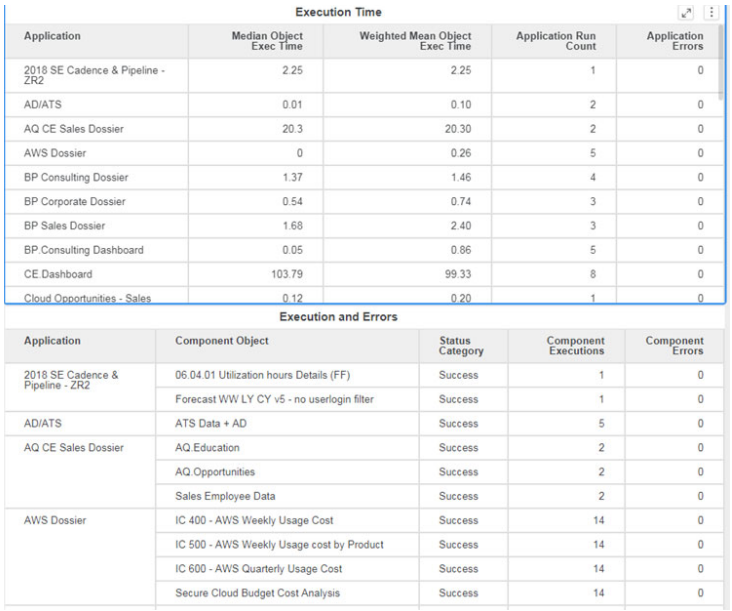
5 Your dossier should now look like the image below. What does the Application Error Rate visualization tell you? Analyze this dossier page to explain each of the visualizations and their correlations.



6 Click the **Save** icon to save your work.

Add a filter and analyze the Executions page

1 From the Table of Contents, click the **Executions** page.



Execution Time

Application	Median Object Exec Time	Weighted Mean Object Exec Time	Application Run Count	Application Errors
2018 SE Cadence & Pipeline - ZR2	2.25	2.25	1	0
AD/ATS	0.01	0.10	2	0
AQ CE Sales Dossier	20.3	20.30	2	0
AWS Dossier	0	0.26	5	0
BP Consulting Dossier	1.37	1.46	4	0
BP Corporate Dossier	0.54	0.74	3	0
BP Sales Dossier	1.68	2.40	3	0
BP Consulting Dashboard	0.05	0.86	5	0
CE Dashboard	103.79	99.33	8	0
Cloud Opportunities - Sales	0.12	0.20	1	0

Execution and Errors

Application	Component Object	Status Category	Component Executions	Component Errors
2018 SE Cadence & Pipeline - ZR2	06.04.01 Utilization hours Details (FF)	Success	1	0
	Forecast WW LY CY v5 - no userlogin filter	Success	1	0
AD/ATS	ATS Data + AD	Success	5	0
AQ CE Sales Dossier	AQ Education	Success	2	0
	AQ Opportunities	Success	2	0
	Sales Employee Data	Success	2	0
AWS Dossier	IC 400 - AWS Weekly Usage Cost	Success	14	0
	IC 500 - AWS Weekly Usage cost by Product	Success	14	0
	IC 600 - AWS Quarterly Usage Cost	Success	14	0
	Secure Cloud Budget Cost Analysis	Success	14	0

The Execution Time visualization displays how long objects in an app take to run and how many times the app has been accessed.

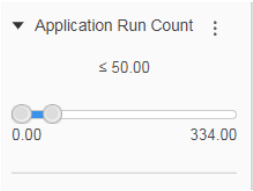
The Execution and Errors visualization displays component objects, such as dossier pages, whether they were successfully loaded, how many times they were accessed, and number of errors.

2 You want to view applications with a low run count to see which apps are being run the least and strategize how to increase utilization. Select the **Filter panel** to add a filter to the chapter.

3 Click and drag the **Application Run Count** metric to the Filter panel.

The slider filter is added to the chapter and applies to all pages in the dossier.

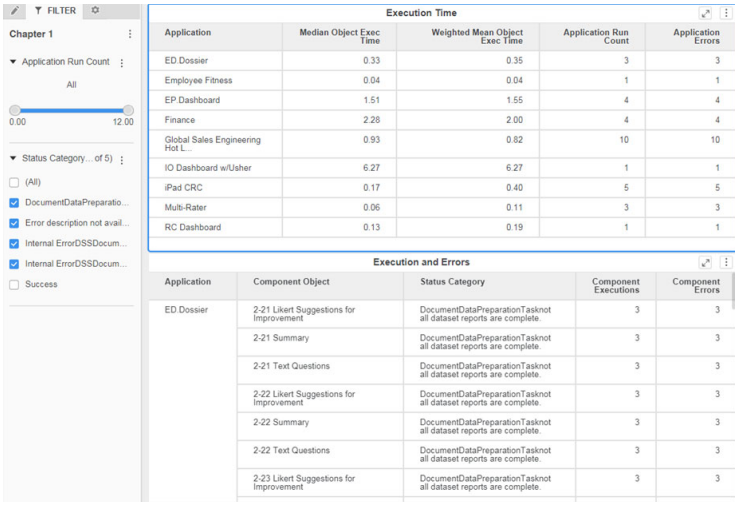
4 Click and drag the **slider filter** to the left so only those apps with 50 or fewer run counts are displayed.



5 Next, you want to view those applications that had components that did not run successfully. Click and drag the **Status Category** attribute to the Filter panel.

6 Clear **Success** to view those components that loaded with errors.

Your dossier should look like the image below. You can now direct your team to those applications with errors to help de-bug the issues.



7 **Save** your work.

8 Clear the filters by clicking the **Menu** icon next to **Chapter 1** in the Filter panel, and select **Unset All Filters**.

The filters in the Filter panel applies to all pages, and you want to view all data for the Users page.

Organize and analyze the Users page

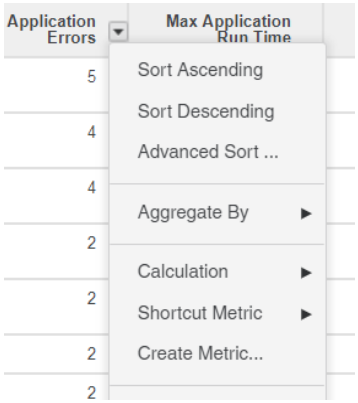
1 From the Table of Contents, click the **Users** page.

This page displays statistics of individual users, such as what applications they're accessing and how many errors they've encountered.

User Name	Application	Application Run Count	Application Errors	Max Application Run Time	Median Object Exec Time
Adah Valdez	BP Corporate Dossier	1	0	1.15	1.15
Aiko Tillis	Finance	14	1	14.65	0.1
	iPad CRC	2	0	0.48	0.03
Alayna Maitland	iPad CRC	3	0	0.05	0.03
Alethia Claybrooks	iPad CRC	1	0	0.04	0.04
Angel Pixler	Sales	6	0	2.54	0
Anisa Kleinschmidt	BP Corporate Dossier	2	0	0.59	0.48
	BP Consulting Dashboard	2	0	2.14	2.03
	CE Dashboard	1	0	140.87	140.87
	ED Dossier	2	2	0.43	0.38
	Finance	1	0	4.31	4.31
	iPad CRC	3	0	0.07	0.03
	Marketing	4	0	6.45	0
	Multi-Rater	1	1	0.06	0.06
	Sales	11	0	2.64	0.13

You want to sort the grid by Descending based on number of errors. From there, your team can reach out to the users who encounter the most errors and schedule interviews to learn more about the errors they're encountering and work to enhance their user experience.

2 Right-click **Application Error**, and select **Sort Descending**.



3 **Save** your work.

Proactive problem solving: Diagnostics and statistics

Tasks performed by MicroStrategy Mobile users may generate errors, warnings, or messages. This information can be stored in log files, that can be aggregated for analysis. A best-in-class Mobile Architect should require the mobile team to frequently reviews logs to proactively locate potential issues and troubleshoot user-reported problems. In the exercises below, we review setting up and analyzing diagnostics and statistics for the Mobile Server and applications.

Exercise 6.2: Set up diagnostics

To establish diagnostics, define what information is logged by the MicroStrategy Mobile Server, as well as where it is logged. MicroStrategy has an interface to view the logged information. As the Mobile Architect, you should set up diagnostics for your mobile team.

Set up internal diagnostics

1 From the Welcome to MicroStrategy Landing page, hover over **MicroStrategy Mobile Administration**, and click **Configure**.

2 Login with your MicroStrategy credentials you received in MicroStrategy cloud email.

3 On the MicroStrategy Mobile Server Administrator page, click **Configuration** under Diagnostics.

You can choose between two diagnostic setups: Internal and Custom.

4 Keep **Internal** selected. Custom is typically used to load a logger.properties file provided by MicroStrategy Technical Support.

5 As you launch the Intelligent Enterprise, you want to log all levels of information to keep track of utilization and errors. In the **Level** drop down, select **Messages.** This logs all errors, warnings, and messages.

6 Keep the **Maximum output file size** at 10,000,000 bytes. When the maximum file size is reached, a new log file is started.

7 Keep the **Number of file outputs** at 100. This is the maximum number of log files that are created.

8 Leave the **Flash profiler** drop-down set to Off. The Flash profiler helps troubleshoot Flash documents, which the BGH enterprise does not use.

Enable XML-API logs

For your team to better identify and debug API issues, the Mobile Architect can enable XML-API logs. The logs provide:

- The name of the class and method being accessed.
- Custom messages used in the code to specify the reason why the message was logged.
- All argument values sent to the method that logged the message.

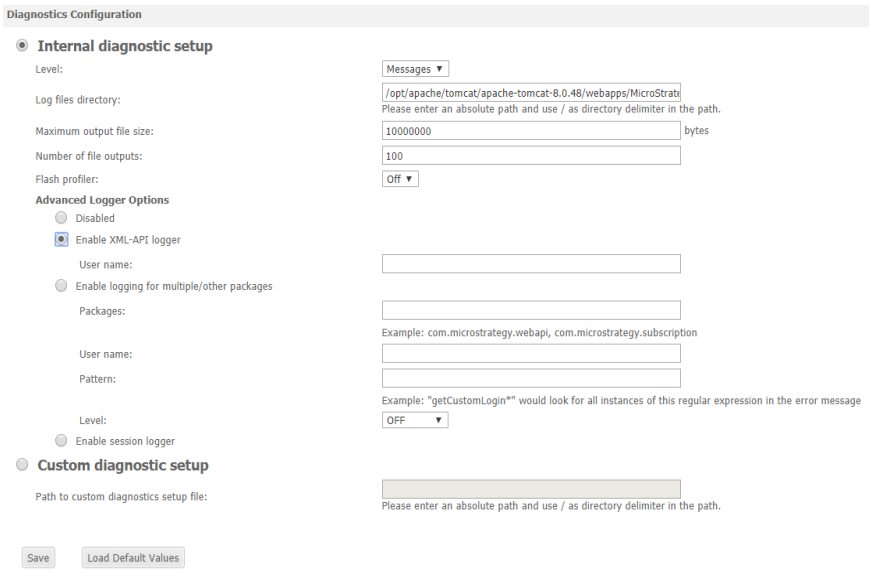
In the steps below, you use the Advanced Logger option to enable XML-API logging. Once enabled, your team can determine the specific user and package that is causing errors. This makes it easier to debug API issues as each log contains unique information about a specific user, date, and package.

1 Select **Enable XML-API logger** under Advanced Logger Options.

2 Leave **Username** blank. Only add users when you want to specify that the XML-API logger only saves messages for that user. If you do not specify a user, messages are logged for all users.

3 Click **Save**. The XML logs are now generated in the MicroStrategy Mobile deployment path \WEB-INF\log.

The diagnostics configuration should look similar to the image below:



As the XML logs use a significant amount of storage space, after retrieving the logs your team needs, you should disable XML-API logging.

4 Select **Disabled** under Advanced Logger Options.

5 Click **Save**.

Analyze system and server performance: Statistics

You can enable statistics to interpret and analyze system and server performance and disable them when you no longer need to monitor performance. For example, you can obtain information about the time taken by the Mobile Server and Intelligence Server to complete an operation, how much data is received and sent, the waiting time to receive some data from the Intelligence Server, and so on.

Best Practice

If the statistics are being saved in the file, the file size grows quickly, which is unnecessary. You should not log statistics unless the system is not working properly and you want to analyze the data for system tuning or troubleshooting.

Exercise 6.3: Configure Mobile statistics

You want to configure your Mobile Server's statistics to ensure you and your team have access to the appropriate information to monitor enterprise mobile applications.

Configure mobile statistics

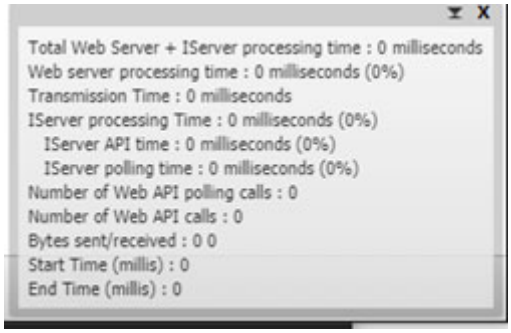
1 On the Mobile Server Administrator page, under **Diagnostics**, click **Statistics**.

2 For **Mode**, select **Screen**. Once saved, this option displays on all MicroStrategy Web pages.

The other options are:

- *File: Statistics are written to the file specified in the **source code control** box. Specify the absolute path to the file with / as the directory delimiter. For example, if the file is DemoStats and it is stored on the /C/ drive of the Mobile Server, enter C:/DemoStats.*
- *OFF: Statistics are not displayed on screen nor written to a file. By default, the mode is set to this option.*
- *Screen and file: Statistics are both displayed on screen and written to the file specified in the **Statistics file** box.*

3 Click **Save**.

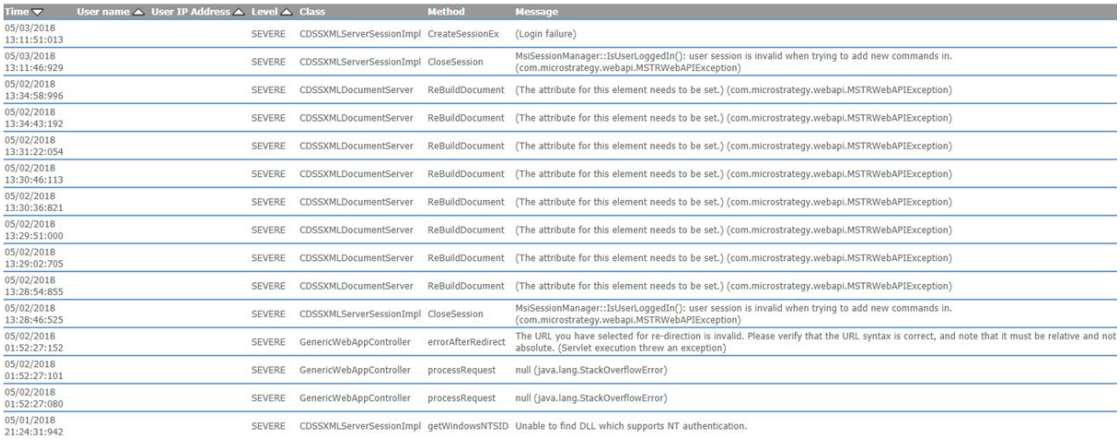


Notice the pop-up on your screen displays web page statistics.

4 To revert to the default settings, click **Load Default Values.**

Viewing server logs

The information collected in server logs can be difficult to understand if read directly from the log file itself. As an alternative to scanning a log that contains all information collected for your system activity, you can filter and view logged information using the View Logs page. This allows you to easily locate and troubleshoot application errors in the system.



Time	User name	User IP Address	Level	Class	Method	Message
05/03/2018 13:11:51:013			SEVERE	CDSSXMLServerSessionImpl	CreateSessionEx	(Login failure)
05/03/2018 13:11:46:929			SEVERE	CDSSXMLServerSessionImpl	CloseSession	MsiSessionManager::IsUserLoggedIn(): user session is invalid when trying to add new commands in. (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:34:58:996			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:34:43:192			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:31:22:054			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:30:46:113			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:30:36:821			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:51:000			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:29:02:705			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:28:54:855			SEVERE	CDSSXMLDocumentServer	ReBuildDocument	(The attribute for this element needs to be set.) (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 13:28:46:525			SEVERE	CDSSXMLServerSessionImpl	CloseSession	MsiSessionManager::IsUserLoggedIn(): user session is invalid when trying to add new commands in. (com.microstrategy.webapi.MSTRWebAPIException)
05/02/2018 01:52:27:152			SEVERE	GenericWebAppController	errorAfterRedirect	The URL you have selected for re-direction is invalid. Please verify that the URL syntax is correct, and note that it must be relative and not absolute. (Servlet execution threw an exception)
05/02/2018 01:52:27:101			SEVERE	GenericWebAppController	processRequest	null (java.lang.StackOverflowError)
05/02/2018 01:52:27:080			SEVERE	GenericWebAppController	processRequest	null (java.lang.StackOverflowError)
05/01/2018 21:24:31:942			SEVERE	CDSSXMLServerSessionImpl	getWindowsNTSID	Unable to find DLL which supports NT authentication.

Exercise 6.4: View the Mobile Server log file

View the Mobile Server log file

1. On the Mobile Administrator page, under Diagnostics, click **View logs.**
2. In the Display area, select the check boxes for the log information you want to display on the View logs page:
   - **Errors**: Logged errors are displayed on the screen. This is selected by default.
   - **Warnings**: Logged warnings are displayed on the screen.
   - **Messages**: Logged messages are displayed on the screen.

   For this exercise, select **Warnings**.
3. In the **From** area, specify the start date of logged information to display on the screen.
4. In the **To** area, specify the end date of logged information to display on the screen.

5 To display logged information, click **Refresh**. The log file information is displayed at the bottom of the page, containing:

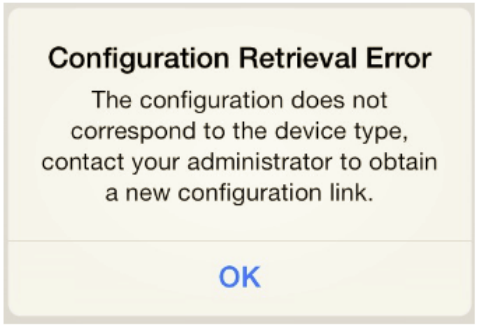
- **Time**: The time and date when the event log was created
- **User name**: The name of the user who logged in
- **User IP Address**: The IP address of the user who logged in
- **Level**: Error, Warning, or Message, as above
- **Class and Method**: Code references for the issue
- **Message**: The text of the logged message

6 To sort a column, click the **Sort** icon in the column's header.

Analyze issues specific to mobile apps: Device logs

The device log is useful for troubleshooting certain issues that occur within the MicroStrategy Mobile app for iOS or Android. When the logging level is set to All, the log captures diagnostics about document and report executions in addition to network communication.

For example, a construction supply company wanted to run MicroStrategy Mobile on iPad minis for their managers on construction sites. The mobile team created an iPad device mobile configuration and used Mobile Iron MDM to distribute the configuration link to their devices. However, when the configuration was applied to the iPad Mini, they received an error message that said the device type for the configuration did not match the actual device type.



To start solving the problem, the team tested the app on an iPhone with an iPhone Configuration URL. The iPhone app was working properly, so they were able to determine the issue was not with the configuration itself. Looking through the device logs, the team found the following message:

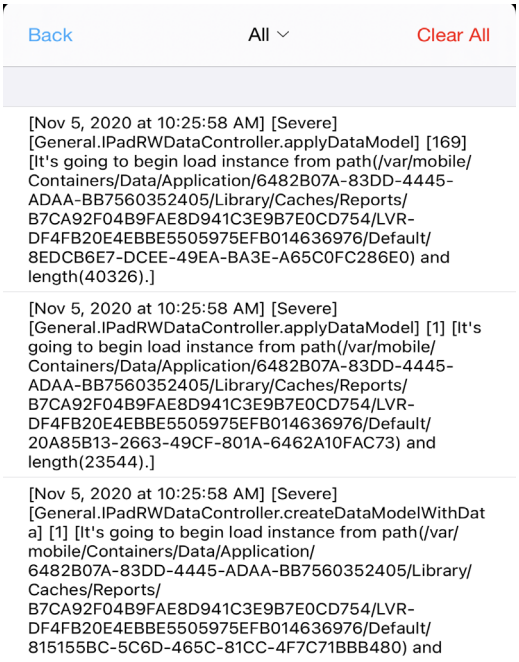
```
[Jul 13, 2018 at 10:39:45 AM] [Severe]
[Preferences.ApplicationLauncherController.startLaunch] [1]
[The device type of the configuration you are trying to apply
(2) is not compatible with the actual device type (1).]
```

Because the actual device type in the log file displayed 1 instead of 2 like it should have, the Mobile Architect reviewed the MDM settings. She found that the Mobile Iron Administrator installed the MicroStrategy Mobile for iPhone in the MDM, then pushed the iPhone version to the iPad Mini.

To fix the error, the Mobile Iron Administrator installed the MicroStrategy Mobile iPad client in the MDM, deployed the MicroStrategy Mobile iPad client using Mobile Iron to the iPad Mini client, then applied the correct iPad mobile configuration, allowing the app to launch on iPad Minis.

Generate device logs directly from the app

You can generate device logs directly from the app. Device log entries provide app developers and administrators with greater visibility and insight into performance, memory usage, notifications, network, preferences, EMM, restrictions, and other general usage statistics.



The image above is an example of the logs that can be generated from a mobile device. An app developer can review the error log to better understand why an end user did not receive a push notification or to analyze whether a device has been successfully registered with Google or APN services. Administrators can also

leverage user-generated device logs to tweak their environment to optimize performance.

Exercise 6.5: Analyze device logs

You want your team to view and analyze device logs for the BGH app, since some users have had several issues with it. Set up the mobile configuration so you can view the logs on your mobile device.

Change the logging level

1 In the Mobile Administrator page, click **Mobile Configuration**.

2 Click the **Modify** icon for your **BGH - *device*** mobile configuration.

3 On the Settings tab, set the **Logging level** to **All**.

4 Set the maximum log size to **9,000** to sufficiently capture the data in the file.

5 Make sure that **Allow users to access advanced and connectivity settings** is selected.

6 On the Home Screen tab, select **Display the default home screen**.

7 Click **Save**.

Generate a URL

1 Click the **Generate URL** icon.

2 Verify that the **port** is **443** and the **Request type** is **HTTPS**.

3 Click **Generate URL**.

Be sure to select the **Use short URL** check box for Android devices.

4 Copy and paste the URL in an email and send it to an email account accessible from your device.

5 Tap the URL in your email on your mobile device.

View device logs on your mobile device

1 Tap the **Settings** icon, then tap **Logging**. You may need to display your Navigation menu to see it.

2 Tap **View Log**. Here, you have the option to email the log.

3 Since this environment hasn't been heavily used, there won't be many errors. Take a look at sample errors your team might encounter, and how to fix them:

```
286:[THR:480][02/07/2003::12:24:25:342][DSS
ReportServer][Report Source Tracing]Not found in
cache: ReportInstance(Name="Length of Employment"
ExecFlags=0x1000180(OSrcCch UptOSrcCch) ExecActn=
0x1000180(RslvCB LclCch))
```

```
[May 1, 2018 at 5:14:33 PM] [Severe]
[Network.MSITaskRequest.handleResponse] [970]
[java.lang.NullPointerException(0)]
```

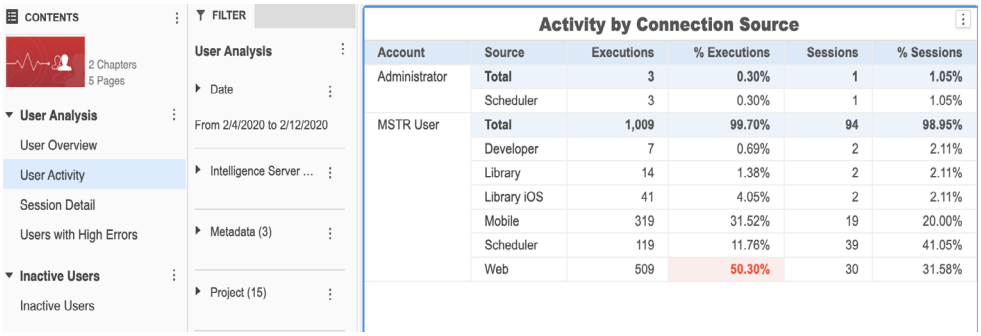
Monitor usage statistics in pre-created reports

MicroStrategy Platform Analytics provides reports and dossiers that display statistical data on MicroStrategy usage. As the Mobile Architect, you can use these pre-created reports and dossiers to make decisions about tuning the MicroStrategy mobile system or identify problem areas.

You can use the reports and dossiers as they are, copy them and then modify the copies, or build your own custom reports and dossiers to suit your needs. You can create new objects to perform the analysis you need. All the predefined objects are in the Public Objects folder in the Platform Analytics project.

For example, the User Activity dossier measures your MicroStrategy business intelligence system's use by mobile device users, and the overall contribution that Mobile usage contributes to the total business intelligence system use. This dossier gives insight into important KPIs. The Source attribute can be used to filter

the dossier to display only data for mobile. A portion of this dossier is displayed below.



Other pre-made dossiers in the Platform Analytics project include data on the following:

- **Intelligence Center**: This dossier provides insight on System, Application, and Dataset usage of your Intelligent Enterprise.
- **Object Telemetry:** Enables the administrator to review popular content based on executions, identify impactful objects with slow execution time and high error rates, and suggest unused content for clean-up.
- **Project Overview**: Provides guidance on the health of projects based on KPIs, comparison of execution load across Intelligence Servers.
- **Error Analysis**: Analyze errors based on session source, trends in error rates, and most frequently occurring errors.

Supporting the mobile experience: Troubleshooting

Given enterprise mobility's growing importance in day-to-day business operations, Intelligent Enterprises should focus on supporting mobile technologies and creating better user support experiences. As such, the Mobile Architect should create guidelines for troubleshooting Mobile Server issues. By creating troubleshooting guidelines recorded in a troubleshooting guide, the Mobile Architect provides a consistent methodology to zero in on root causes and search for resolution options.

Create troubleshooting guidelines

A common troubleshooting process is outlined below:

1. Gather detailed steps to reproduce the issue. Steps to reproduce should be specific step-by-step instructions to see the issue
2. Perform each step to confirm if the issue is encountered.
3. Record the steps that reproduce the issue 100% of the time or note if the issue is intermittent. Every time an option is clicked, a folder is navigated, or a task is performed, then a step should be noted.
4. Confirm the type of mobile application the issue appears on – out-of-the-box (OOTB), re-branded, or custom.
5. Use the steps recorded in adherence to these guidelines, and isolate the issue to the out-of-the-box MicroStrategy Mobile app if it appears on a re-branded or custom app.
6. Collect Mobile logs if further troubleshooting is necessary, and review for underlying errors when the issue is reproduced.
7. If the issue cannot be resolved by options or workarounds discovered by testing different behaviors in the various editors in MicroStrategy, explore possible back end errors.
8. Collect details on Mobile configuration if an issue appears to be configuration-specific.

Best Practice

Make sure your troubleshooting guide includes the following:

- Reproducing the issue with exact step-by-step instructions is critical to understanding the issue's behavior, which helps to formulate resolution options.
- Your team should request screen shots when applicable.
- Reference the *Proactive problem solving: Diagnostics and statistics* section to find which logs store what information, and how to set them.
- Work with your Platform Administrator to receive a copy of the DSSERRORS.log file to search for anything logged by the Intelligence Server that could be causing an app to crash.

7

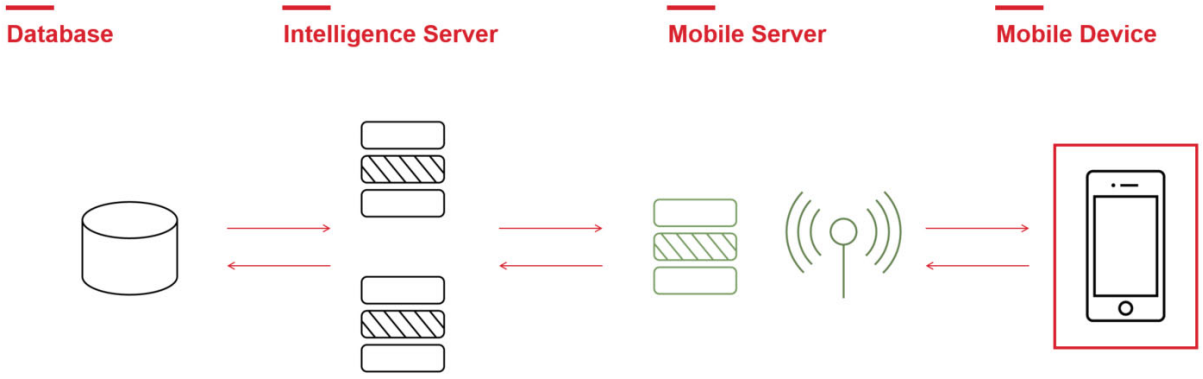
Optimizing the User Experience

An intelligent enterprise drives the adoption and success of enterprise BI applications. The mobility of that enterprise is essential as more employees are working off-site and traveling to meet with stakeholders. As such, the Mobile Architect is responsible for driving initiatives that advance and enhance the mobile user experience to truly enable intelligence everywhere.

Mobile performance improvement strategies

Performance is a product of multiple factors within the mobile workflow. To achieve optimal mobile performance, pay attention to each layer of the mobile workflow shown below. The Intelligence Server connects to your data sources, and the Mobile Server connects to your Intelligence Server. The Mobile Server acts as a medium for transferring data to the client over the wireless network. The

Mobile Architect should establish performance optimization standards that produce a best-in-class mobile workflow.



In this chapter we review various mobile performance improvement strategies that can implemented to optimize your enterprise mobile application, such as:

- Implementing mobile caching strategies, such as server caching, device caching, and offline functionality to optimize mobile performance.
- Optimizing data used for the mobile app, by leveraging prompts to limit the amount of data displayed on an app page or combining datasets to help decrease load time on a user's device.
- Encouraging user engagement by enabling mobile push notifications.

Performance influences: Mobile caching

One of the most important strategies for increasing mobile performance is caching. Creating and storing caches, on both the server and the mobile device, helps in optimizing the performance of the MicroStrategy Mobile application. As the Mobile Architect, you want to implement and enforce a comprehensive mobile caching strategy to help optimize performance by bypassing key bottlenecks on the server. While a lot of concepts concerning mobile caches are similar to caching within MicroStrategy in general, there are a few unique configurations specific to mobile, such as:

- Report and document caching
- Caching attribute elements
- Configure mobile devices to pre-load subscriptions
- Pre-caching content for offline use

Planning a mobile caching strategy

To enhance the user experience and plan for offline usage, evaluate and determine the enterprise's mobile caching strategy. Use the guidelines below to decide what methods are best for your organization's mobile applications.

Best Practice

The steps below outline what the Mobile Architect should consider and implement for a mobile caching strategy, including caching best practices.

1. Work with your Platform Administrator to enable caching at all levels, from database connections to element caches to reports.

2. Ensure that enough memory is allocated for caches. Insufficient memory reserved for caches causes reloading of cache files from disk. By allocating sufficient memory, you can eliminate or minimize repeated loading of caches.

3. Determine which reports or documents are used frequently in mobile apps and ensure that caching is enabled. Because of the degree of personalization and security for reports, the amount of RAM available on the Intelligence Server, and the available batch window time, caching all reports and documents is not feasible.

4. Ensure XML caching is enabled for reports and documents.

5. Specify that caching is disabled for highly prompted reports and documents, or prompts on attributes. Again, this saves space on the Intelligence Server. For highly prompted datasets, require the use of Intelligent Cubes with prompted OLAP reports as the document's data source to increase data retrieval speed. Also ensure attribute elements are cached, as this increases prompt execution speed.

6. Device caching is a must for optimal performance. Device caches can render directly from device memory with no need for interaction over the network or server and hence are the most optimal method of running apps on mobile devices.

7. Ensure mobile devices are configured to pre-load caches so apps can cache information and work offline.

8. Implement cache maintenance policies to clean and replace older versions. Every time a device logs on to the network, it checks against the server version of caches. If they have been updated, the device drops the cache it has currently and picks these fresh caches for future use to ensure data integrity.

9 Use subscription caching when personalized caches are necessary due to the dependency of prompts within the supporting datasets. The caches are delivered to the device when the user opens the app.

10 Configure adaptive caching on documents and supporting datasets so that the document executes and stores its cache on the mobile device until the cache is expired or invalidated.

After reviewing each of the caching strategies listed above, implement the appropriate methods for your organization.

Access data wherever, whenever: Use caching to optimize offline functionality

In today's mobile world, employees rely on being able to consistently view and analyze their data, no matter where they are. As the Mobile Architect, you should establish guidelines that ensure mobile apps can be accessed even when wi-fi or cellular networks are not an option by establishing offline functionality.

Best Practice

To optimize offline functionality of enterprise apps, the Mobile Architect should require the following:

- Selectors should be set to slice, instead of filter. This forces all potential selections within a filter to be included in the binary sent to a device ahead of time.
- All levels of pre-caching should be enabled to force the download of all required components before connection to the servers is interrupted.
- To allow MicroStrategy Mobile to download data when the application is running in the background, select the **Enable Background Download Mode** check box while defining the mobile configuration.
- The Enable Push Notification option must be selected before enabling background downloading.
- To ensure application up time while offline, your team should create mobile subscriptions. Scheduling a subscription for users ensures a physical copy of the document is sent to devices and is readily available at all times. This also helps engage users as they get an email notification that the document is ready to view.

Your team can continue to add Transaction Services to documents. Write-backs can be performed while in offline mode. The actual insert or update does not occur until connection is restored. At this point, the queue is emptied and the

inserts and updates are initiated. In the interim, write-backs are stored in-memory on the device.

Device caching: Render directly from device memory

With Mobile caching, documents can render directly from device memory with no need for interaction over the network or server. This improves mobile performance and allows for faster load times. When a user executes a document in Mobile, the app executes the following workflow:

If a device cache exist:

- The app renders the dossier from the device cache. Then, the app reconciles against the server to check if the device cache includes the latest dossier updates. If the cache is not up-to-date, the device downloads the latest cache from the server and allow the user to refresh the dossier.

If the device cache does not exist:

- The app searches the server for a cache, then uses that cache to render the dossier.
- If a server cache isn't present, the document or dossier executes against the data warehouse or Intelligent Cube.
- A new device cache is created.

Document and dossiers that have a device cache can also be run offline, allowing users to access their business intelligence information on the go.

Exercise 7.1: Set device caching for a document

Most documents that app designers use execute datasets, then store the results in a cache until the cache expires or is invalidated. As a best practice, you want your designers to use device caching for these types of documents.

In this exercise, you set device caching for a document used in a mobile app.

Access MicroStrategy Developer

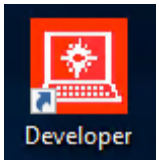
1 On the Welcome to MicroStrategy Landing page, scroll down and hover your cursor over **Remote Desktop Gateway**, then click **Launch**.

2 In the Remote Desktop Connection window, in the **Username** and **Password** boxes, type the user name and password listed in your Welcome to MicroStrategy email.

3 Click **Login**.

4 On the web page, under All Connections, click **Developer Instance RDP**.

5 Double-click the **Developer** icon. When prompted, log in with your username and password.



Configure device caching for the Category Management document

1 Expand the **MicroStrategy Tutorial** project, then double-click the **Public Objects** folder.

2 Double-click the **Reports** folder, then **Enterprise Reporting Documents**.

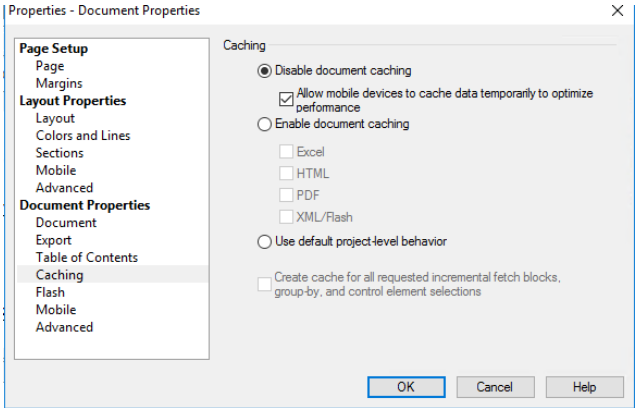
The Store Performance Management Dashboard used in BGH's mobile app has a large dataset and takes time to load. To help with loading time in the app, you want to set up device caching.

3 Right-click the **Store Performance Management Dashboard (For a specific Region)**, and select **Edit** to open the document in Edit mode.

4 From the **Format** menu, select **Document Properties**.

5 Select **Caching** from the list of Document Properties on the left.

6 Select **Disable document caching**, then select the **Allow mobile devices to cache data temporarily to optimize performance** check box.



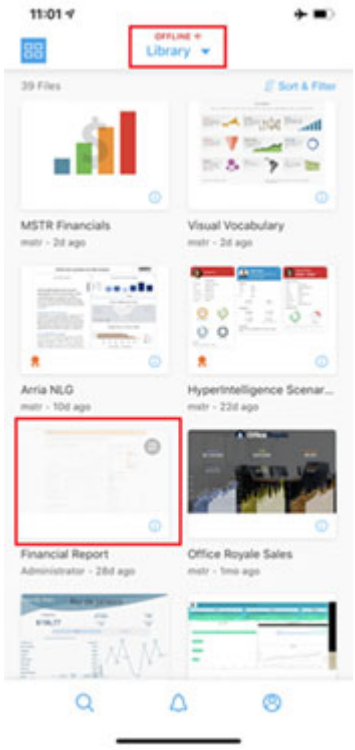
7 Click **OK**. Now, when the document runs on a mobile device, the device temporarily stores the document information.

8 **Save** and **Close** the document.

9 Keep **Display prompt and use current answers as the default answers** and click **OK**.

Accessing offline dossiers in MicroStrategy Library

You can access your dossiers in areas with limited or no network availability, helping you maximize your productivity while on the move. Dossiers that are unavailable in offline mode are grayed out.



Background loading of all dossier pages

For 2021, iOS Library Mobile implements background loading of all dossier pages when a dossier is executed. This provides a faster response time and a better user experience, because the dossier's chapters and pages are available before the user requests them.

Download multiple versions of a dossier

End users can download multiple versions of the same dossier, in their preferred format: PDF, Excel, or mstr. Multiple versions of the same dossier allow analysts to easily compare previous data against current data. When a dossier is downloaded, regardless of the format, it is saved in one common location. Combining downloads in a single, easily accessible folder provides a simpler viewing experience in iOS Library Mobile 2021.

Configure mobile devices to pre-load cache

Best Practice

Not all BGH documents or dossiers use prompts. For these documents and datasets, you should require your team to pre-load cache when the app is opened. Caches can be pre-loaded through mobile subscriptions or pre-caching. By default, if a cache exists for a subscribed report or document, that cache is loaded when the user opens that report or document. This speeds op the initial access to the application. However if you choose to pre-load caches, mobile users may experience a longer initial load time, but instant access is provided to individual documents with available caches. Test bot methods to determine which option is suitable for your environment and user base.

Exercise 7.2: Enable Subscription caching

The Store Performance Management Dashboard has a region prompt so that managers only view data for their region. Per your caching guidelines, when apps use prompts or security filters, it is necessary to use subscription caching. Allowing caches to be delivered to the device when the user opens the app.

In the exercise below, set the cache update subscription in Developer.

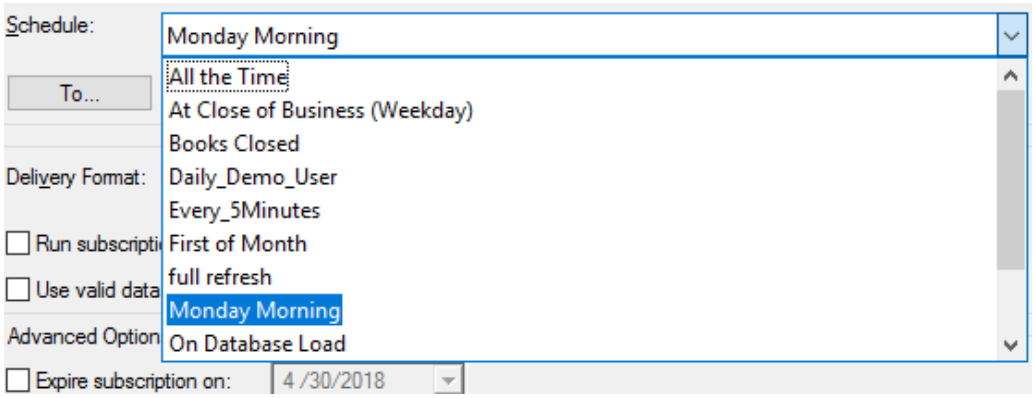
This can be done in MicroStrategy Web as well.

Set the Cache Update Subscription for the Store Performance document

1. In the Enterprise Reporting Documents folder, right-click the **Store Performance Management Dashboard (for a specific Region)**.
2. Point to **Schedule Delivery To,** and select **Update Cache**. The Personalization Editor opens.
3. Make sure all of the Regions are selected, and click **Next**.
4. Click **Finish**. The Subscribe to Cache Update dialog box opens.

   From the Schedule drop-down list, you can select a schedule to control how often the subscription occurs. You want the cache update to be sent at the beginning of the week.

5 From the **Schedule** drop-down list, select **Monday Morning**.



You can create new schedules in the MicroStrategy Developer Schedule Manager. For steps, see Scheduling Jobs and Administrative Tasks in the System Administration Guide.

6 Leave the **MSTR User** as the **To** recipient.

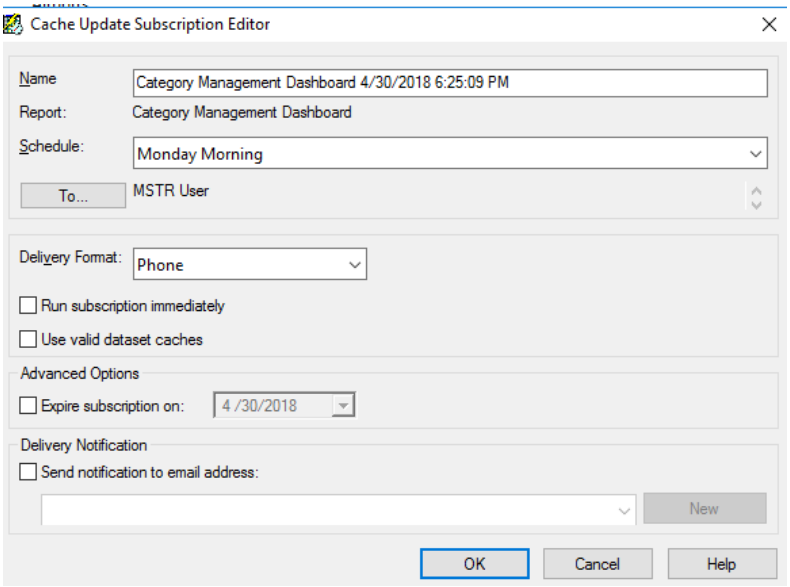
By default, you are the recipient for the subscription. To add additional recipients or to remove yourself from the subscription, click **To**. The Recipients Browser dialog box opens. Select the users and groups that you want to receive the subscription and click **OK**.

7 From the **Delivery Format** drop-down list, select **Phone**.

8 To execute the report or document immediately when the subscription is saved, select **Run subscription immediately** check box.

9 To use the valid caches for the datasets in the document, select the **Use valid dataset caches** check box.

10 To send a notification when the subscription executes and the cache is updated, you can select the **Send notification to email address** check box. For this exercise, leave the check box cleared.



11 Click **OK**.

Pre-caching: Download online content for offline use

As the Mobile Architect you can configure online content, such as videos or files, to cache on a mobile device. This allows a user to view the content when the device is offline. For example, when a user views a streaming video on a mobile device using the Video Player widget, the streaming video is buffered and played as it downloads.

By default, the downloaded video file is discarded when the MicroStrategy Mobile application is closed. If the video is downloaded instead of streamed, pre-caching downloads the video before the user requests it. This allows a faster response time, and you can also ensure that the video is downloaded and stored on the mobile device for offline use.

To accomplish this, you must provide a WebDAV folder that contains the video, and create the following:

- A document that contains the content to be pre-cached. For example, if you are pre-caching a video, you can create a document with a Video Player widget that displays either of the following on an iPhone or iPad:
  - A streaming video which provides the location of the video file using the Alternate Download URL setting.

- A downloaded video which provides the location of the video file using the Video URL setting.

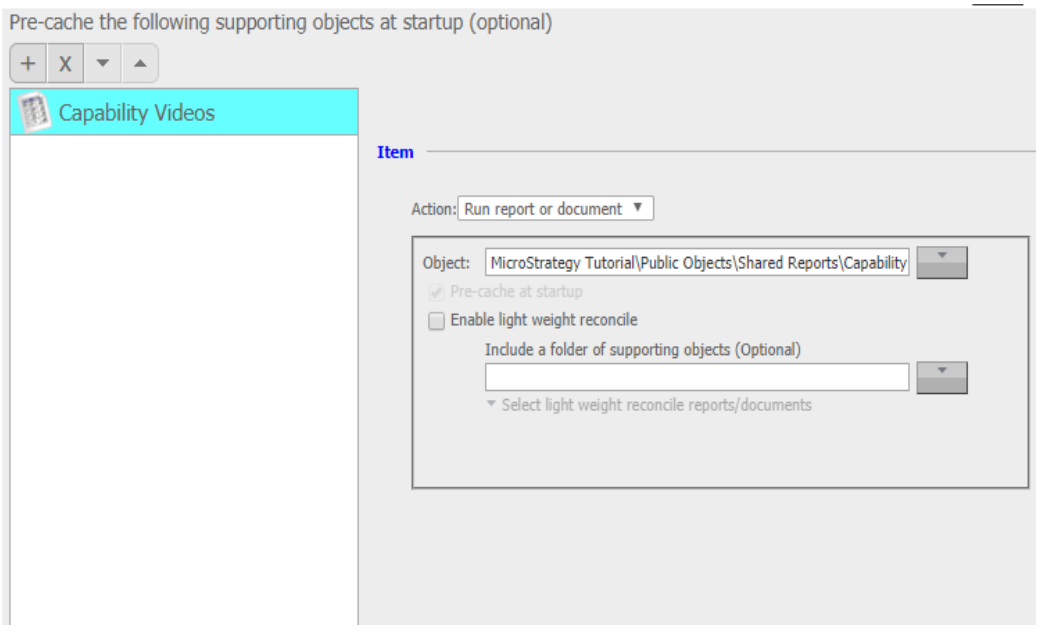
Exercise 7.3: Enable pre-caching for a document

In this exercise, you cache a document with a video. When a user loads the app's home page, the app downloads the video onto the mobile device.

Pre-cache the Capability Videos document

1. In Mobile Administrator, click **Mobile Configuration**.
2. Select **Modify** next to your **BGH** configuration previously created.
3. Select the **Home Screen** tab, then select **Display a custom home screen**.
4. Click the **plus** icon to add pre-caching to a supporting object.
5. Click the down arrow next to Object. Log in with your credentials.
6. Type **Capability Videos** in the search bar and press **Enter**.
7. Select the first document on the list.

   The document has been added to the pre-caching list, and the video loads once the app is launched.



8 Click **Cancel** to close the configuration.

Portable images: Embedding

To ensure that images are portable and available when the app is opened offline, the Mobile Architect should require app designers to embed the image into an HTML container object using the <img> HTML tag along with the base64 encoded binary-to-text values.

This can save time and improve the user experience when network connections are slow. By embedding an image, the file is embedded into the document itself, without having the image file exist as a file in a specific folder pathway, or within a public or private URL that hosts the image.

These steps below are for reference only, and not intended to be performed in class.

Embed an image in a document

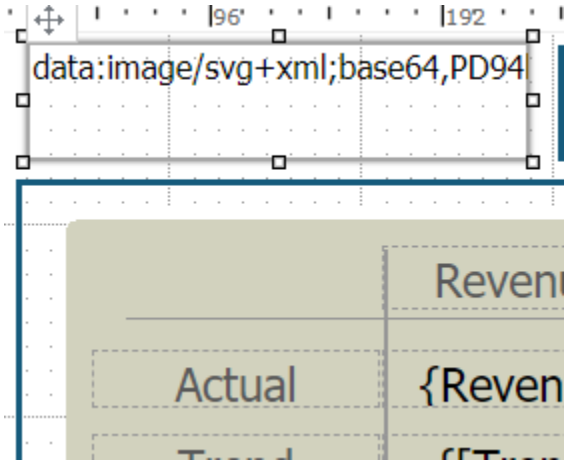
1 Acquire the HTML of a base64 encoded image, or take an existing image file and convert it to base64 via a third party website or tool.

Base64 is a generic term for a number of similar encoding schemes that encode binary data by treating it numerically and translating it into a base 64 representation.

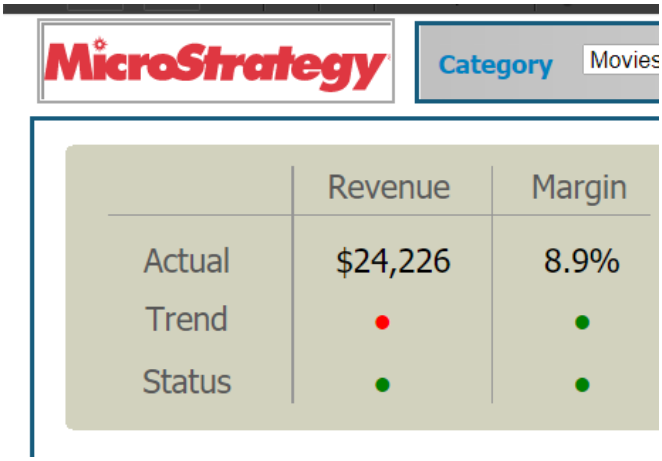
2 Open the document where you want to embed the image.

3 Select the **Insert** menu, then click **HTML Container**.

4 Copy and paste the **HTML** into an HTML container object.



5 Run the document in **Presentation Mode** to view the embedded image.



With iOS applications, you can also embed images directly to the mobile client by bundling images with your MicroStrategy Mobile application during compilation.

Continued app success: Optimizing mobile performance

The Mobile Architect should establish guidelines to optimize mobile application performance to ensure continued app success. Below are methods the Mobile Architect and app development team should leverage to increase efficiency.

- Use Intelligent Cubes: When possible, run documents and dossiers against an Intelligent Cube to increase data retrieval speed. If the cube is large, use

prompted reports that access the cube to reduce the columns and rows being sent to the mobile application document.

- Combine datasets and remove extraneous objects: Because all datasets require time to execute and join together, remove any datasets that are not used on your document or dossier.
- Use prompts and selectors to reduce document size: This helps reduce the app's footprint on a mobile device.
- Data rendering: Information Windows on iPad documents and dossiers can help improve speed. Information Windows can be configured to display additional data when a user clicks an element on a grid or graph, thereby only incrementally loading the data if the user requires it.

Faster app load time: Limiting and managing data

To ensure each page of the app performs well, the Mobile Architect should require that app designers are only using necessary data and the correct datasets. The less data transfered over the network and rendered on the device, the faster the app loads and renders.

For example, if a document is unable to be cached, the app author should create an Intelligent Cube of the document's data. Intelligent Cubes execute quicker than data from the warehouse because, rather than returning data from the warehouse for a single report, you can return sets of data from your data warehouse and save them directly to Intelligence Server's memory.

In addition to following the best practices outlined below, use Platform Analytics dossiers to monitor data usage and remove unused datasets from the app.

Best Practice

Below are best practices that the Mobile Architect can implement to manage app data:

- Limit on-screen data as necessary -- the user can always drill into more details later. Only display the data and elements that are absolutely necessary for the document or dashboard.
- Customize the workflow based on audience. As mentioned in the second chapter, creating a light-weight landing page with links to different documents helps to guide the users to only the documents they need.
- Intelligent Cube-based executions are faster than warehouse executions. The reports created from the shared sets of data are executed against an Intelligent Cube, rather than having to be executed against a data warehouse.

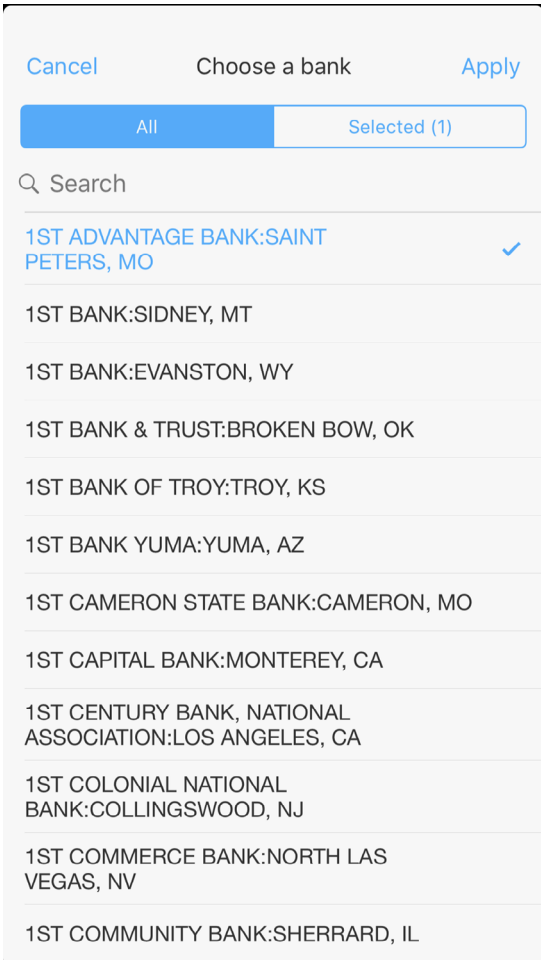
- Images add to the overall size and rendering complexity of the app. Use the appropriate format, resolution, size and number of images. Require that high resolution images are compressed. Images should provide no more resolution or size than users' devices can render
- Avoid using derived metrics and derived attributes when possible. Derived attributes and derived metrics can negatively impact data performance because they add processing time on the Intelligence Server to calculate the values.
- Reduce virtual dataset size by consolidating individual datasets and remove any datasets that are not used in the app.
- Where possible, replace custom groups and consolidations with attributes or look up tables. This helps speed the processing time for the dataset. Data processing should be pushed to the database (or data source), where the data executes faster than it would if executed at the Intelligence Server level in the Analytical Engine.

Reducing page size with prompts

Best Practice

The Mobile Architect should require that prompts be implemented when possible to reduce the amount of data that is returned. A prompt is a question the system presents to a user when a report is executed, such as asking the user to specify a geographical region or specifying a Revenue amount. Prompts help decrease the size of the document or dossier, as the data is filtered before the document is rendered. To control user input for a prompt, app designers should specify a

minimum or maximum value, or set limits on the number of possible answers. This allows designers to create a more predictable result set for each prompt.



In the example above, when the user taps a document in the app, the app displays a bank prompt. This narrows the data on the resulting document, which displays key metrics, to just the selected banks, helping improve app performance.

Display a subset of data with selectors

To display only a subset of the available data and allow users to interact with the result set after the document or dossier has been displayed, require app designers to use selectors. Implementing selectors can clear up screen space by minimizing the amount of data that is displayed at one time. Use a slicing selector or filtering selector based on the unique characteristics of each document.

- The slicing selector, which is the default type, loads the entire data set that is targeted by the selector into the mobile client. Once the document is

loaded, users have the ability to instantly change the selected element. However, depending on the amount of data available, users should expect a significant initial load time.

- The filtering selector loads only the data for the currently selected element. Users can expect a shorter initial load time, but the mobile client must connect to the Intelligence Server to download data each time that a new element is chosen in the selector.

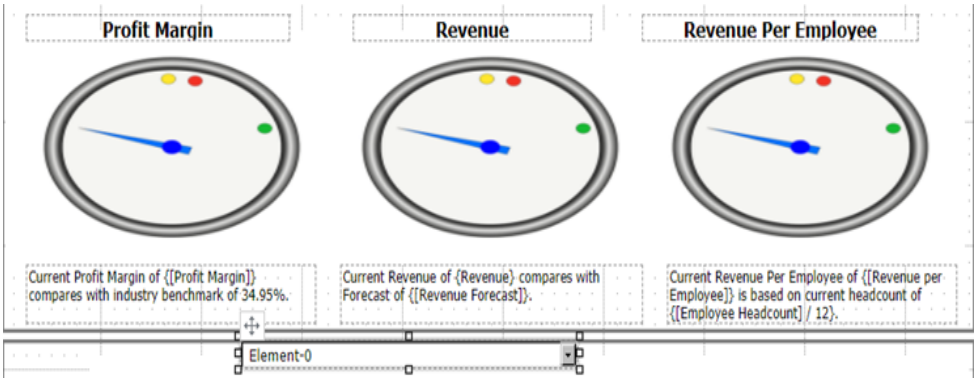
Exercise 7.4: Create a slicing selector

The Enterprise Performance Management Dashboard in the enterprise app is loading slowly and causing frustration for users. To help fix this issue, you've requested that the app designer add a selector that is set to slice. Additionally, this document is accessed in offline mode. If a selector is applied using slicing, all the slices, and therefore all the data, are included in the document. An offline user can change the selector and update the target.

In the exercise below, you create a slicing selector.

Add the selector to the document

1. In MicroStrategy Web, navigate to the **Shared Reports** folder in MicroStrategy Web. Then click the **Enterprise Reporting Documents** folder.
2. Right-click the **Enterprise Performance Management Dashboard** document and click **Edit.** The document opens in Edit mode.
3. Click **Insert**, point to **Selector**, and select **Drop-down** for the selector type.
4. Use the cross-hairs to draw the selector below the gauge widgets.



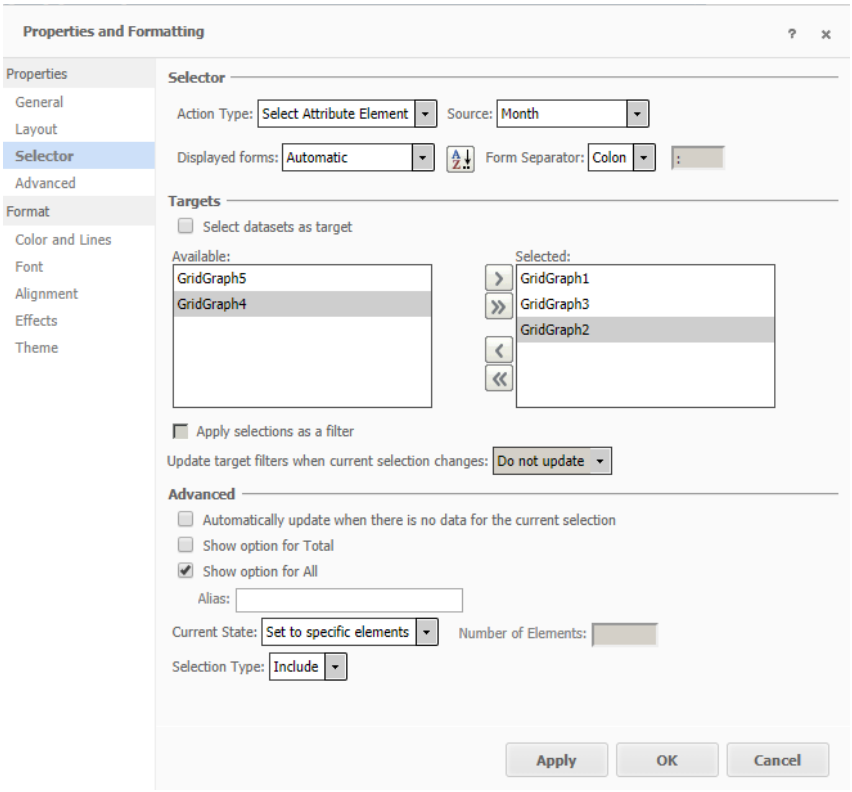
5 Click the **Month** attribute from the Dataset Objects panel and drag it to the selector. The selector now shows Month-0.

Set the target for the selector

1 Right-click the **Month-0** selector and select **Properties and Formatting**.

2 Double-click **GridGraph1**, **Gridgraph2**, and **GridGraph3** to target the gauge widgets.

3 Leave **Apply selections as a filter** cleared. Selecting this would change the selector to a filter instead of slice.



4 Click **OK** to apply the changes.

5 Click the **Presentation Mode** icon to view and interact with the selector.

Stay up-to-date: Monitor device releases

The Mobile Architect is responsible for staying up-to-date with the latest device releases. When a new device is released, you can assess how your apps work within the new device. This helps you and your team determine what changes need to be made to accommodate users with those new devices. For example, with the release of the iPhone X, you can incorporate FaceID to your app to provide a native design feature in your app.

Avoid EMM issues: Testing updates

In addition to device releases, the Mobile Architect should proactively check for updates to the enterprise EMM solution. If there are updates, MAM and MDM should be tested to identify and address any potential conflicts between the EMM update and the app. This avoids users experiencing issues due to an EMM upgrade that was not tested.

Continually optimize apps through performance testing

To continue ongoing optimization, the Mobile Architect should require that the mobile team continually tests the performance and stability of your app and the Intelligence Server to proactively solve performance issues your users might encounter. For example, if testing uncovers a document that takes 30 minutes to load, you can remove unused datasets to decrease load time.

To test the Intelligence Server capacity and load time, specialized third-party test tools, such as Borland SilkPerformer, HP LoadRunner, or Apache JMeter, can be used to automate the performance testing process of the Intelligence Server. Generally, these tools require recording a set of user actions from the Web server, which is used as a proxy for the Mobile Server for testing purposes. For example, opening a home page document, then navigating to a report.

These actions are then parameterized and automatically executed multiple times, simulating user actions and generating load on the Intelligence Server to understand how different scenarios affect performance. If testing finds the load time is exceptionally high, the Mobile Architect should work with the Platform Administrator to improve performance, for example increasing memory on the machine that hosts the Intelligence Server.

When using third party testing tools, the Mobile Architect should discern how the enterprise's users interact with MicroStrategy Mobile, to then build a script to simulate user interactions. For example, the Mobile Architect should understand:

- Do you plan to use Mobile during the system's peak usage time?
- How many users are expected to access the system via mobile devices?
- What actions will users take within the app? Will they simply open one page, or navigate to multiple pages? You should estimate how many processes users trigger.
- How many users are expected to submit mobile jobs concurrently?
- How many interactive jobs a minute are expected to be submitted via mobile devices during the system's peak usage time?
- How many of the jobs are going to be pre-cached?
- Provide the best estimate of the average number of datasets in each dashboard.

For testing app performance, you can manually run the app with the following scenarios to identify poor performance areas:

- Track the load time when nothing is cached.
- Track the load time when server caches exist.
- Track the load time when the device cache exists.

What other scenarios could you test to identify poor performance?

Engaging users for app success

By designing apps that follow the guidelines you created, your enterprise has access to powerful, intuitive, and convenient mobile apps. However, apps can only be successful if users are engaged. A successful Enterprise Mobility program within an Intelligent Enterprise continually improves the user experience by incorporating feedback from users within mobile applications.

Push notifications and subscriptions that deliver data to the app can also be leveraged to help keep users engaged and avoid usage attrition.

Learn from the users: Collect feedback

Mobile app creation is an iterative process. To enhance user experience and continually enhance apps. The Mobile Architect establishes a process to collect feedback from users. This helps app designers identify app enhancements that

can increase app utilization and engagement. Strategies you can require to collect feedback include:

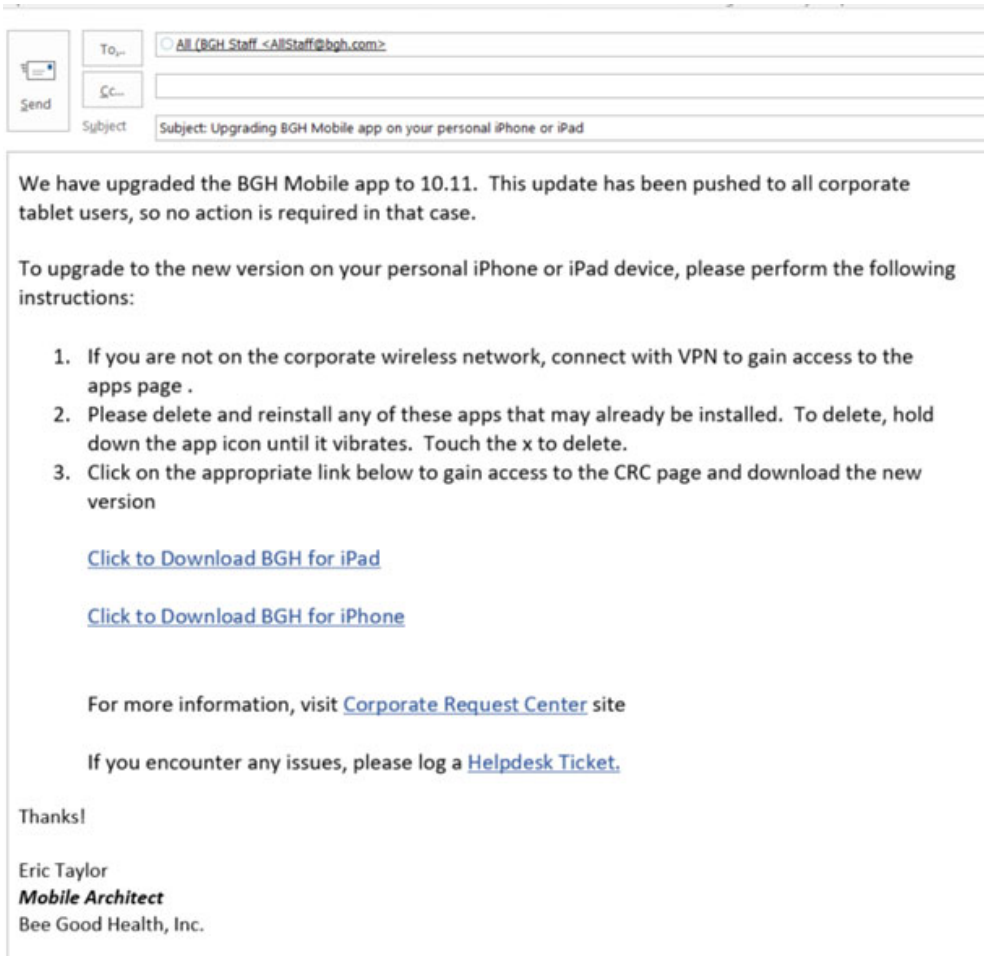
- Provide space for in-app feedback through Transaction Services. For dossiers, users should be encouraged to add comments to the dossiers.
- Gather one-on-one feedback via email and scheduled interviews.
- Send surveys on a periodic basis to the user base.
- Leverage reports to view which users have stopped accessing apps, to better understand why they no longer use the app and what can be done to reengage them.

What are some questions you would ask users to gather their feedback?

Engaging users through communication

As the Mobile Architect, you should send frequent communication to users to continually engage them. You can send app tips and tricks, or remind users to update their OS to the latest version to take advantage of new functionality. In the

example below, the Mobile Architect sent instructions to update the corporate app to the latest version.



Speak directly to users: Push notifications

Push notifications help increase user engagement for mobile apps and improve user experience by sending them notifications when a specific event occurs within the data. For example, with an alert-based subscription, users receive a push notification that a report has been executed and delivered to their mobile device.

Working with the System Administrator, the Mobile Architect ensures push notifications are set up between the Intelligence Server and iOS Apple Push Notification Service (APNS) and Google Cloud Messaging Services.

What notifications might users want to see from their apps?

Prerequisites for enabling push notifications

The prerequisites for enabling push notifications for iOS and Android devices are similar. The first requirement is to have a service that can send messages between servers and client applications. MicroStrategy delivers alert notifications for iOS devices through the Apple Push Notification Service (APNS). You must have an Apple iOS developer license to send mobile push notifications. MicroStrategy delivers alert notifications for Android devices through the Google Cloud Message (GCM). GCM is a free service, but for some customizations you may need an API key obtained from Google Play Console.

Secondly, your Intelligence Server must be able to connect to the APNS or GCM service either with a direct connection or through a proxy, and you must have MicroStrategy Distribution Services. Next, push notifications must be enabled on the Mobile Server, and the Mobile Server must be configured to use Secure Socket Layer (SSL) encryption. Lastly, push notifications must be enabled on the mobile device. The first time MicroStrategy Mobile is opened on the device, the user is prompted to allow the app to receive push notifications.

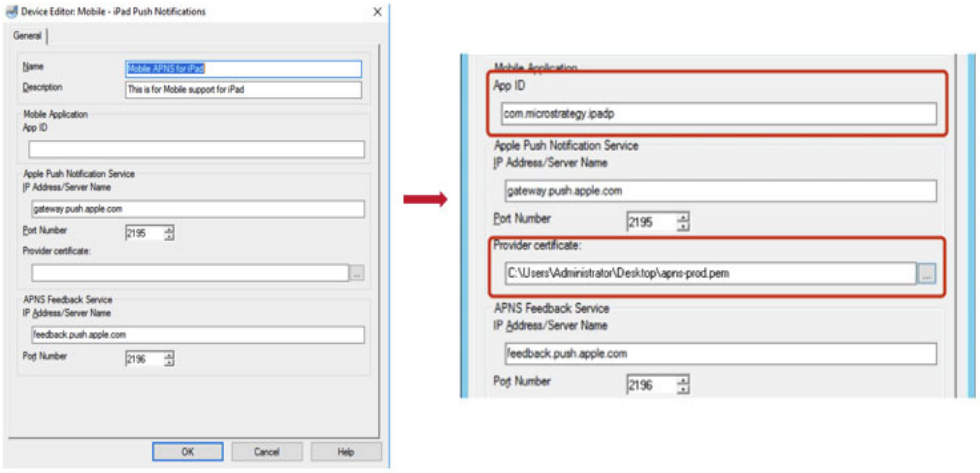
The following steps are for reference only, and not intended for an in-class exercise.

High-level steps for adding push notifications

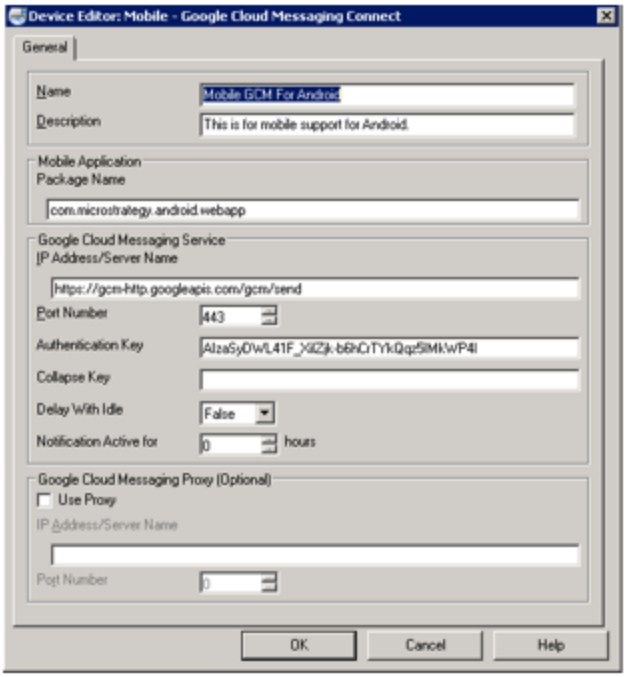
1 In Developer, create a **Distribution Services device** for iPhone, iPad, or Android. For detailed instructions, see Scheduling Jobs and Administrative

Tasks in the *System Administration Guide*, or see the *MicroStrategy Developer Help*.

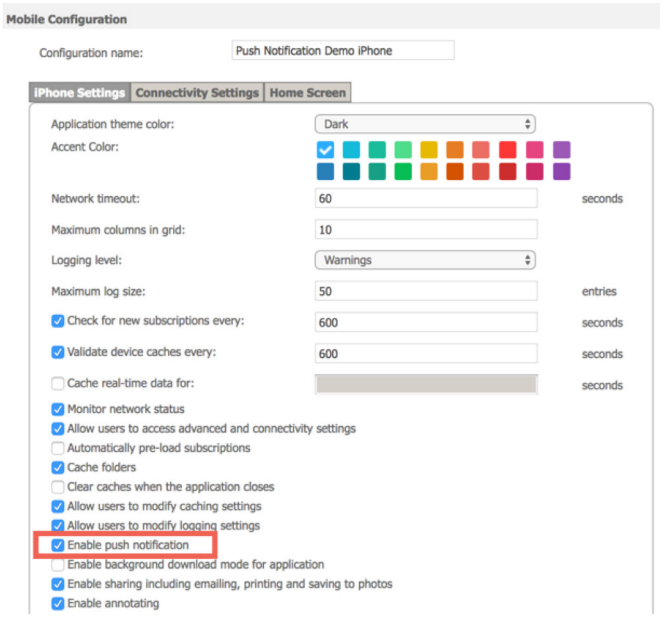
The image below shows the APNS certificate and the Distribution Service on the Intelligence Server for iOS devices.



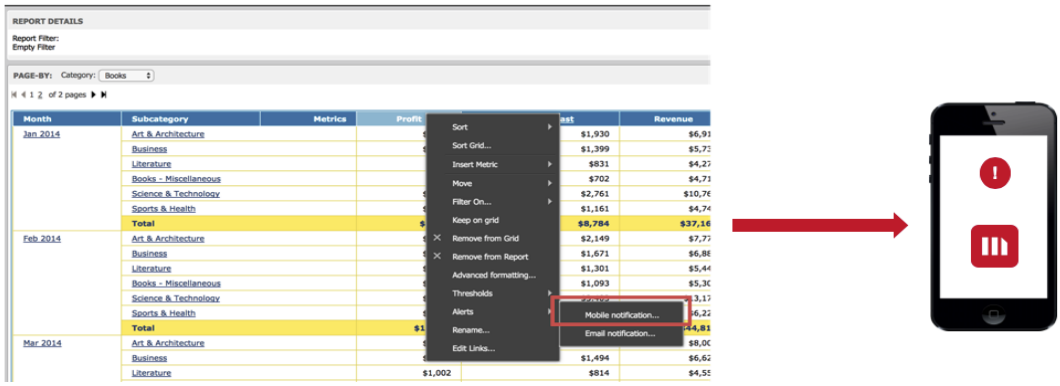
The image below shows the Distribution Service on the Intelligence Server for an Android device.



2 Create a mobile configuration for the application. In the **Settings** tab of the configuration, make sure that you select the **Enable Push Notification** check box.



3 Schedule a mobile notification using MicroStrategy Distribution Services.



4 The first time the user opens MicroStrategy Mobile on their device, they are prompted to allow push notifications for MicroStrategy Mobile. The user must select **Allow push notifications**.

Best Practice

Below are best practices and requirements for enabling push notifications:

- MicroStrategy delivers alert notifications through the Apple Push Notification Service (APNS). Therefore, you must have an Apple iOS developer license to send mobile push notifications. For information about the various iOS developer licenses available, see http://developer.apple.com/programs/.

- Your Intelligence Server machine must have a direct Internet connection to the Apple Push Notification Service (APNS).
- You must have MicroStrategy Distribution Services to use push notifications with MicroStrategy.
- Push notifications must be enabled on an iPhone or iPad for the device to receive MicroStrategy push notifications. The first time MicroStrategy Mobile for iPhone or iPad is opened on a device, the user is prompted to allow the application to receive push notifications.
- To enable push notifications, Mobile Server must be configured to use Secure Socket Layer (SSL) encryption. For detailed instructions on configuring Mobile Server to use SSL, see Enabling Secure Communication in the *System Administration Guide*.

Push reports and documents: Mobile subscriptions

To both engage users and improve app performance, you should require your app designers to create a mobile subscription to push the document to mobile devices at a specified date and time. This reminds users to access their app and interact with the data.

For example, if your company-wide Monthly Revenue report is run on a schedule and displays revenue in the Northeast region below $1,000,000, the Sales Report for the Northeast Region can be automatically emailed to users based on the $1,000,000 threshold that is part of the company-wide Monthly Revenue report. This is a subscription based on an event, as opposed to a time-triggered event.

The high-level steps to create subscriptions are below

1. A subscription to the report is created. Any prompts are answered, and the scheduled event or time is set. Subscriptions can be created by an administrator or mobile user in Developer or Command Manager.
2. The report or document is executed when the scheduled event or time is triggered. Upon execution of the report, the new report cache is sent to the MicroStrategy History List. An email is sent to the user alerting them that the document has been updated.
3. The user retrieves the report from the Mobile Server, and it is available on the user's mobile device.

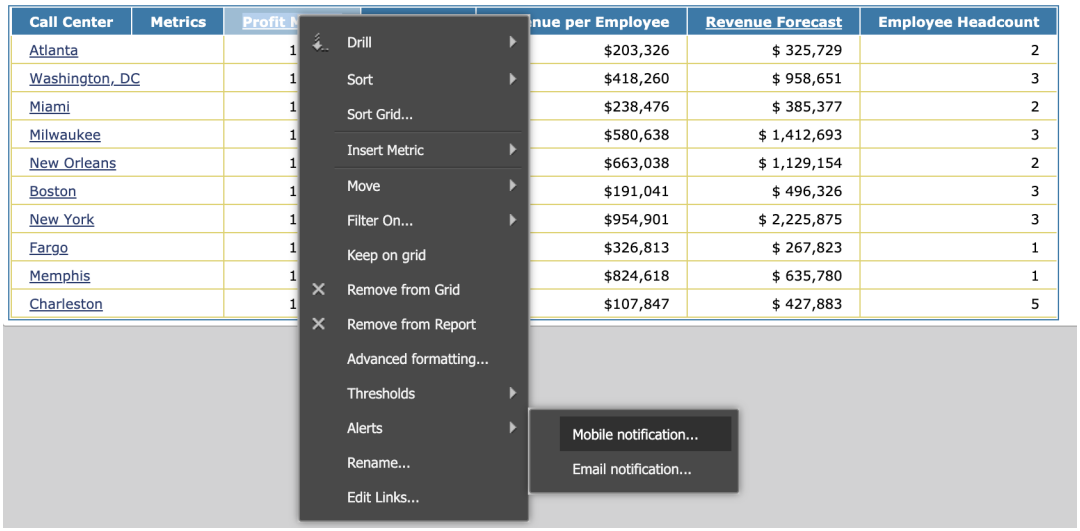
Exercise 7.5: Configure an alert-based Mobile subscription

The last requirement that needs to be met for the BGH mobile app to be complete is configuring an alert-based Mobile subscription. One of the store manager's request was to receive a mobile notification when the profit margin metric drops below a pre-set threshold.

In this exercise, you create an alert-based subscription to the Store Gauges report that is used as one of the dataset in the Store Performance Management Dashboard. Users receive a mobile alert if the Profit Margin metric drops below 10%. In addition, recipients should get a push notification and should be able to change their delivery setting, but cannot unsubscribe.

Configure an alert for the report

1. Log in to MicroStrategy Web, and navigate to the **Shared Reports** folder.
2. From Shared Reports, navigate to **MicroStrategy Platform Capabilities/ MicroStrategy Report Services/Datasets/Performance Management Dashboards** and open the **Store Gauges (Prompted)** report.
3. Click the **Grid** icon on the toolbar.
4. When the report grid displays, right-click the **Profit Margin** metric header, point to **Alerts**, and select **Mobile notification**.



5 In the Alerts Editor, in the Filter On drop-down list, select **Profit Margin**.

6 In the operator drop-down list, select **Less than**.

7 In the value text box, type **0.1**, and click **Apply**.

8 On the Alerts Editor toolbar, click **Cell Formatting**.

9 In the Name text box, type **Problem Region**.

10 On the Font tab, in the Color drop-down list, select **Red** and click **OK**.

Subscribe recipients to the alert report

1 In the Alerts Editor, expand **Delivery Settings**.

2 Under Alert Subscription, in the Schedule drop-down list, select **Daily**.

3 Click **To** to subscribe multiple recipients to this report.

4 In the Recipients Browser, search for and select the **Mobile Users** group.

5 Click **OK**.

6 In the Device type drop-down list, select **Phone**.

7 In the Target Application drop-down list, select **Mobile APNS For iPhone**.

A separate subscription needs to be created for each device type where the alert should be delivered.

8 Expand **Advanced Options**.

9 Select the **Allow recipients to change delivery setting** check box, and click **OK**.

A

Appendix: Mobile Architect Checklist

Mobile Architect description

Responsible for the Enterprise Mobile Applications, including Enterprise Mobile Management (EMM), Mobile Server, Mobile Application Standards & Guidelines, Mobile Style Guide, Mobile Caching Protocols, Transaction Services, Mobile Authentication, and Mobile Curation. Enterprise Mobile Applications are Enterprise Applications deployed on mobile devices, as well as iOS & Android apps integrated with the Enterprise Security, Data, and Application model.

Check list overview

Assess

1. % Uptime
2. No. of User Sessions
3. No. of Active Users
4. Avg. Session Time
5. No. of Apps Deployed
6. %Error Rate
7. Avg. Load Time
8. Avg. Run Time
9. No. of Executions
10. Experience Score
11. No. of Phones
12. No. of Tablets

Plan

1. Mobile Server Life Cycle Management
2. Mobile Applications Life Cycle Management
3. Mobile Deployment
4. Mobile Cache Protocols
5. Mobile Authentication & Security
6. Mobile

Create

1. Mobile Applications
2. Mobile Platform Configuration
3. Mobile

Publish

1. Mobile Applications
2. Mobile

Operate

1. Monitor Mobile Applications
2. Monitor Mobile Server
3. Troubleshoot Mobile Applications
4. Troubleshoot Mobile Server
5. Coordinate with the Intelligence Center
6. Provide Mobile Administration Support

Optimize

1. Mobile Application Performance
2. Mobile Server Performance
3. Mobile Reliability
4. Mobile Usability
5. Mobile Caching

Assets and tooling

Assess

- AWS Device Farm used to test Mobile Applications on multiple devices and operating systems

Plan

- Word Processor used to create documentation of Mobile Applications
- Wireframe Tool Balsamiq used to produce a mock prototype of Mobile Application
- Drawing Tool Draw.io used to create Storyboard in Mobile Application design workshops
- Xcode IDE used to build iOS Mobile Application archive file to be deployed
- Android Studio IDE used to build Android Mobile Application package file to be deployed

Create

- MicroStrategy Developer used to build objects used in Mobile Applications
- MicroStrategy Web or Workstation used to build objects used in Mobile Applications
- Xcode IDE used to build iOS Mobile Application archive file to be deployed
- Android Studio IDE used to build Android Mobile Application package file to be deployed
- MicroStrategy Mobile Administration page used to configure Mobile server and create Mobile Configurations

Publish

- Apple App Store used to deploy iOS Mobile Applications to users outside the enterprise
- Google Play used to deploy Android Mobile Applications to users outside the enterprise

- AWS Device Farm used to test Mobile Applications on multiple devices and operating systems
- MicroStrategy Workstation used to build Dossiers
- EMM Provider (App Config) used to deploy Mobile Applications within the Enterprise
- Test Flight used to deploy iOS Mobile Applications to test users inside and outside the Enterprise
- JMeter used to perform load testing of application servers

Operate

- Enterprise Manager displays statistical usage of objects within the MicroStrategy Platform
- Text Editor used to view log files
- Source Code Repository GIT used to archive the Mobile Application client software
- MicroStrategy Developer
- MicroStrategy Mobile Administration page
- AWS Device Farm used to test Mobile Applications on multiple devices and operating systems

Optimize

- MicroStrategy Developer used to build objects used in Mobile Applications
- MicroStrategy Web used to build objects used in Mobile Applications
- Speedtest.net used to measure network performance
- JMeter used to perform load testing of application servers

Detailed check list

Assess

Users

Key Performance Indicators:

- No. of Active Users
- No. of Phones
- No. of Tablets

Troubleshooting

- If the number of Active Users is significantly less than the previous day's value, initiate mobile application testing via AWS Device Farm or manually run applications to verify applications and network connectivity to find root cause of why users aren't logging in

Usage

Key Performance Indicators:

- % Error Rate
- No. of User Sessions
- Avg. Session Time
- No. of Executions

Troubleshooting

- If the % Error Rate is greater than 0, the Mobile Architect needs to determine the applications where errors were encountered and perform tasks listed in the Troubleshoot Mobile Applications document
- If the No. of User Sessions is significantly higher than the values over the past couple of weeks, then work with the System Administrator to analyze the

capacity of the Mobile server and Intelligence Server to ensure it can handle the increased load

- If No. of Executions has suddenly decreased, mobile applications should be tested via the AWS Device Farm or manually to ensure availability and functionality and, if issues are found, the steps listed in the Troubleshoot Mobile Applications document should be followed
- If No. of Executions or No. of User Sessions has significantly decreased and the EMM Administrator recently pushed mobile application updates, the EMM should be checked to see if the update was successfully deployed to all users

Applications

Key Performance Indicators:

- % Uptime
- No. of Apps Deployed
- Avg. Load Time
- Avg. Run Time
- Experience Score

Troubleshooting

- If % Uptime is less than 99%, then work with the System Administrator to identify issues that could have led to the outage
- If the Avg. Load Time exceeds 5 seconds, then the steps in the Mobile Application Performance section within Optimize should be followed

Plan

Mobile Server Life Cycle Management

- Work with the System Administrator to implement Mobile server hardware specs with regards to Memory, CPU's, and Disk Space for initial setup and future needs
- Work with System Administrator to deploy the Application servers that host the Mobile server software application

- Work with the System Administrator to configure Push Notifications between the Intelligence Server and iOS APNS and Google Cloud Messaging services for Android
- Work with the Platform Administrator to implement the MicroStrategy Mobile server software and upgrades
- Work with the Platform Administrator to deploy visualization plugins available from MicroStrategy on the Mobile server
- Refer to the MicroStrategy Community to ensure that all MicroStrategy Visualizations meet minimum requirements and prerequisites for all visualization plugins deployed on the Mobile server
- Work with the Services Architect to ensure all Best Practices are met for any custom plugins that are deployed on the Mobile server

Mobile Application Life Cycle Management

- Gather requirements for Mobile Applications to identify the content, device form factors, device operating systems, source data, users, workflow, transactions, data security, filter criteria, and user roles to design the Mobile Application
- Work with UI/UX team to build Mobile Application storyboards using a Drawing Tool to define the layout, navigation, and data presentation based on Mobile Application requirements
- Work with UI/UX team to build wireframes of the application based on the Storyboard using a Visual Prototyping Tool to present a mock prototype of the Mobile Application to the business sponsor for review before application development begins
- Work with the UI/UX team who will be responsible for creating image assets and design specs for the Mobile Application branding according to the Corporate Style Guide
- Document the Objects in the Mobile Application using a Word Processor for future developers to review before modifying the Mobile Application
- Use a lightweight Landing Page which links to other Mobile Documents for top application load time performance so linked pre-caches and subscription caches can be delivered ahead of execution by the user
- Download and update the latest MicroStrategy Mobile iOS SDK and Android SDK and apply necessary customizations to keep Mobile Applications on the latest Mobile Client version since MicroStrategy releases Quarterly updates which add new features and bug fixes

Mobile Deployment

- Deploy internal Enterprise Applications using an Enterprise Mobile Management (EMM) System to push Mobile Applications to mobile devices in the Enterprise
- Use vendor supported EMM SDK Integration when you have users who run the Mobile Application on their personal device and you do not need to have the ability to remote-wipe their device through the Mobile Device Manager (MDM)
- Use App Config EMM Deployment for Applications that support both iOS and Android and need to be on the quarterly-released MicroStrategy Mobile Client version
- Selectively distribute the iOS Mobile Application using Test Flight to users who will test the Mobile Application ahead of the official launch
- Distribute Android .apk files directly via email or provide an Over the Air link for users to install and test the Mobile Application ahead of the official launch as part of Beta Testing
- For iOS App Store Submissions, build and maintain a checklist of items to collect from the Marketing team prior to submitting the iOS App Store
- For Android App Store Submissions build and maintain a checklist of items to collect from the Marketing team prior to submitting to the Google Play Store

Mobile Cache Protocols

- Use Device Caching in Documents when the document must execute datasets and then store the results in a cache on the Mobile Device until the cache is expired or invalidated
- Use Subscription Caching when personalized caches are necessary due to the dependency of prompts or security filters within supporting datasets so that caches are delivered to the device when the user opens the Mobile Application
- Use Pre-caching when the document and supporting datasets are not personalized and caches can be updated via update cache subscriptions which are delivered to the device when the user opens the Mobile Application
- Work with the Platform Administrator to manage and administer Mobile server caches
    - Expiration settings should be set appropriately, to avoid unintentional expiration and discarding of server-side caches

- Dirty or invalid caches need to be cleaned and replaced, to make sure devices can always leverage server-side caches at all times, as needed
- Make sure there is enough memory for storing caches on the Intelligence Server

Mobile Authentication and Security

- Work with the System Administrator to establish remote device access via VPN including On-Demand, App Tunneling, Signed Certificates, Reverse Proxy, HTTPS / SSL, DMZ
- Work with the System Administrator and Services Architect to configure Authentication for the Mobile server and Mobile Application SSO, custom authentication, or Trusted Authentication
- Work with the System Administrator to configure authorization via LDAP or Active Directory
- Work with the System Administrator and Platform Administrator to place Mobile server in a DMZ to provide connectivity from outside the corporate network
- Work with the Platform Administrator to set up the Certificate Server using MSTR GUI-free utility for issuing Mobile Client side x-509 certificates
- Work with the Platform Administrator to configure authentication from the Mobile Device through Mobile server and MSTR Metadata
- Authentication Types
  - LDAP, Kerberos, ADS, Trusted Authentication, Multi-Factor Authentication, Usher
  - Trusted Authentication OOTB support for
    - Tivoli, Siteminder, Oblix, SAML, Ping Federate
- Work with the Services Architect for Custom Login Pages and HTML Login Forms Consumption
- Work with the Services Architect for Custom Authorization workflows
- Work with the Platform Administrator to setup user roles and permissions related to Application functionality and privileges
- Work with the Analytics Architect to create Data access security via security filters

- Work with the Platform Administrator to set up the Object access security via ACL
- Define Mobile Security settings using the MicroStrategy Mobile Administration page for Mobile Configuration
    - Confidential Projects, Device Auto-Lock, Failed Attempts Limit, Remote Wipe Device, Password expiration, Never Persist Credentials
- Work with the Services Architect to create Mobile server-Side Java task frameworks for custom mobile functionality such as Mobile Log-on tasks

Mobile

- Work with the Platform Administrator to capture the Configuration Link and send to Mobile Users via Email
- Work with the Platform Administrator to govern Dossier caches using MicroStrategy Workstation to set cache limits
- Mobile can be deployed by an EMM using the App Config method

Create

Mobile Applications

- Work with the Analytics Architect to create or modify Schema Objects to use in the Mobile Application
- Create Metrics, Prompts, Filters, Datasets, Dossiers, and Documents using MicroStrategy Developer or MicroStrategy Web to use in the Mobile Application
- Pass pertinent filtering and selection criteria between Documents using the URL API to meet the Navigation and Workflow requirements of the Mobile Application
- Work with the System Administrator to integrate an Identity Management System (SAML, O-Auth) in the Mobile Application to authenticate users access to the Mobile Application
- Work with the System Administrator to provide connectivity between the Mobile Clients and Mobile servers for users inside or outside of the corporate network

- iOS Deployment pre-requisites
  - Enroll in the iOS Enterprise Developer Program
  - Install Xcode IDE on a Mac Computer
  - Download latest MicroStrategy Mobile iOS SDK
- Android Deployment pre-requisites
  - Install Android Studio IDE
  - Download latest MicroStrategy Mobile Android SDK
- Compile iOS and/or Android Mobile Applications for distribution using Xcode or Android Studio
- Work with the Services Architect to integrate all custom client-side modifications
- Create Subscriptions and enable Document Caches using MicroStrategy Developer for Mobile Application performance and Off-line functionality
- Create mobile Alerts using MicroStrategy Web to send Push Notifications to Mobile Applications when defined Metric exceeds thresholds

Mobile Platform Configuration

- Use the MicroStrategy Mobile Administration page on the Mobile server:
  - Configure Mobile server and Intelligence Server connectivity, Diagnostics, Statistics, and Security to facilitate the communication between Mobile Devices and the MicroStrategy Platform
  - Create Mobile Configurations for Device, Connectivity and Home Screen settings to be applied in the Mobile Application
- Work with the Services Architect to create or customize Tasks on the Mobile server

Mobile

- Create Dossiers for mobile devices using MicroStrategy Workstation or MicroStrategy Web

Publish

Mobile Applications

- Run Mobile Applications to perform Quality Assurance tests:
  - Validate data in Mobile Applications to ensure Data Integrity
  - Perform functional tests to ensure all Buttons, Links, Information Windows, Visualizations, Grids, Selectors, Filters, Images, and Text all work according to Mobile Application design
  - Test performance of Datasets to ensure compliance with the SLA
  - Test overall performance of the Mobile Application on first load, document links, prompt, filter, and selector change to ensure compliance with the SLA
  - Confirm visualization plugins deployed on the Mobile server are functioning properly within the Mobile Application
  - Collect logs from the Mobile Client to create scripts to use with JMeter to simulate concurrent user session loads to test your Mobile Application performance when the Mobile server and Intelligence Server are under a severe load
  - AWS Device Farm is a testing service used to test your application with several different devices and operating system versions to ensure functionality works across the various device and operating system combinations your Mobile Application needs to support
- For Mobile Application deployments outside of the organization, use the iOS App Store or Google Play Store
  - Collect the App Store Submission Materials from the Marketing team
  - Overview of Publishing on Apple App Store
  - Overview of Publishing on Google Play Store
- Website Deployment of IPA and APK files
- Mobile Applications for employees of the organization can be deployed using an Enterprise Mobile Management system
  - SDK Integration
    - Only Airwatch supported for both iOS and Android

    - Mobile Iron and Blackberry (Good Dynamics) have iOS SDK versions of MicroStrategy Mobile
    - Every time the EMM provider releases updates to their EMM Solution, MicroStrategy produces a new SDK version of the MicroStrategy Mobile SDK
- App Config deployment methodology is the recommended approach to deploying Enterprise Mobile Applications
- Create an Email Template and Email Distribution list for EMM App Pushes to notify users when updates are occurring
- To limit the potential impacts EMM App Pushes can have on users devices during work hours, publish your updates in the evening

Mobile

- Work with the Application Architect to push Dossiers using MicroStrategy Workstation to users who will access them through the Mobile
- Share Dossiers with users using MicroStrategy Web

Operate

Monitor Mobile Application

- Track the application load time as well as the run times for other screens within the app using performance logs to ensure the Mobile Application is performing within SLA
- Run automated or manual tests to ensure Mobile Applications works as designed for each device type
- Report on KPI's listed in the Assess section above using Mobile Statistics from Enterprise Manager to analyze whether the app is adding value with user engagement increases over time, or the inverse, user engagement decreases over time
- Conduct User Interviews to gather feedback to incorporate into future releases of the Mobile Application to improve the usability and add new or missing features to maintain or increase user engagement with the Mobile Application
- Perform Regression Testing when a change is made to a supporting component like the MicroStrategy Mobile SDK, Schema Objects, Cubes, and

Datasets using AWS Device Farm to identify and proactively address any potential issues before the update is published to users

- Establish a process for archiving the Mobile Client using a source code control system to ensure the customizations are in a system that gets backed up so the next time the MicroStrategy Mobile SDK is updated, a compare can be done between the stock MicroStrategy Mobile SDK files and the customized files to identify the changes that need to be made when upgrading the Mobile Application

Monitor Mobile servers

- Review and Analyze the Mobile server Health Report to identify any potential resource issues the Mobile server might be encountering and work with the System Administrator and Platform Administrator to perform capacity planning
- Work with the System Administrator to Monitor User Sessions (Windows TCP/IP Socket Connections number of network threads) (Linux File Descriptors Initial Pool Size and Maximum Pool Size 2048 file descriptors per Linux) on Mobile server and Intelligence Server
- Limit the amount of time users can be considered active without performing any actions by setting the User Idle time parameter for the Intelligence Server

Troubleshoot Mobile Applications

- Contact the user who is reporting the bug with the Mobile Application and get a detailed description of the steps they performed before they encountered the bug in the Mobile Application so you can recreate the bug on your device
- Create a new Mobile Configuration for your Mobile Application using the Mobile Server Administration page to set the logging level to All, and specify 9,999 log lines to ensure the log file captures everything occurring in the client to assist with debugging the Mobile Application
- Email the Mobile Configuration link and apply it to the device you are testing the Mobile Application on
- Run the Mobile Application using the steps to recreate the bug and email the log to yourself for analysis
- Work with the Platform Administrator to receive a copy of the DSSERRORS.log file to analyze in a Text Editor to see if you can find anything logged by the Intelligence Server which could potentially be causing your application to crash

- Access iOS crash logs for the Mobile Application and email them to MicroStrategy Support for analysis along with the relevant steps to re-create the crash

Troubleshoot Mobile server

- Define the level of log messages stored on the Mobile server using the MicroStrategy Mobile Administration page and view the log files on-line in a table format to debug issues occurring on the Mobile server
- Read messages being sent between the Mobile Application and Mobile server using Fiddler to troubleshoot network connectivity issues
- When connectivity issues with Mobile server are present, try using the device web browser to connect to the Mobile server URL

Coordinate with the Intelligence Center

- Attend daily scrum meeting to report on tasks completed, and notify team members of anything that might be blocking you from completing your tasks according to the project plan maintained by the Intelligence Director

Optimize

Mobile Application Performance

- Remove unnecessary objects such as columns from datasets, images, text boxes, shapes, panel stacks, panels, visualizations, and grids from Documents using MicroStrategy Developer or MicroStrategy Web to reduce the overall Document and respective cache sizes
- Use vector font or vector images instead of .jpg and .png image files to reduce the overall mobile application size to improve load time performance
- Review Cache guidelines and leverage or modify the cache protocol used to optimize performance
- Leverage Prompts to reduce volume of data traversing through the network to the Mobile Applications
- Set selectors to Slice using MicroStrategy Developer or MicroStrategy Web to limit the number of requests the Mobile Application is making to the back end database or in-memory cubes when datasets are small

- Set selectors to Filter using MicroStrategy Developer or MicroStrategy Web to restrict large amounts of data being transferred over a slow network connection
- Combine multiple datasets into one dataset using MicroStrategy Developer or MicroStrategy Web to reduce the footprint of the virtual dataset used in the document
- Do not use OLAP Consolidations and Custom Groups in your datasets in favor of implementing the logic in the Data Warehouse, and surface into MicroStrategy as a regular Attribute
- Test the following scenarios while tracking amount of time to identify areas where the Mobile Application is performing poorly
    - Manually run Mobile Applications to track the load time when nothing is cached
    - Manually run Mobile Applications to track the load time when server caches exist
    - Manually run Mobile Applications to track the load time when the Device cache exists
- Implement a Landing Page or Home Screen for your application that contains mainly images and document links for the screen to load fast and allow pre-cached and subscription cached documents to be loaded while the user determines which link to navigate to within the Mobile Application
- Track actual Wi-Fi network speeds using Speedtest.net at different times and locations where the Mobile Application will be used, to determine the amount of data that can be transferred over your network to meet performance SLA's

Mobile server Performance

- Collect logs from the Mobile Client to create scripts to use with JMeter to simulate concurrent user session loads to test your Mobile Application performance when the Mobile server and Intelligence Server are under a severe load

Mobile Reliability

- Proactively check for Mobile Device OS Updates and test Mobile Applications running on devices with the updated OS ahead of official release to identify and address potential conflicts between the Mobile Application and the device, to prevent users from experiencing the issues with the Mobile Application when they update the OS on their device

- Proactively check for EMM Updates and test MDM and MAM with Mobile Applications to identify and address any potential conflicts between the EMM update and the Mobile Application, to prevent users from experiencing issues with the Mobile Application when the EMM updates are applied
- Proactively check for MicroStrategy Mobile SDK updates and perform regression testing on each Mobile Application with the new SDK client build, to prevent users from experiencing issues with the Mobile Application when the MicroStrategy Mobile SDK is updated
- Manually test Mobile Applications to ensure users have access and are able to log into them
- Work with the Analytics Architect to create reports to verify integrity of data within the Mobile Application and setup Alerts for the Mobile Architect and Analytic Architect to be notified when the data is out of tolerance
- Use the latest supported Mobile Device and OS to achieve top performance in your Mobile Applications by leveraging the additional Memory, CPU, and GPU processing capabilities the latest hardware offers

Mobile Usability

- Establish a process to periodically collect feedback from users of Mobile Applications and identify enhancements to the Mobile Applications that can enhance the user experience
- Proactively monitor Device releases to assess how your Mobile Applications will work within the new Device so changes to the Mobile Applications to accommodate users with those devices can be planned, to prevent users from abandoning your Mobile Applications because of poor user experience on the new device
- Follow up Mobile Application launches with periodic tips and tricks emails to increase the visibility to your application
- Open the DSSErrors.log file in a text editor to proactively check for errors and crashes associated with any datasets and documents used in the Mobile Application and address any issues in future releases
- Enable All in the logging level in the Mobile Configuration for the Mobile Application to enable client side logging to review and analyze the following
    - Total time it takes to load the Mobile Application which includes the Intelligence Server response time, query response time, network transmission time, and client rendering time

- When query response time causes delays, use caches, cubes, or tune the SQL to improve the response time for the Mobile Application
- When Intelligence Server response time causes delays, push calculations performed in the Analytical Engine to the database to reduce the processing time in the Intelligence Server
- When Network data transmission causes delays, modify datasets to use prompts to reduce the amount of data being transmitted to the Mobile Application
- Use external tools such as Wire Shark or Speedtest.net to analyze network connectivity performance periodically
- Break large documents down into additional documents with prompted datasets to reduce the load time of the Mobile Application

Mobile Caching

- Use Device Caching in Documents when the document must execute datasets and then store the results in a cache on the Mobile Device until the cache is expired or invalidated
- Use Subscription Caching when personalized caches are necessary due to the dependency of prompts and or security filters within supporting datasets so that caches are delivered to the device when the user opens the Mobile Application
- Use Pre-caching when the document and supporting datasets are not personalized and caches can be updated via update cache subscriptions, which are delivered to the device when the user opens the Mobile Application
- Use Apple's APNS to enable background downloads of caches, which are used with iOS Mobile Applications when lightweight caches need to be delivered to the Mobile Application ahead of the user opening the Mobile Application
- Use Real-time document caching when data can be stored locally on the Mobile Device for a specific amount of time specified in the Mobile Configuration for the Mobile Application
- Use folder caching in the Mobile Configuration for Mobile Applications that allows users to selectively run MicroStrategy Reports individually to reduce the amount of data being transferred to the client as the user navigates through folder structures
- Enable Clear Caches on Close in the Mobile Configuration for your Mobile Application to remove all caches from the Mobile Device your Mobile Application either retrieved or created during the user session

- Enable Clear Caches on Logout in the Mobile Configuration for your Mobile Application to remove all caches from the Mobile Device your Mobile Application either retrieved or created prior to the logout initiated by the user
- Enable Mobile Device caching for Documents and Reports using MicroStrategy Developer to improve the performance of the Mobile Application

Definitions

Term	Definition
.apk file extension	Android Package File
.ipa	iOS Application Archive File
Device Cache	Document layouts with corresponding data stored on mobile device
Alerting	A subscription that sends push notifications to mobile devices when a metric condition is met
	Open-source operating system used for smartphones and tablets
Android SDK	Set of development tools used to develop applications for devices running the Android operating system
Android Studio	Provides the fastest tools for building apps on every type of Android device
APNS	Apple Push Notification Service provides a mechanism for developers to push messages to mobile devices
App Config	XML file that is used to define configuration key value pairs for an application
BYOD	Bring Your Own Device, refers to individuals who bring their own mobile device to connect to enterprise networks and run enterprise applications
Cache	Optimize performance by bypassing key bottle-necks on the server and network side
Capacity Planning	Process of determining the production capacity needed by an enterprise to meet the changing demands for its products

Term	Definition
Compile	To build a iOS or Android native app with the MicroStrategy Mobile SDK
Corporate Style Guide	Dictates the fonts, colors, and logos to be used in applications or marketing documents
Data Integrity	Assurance of the accuracy, consistency, and availability of data
Datasets	A grid report in MicroStrategy that is used in a Document or Dossier
Device Type	Phone or Tablet
Diagnostics	The logging of errors, warnings, or messages that occurs on the Mobile server
Drawing Tools	Draw.io or PowerPoint allow you to use shapes, text, and lines to represent application objects and depict workflows of an application
Enterprise Mobile Management (EMM)	Software that allows organizations to securely enable employees' use of mobile devices and applications
GCM	Google Cloud Messaging service provides a mechanism for developers to push messages to mobile devices
Home Screen	A document that is configured to be loaded when a Mobile Application is launched
Identity Management System	A system used to manage individual identities, their authentication, authorization, roles, and privileges within or across system and enterprise boundaries
iOS	An operating system used for mobile devices manufactured by Apple Inc.
iOS Enterprise Developer Program	Allows you to develop proprietary, in-house iOS and watchOS apps that you can distribute to your users in your organization and outside the store
iOS SDK	Set of development tools used to develop applications for devices running the iOS operating system
Links	A mechanism used to guide the user from one document to another while passing information

Term	Definition
Mobile Client	Native mobile application created by customizing the MicroStrategy Mobile SDK for iOS or Android
Mobile Configuration	List of settings for a MicroStrategy Mobile application that are stored in a XML file on the Mobile server
O-Auth	Open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords
Off-line	Device is not connected to the Internet via Wi-Fi or Cellular
Pre-Caching	Store On-line content on a mobile device for Off-line use
Push Notifications	A notification that is pushed (delivered) from back end server or application to user interface
Regression Testing	Software testing that ensures previously developed and tested software still performs the same way after it is changed
SAML	Security Assertion Markup Language is an open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider
SDK Integration	SDK integration is when an application contains development tools from two or more software packages that are brought together
Security	Measures taken to guard against espionage or sabotage, crime, attack, or escape
Service Level Agreement (SLA)	Official commitment between a service provider and service consumer where quality, availability, and responsibility of the service are agreed on
SSO	Property of access control of multiple related, yet independent, software systems. With this property, a user logs in with a single ID and password to gain access to a connected system or systems without using different user names or passwords, or in some configurations seamlessly sign on at each system

Term	Definition
Statistics	Data captured which can be used to interpret or analyze Mobile server performance
Test Flight	TestFlight is an on-line service for over-the-air installation and testing of mobile applications, currently owned by Apple Inc and only offered to developers within the iOS Developer Program
User Interviews	A meeting the Mobile Architect facilitates with users who use the mobile application frequently as well as those that don't use the mobile application often, to incorporate feedback on features and functionality to incorporate in future releases to increase user engagement
Virtual Dataset	In-memory dataset generated by joining datasets within a document or dossier at run-time
Visual Prototyping	Axsure or Balsamiq have predefined objects that allow you to easily build out the visual design blueprint of an application
Xcode	Integrated Development Environment (IDE) for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS

Copyright Information

Trademark Information