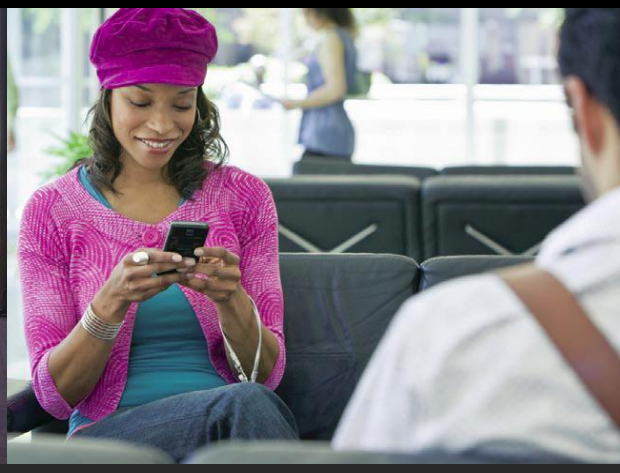
SYBASE®
An SAP® Company

Brand Mobiliser User Manual

[PRODUCT DOCUMENTATION]





www.sybase.com

Contents:

1 Introduction 1
1.1 References 1
1.2 Conventions 1
1.3 Brand Mobiliser Architecture 2
1.4 Workspace 3
1.5 User 4
1.6 Short or Long Code, and Keyword 4
1.7 Brand Mobiliser Applications 5
1.8 Brand Mobiliser Events 6
1.9 Brand Mobiliser Subscribers 6
1.10 Channel 7
1.11 Activation 7
1.12 Development and Deployment Model 8
1.13 Date and Time 9
2 Features and User Interfaces 10
2.1 General User Interface Features 10
2.2 Login 16
2.3 My Account 18
2.4 Dashboard 18
2.5 Assets 22
2.6 Application 27
2.7 Manage Categories 38
2.8 Activate Applications 39
2.9 Events 40
2.10 Activate Events 49
2.11 Subscribers 49
2.12 Simulate Application 53
2.13 Default Menu 56
2.14 Reports 58
2.15 Workspace Administration 60
3 Installation and Configuration 68
3.1 Prerequisites 68
3.2 Software Binary 68
3.3 Database Scripts Installation 69
3.4 Stand Alone Deployment 74
3.5 J2EE Web Deployment 84
4 Platform Maintenance & Tuning 87
4.1 Database Tables Maintenance 87
4.2 Special Maintenances 88
4.3 Log Files 89
4.4 Performance Tuning 90
A. Appendix - Glossary 91
B. Appendix - User Roles 92
C. Appendix - Regular Expressions 94

Principal Author

Sybase Mobiliser Group

Revision History

Version 1.2 - September 2011

Version 1.1 - March 2011

1 Introduction

Brand Mobiliser makes it very easy for companies to mobilise all aspects of their businesses, including: brand awareness, CRM, mobile banking and financial, mobile payment, commerce, CRM, and much more. Brand Mobiliser also fulfills the mobile lifestyle of consumers that increasingly demands the ability to get more things done from their mobile phones using simple mobile interactive applications.

The Brand Mobiliser's web interface provides tools to visually ***compose*** a mobile interactive application, ***test*** it using the built-in simulator, and ***deploy*** it to the processing engine immediately ready to interact with the mobile consumers. The "Live" applications can be easily modified in real time, to meet the new business needs, and then redeployed without disrupting the service availability. This is critical in the mobile world with billions of subscribers globally demanding high availability.

Brand Mobiliser's high-performance and scalable processing engine is designed to serve the continued growth of mobile traffic demanding instant interactions with the mobile applications. The node architecture is suitable for cloud infrastructure deployment and is based on Sybase® 365 experiences in delivering billion of messages daily.

Brand Mobiliser is powered by AIMS, based on the OSGi™ modular architecture, and is fully extensible to further enrich the mobile interactive experiences. Plug-in components can easily integrate with existing corporate enterprise systems, or our partner's and other 3rd party systems using the Service Oriented Architecture (SOA).

1.1 References

1. Brand Mobiliser Development Manual; Version 1.2 – September 2011
2. Brand Mobiliser State Developer's Guide; Version 1.2 – September 2011

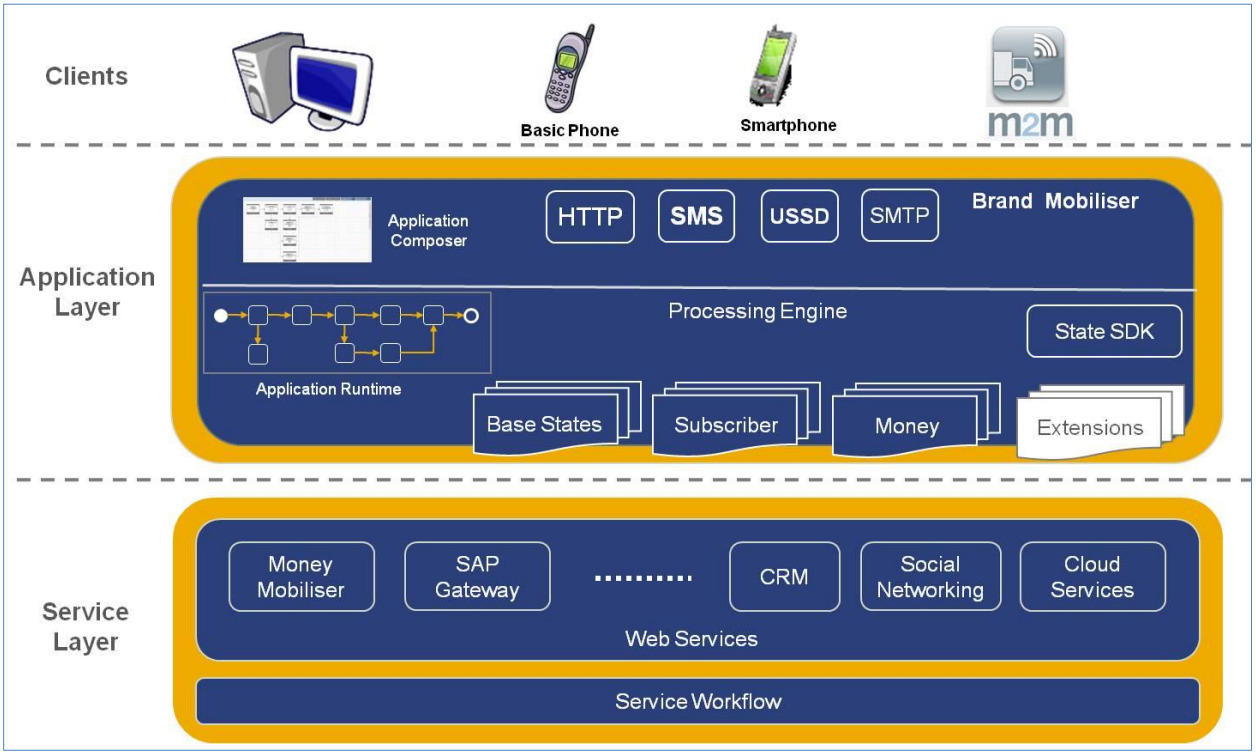
1.2 Conventions

The following conventions are followed within this document:

- Application – refers to either Interactive or Event applications; a process workflow consists of a sequence of connected states
- BrandUI – refers to the Brand Mobiliser web user interface. There are two parts to Brand Mobiliser: the web user interface for human interaction - BrandUI; and the processing engine that serves as a running container for the application – BrandEngine
- BrandEngine – refers to the Brand Mobiliser processing engine that serves as a running container for the application.
- **Manage Workspace** – bolded font indicates screen or page name. The word screen and page are used interchangeably to represent the same thing
- **Logout** – indicates a hyperlink to a screen, or an action hyperlink
- ***"Workspace"*** – bolded italic font and surrounded by quotation indicates a word that is used multiple times in this document. When presented for the first time, the font style is ***bold italic*** and surrounded by quotation; and definition may be provided if necessary
- *at least one assigned unique short code* – underlined and italic indicates that the statement is important
- "My Account" – indicates a link to the relevant section in the document, in this case, the **My Account** section. Clicking on it will move the display to the linked section
- **Best Practice** – added when necessary. The Best Practice provides recommendation so that novice users can avoid implementations that lead to system limitation(s)
- **NOTE: indicates additional notes, and the details are also bolded.**

1.3 Brand Mobiliser Architecture

The following figure presents a high level overview of the Brand Mobiliser platform. Like any other Sybase Mobiliser Product family, Brand Mobiliser is hosted in the AIMS server, that is based on the Apache-based implementation of OSGi™ specifications[1] kernel called "felix". The reader or user are not expected to be familiar with the concept of OSGi™ to use Brand Mobiliser.



The three major components of Brand Mobiliser are: "***Application Management***", "***Processing Engine***", and "***State SDK***". The Application Management is a web UI that is used to develop, deploy and manage applications among others. Applications are developed using the "***Application Composer***". Once completed and tested, the applications are deployed to the processing engine for execution at runtime.

An application is a process workflow consists of a sequence of connected states. A states is the basic building block with a specific capability. The state's capability can be implemented natively within the state (***"Stand alone state"***) or it can be proxy to an external web service (***"Service state"***). State can also be an aggregation of external or internal corporate services.

At runtime, the processing engine executes the application by stepping through the states conditionally. In addition, the processing engine also creates a ***"session context"*** for each mobile subscriber request. Each state is processed and the result is evaluated for choosing the "***follow-up***" state. Some states are synchronous, or the result is available immediately for evaluating the follow-up state. Other states are asynchronous, such as the Send SMS state that sends SMS message to the subscriber's handset and waiting for the subscriber's response before it can evaluate the follow-up state. After processing an asynchronous state, the engine will hibernate the application flow and its corresponding session context. When the response is received, the engine will reload the application instance and its session context, and evaluate the follow-up state based on the response. The processing engine is

[1] http://www.osgi.org

optimized for high throughput session management enabling responsive customer interaction experiences with the application.

Brand Mobiliser provides several basic stand alone states including: basic states for composing workflow referred to as "***Base State***", as well as states for performing operations on the Subscriber storage, or "***Subscriber State***". In addition, service states for integrating with the Money Mobiliser platform are also included out of the box.

The third component is the "State SDK". The State SDK enables 3rd party developers, partners and customers to develop domain specific states, or "***Custom States***". Custom states are developed for various purposes including: to enrich the application capabilities, to integrate with the existing systems such as: CRM, ERP, MRP, enterprise or mobile services, cloud or social networking services, as well as machine-to-machine (M2M) infrastructure, or for back office automations. Custom states can be added to Brand Mobiliser dynamically using the plug-in system enabled by the OSGi™ Services Registry. Please consult the reference "Brand Mobiliser – State Developer's Guide" for more information on how to develop custom states that are pluggable to the Brand Mobiliser platform.

1.4 Workspace

Brand Mobiliser adopts the concept of "***Workspace***" that is a logical grouping of users who are collaborating on the same projects or tasks. Workspace can also be used for partitioning the development, QA, and production environments, as discussed in the "Development and Deployment Model" section. However, more importantly, workspace need to have at least one *unique short or long code*. More than one codes can be assigned to the workspace but they *cannot be shared across multiple workspaces*. Please refer to "Short or Long code, and Keyword" section for discussion on how short or long codes are used in Brand Mobiliser. Each Brand Mobiliser deployment (will be referred to as **"*Platform*"** in the rest of this document), can have multiple workspaces.

In addition, a workspace can have only one outbound "***Channel"*** connection. The inbound channels are not specific to a workspace. All incoming traffics are routed to the corresponding workspaces based on the destination short or long code found in the message. Channel(s) are defined on the platform level, and are shareable by all the workspaces. There are different deployment strategies for partitioning the outbound traffics. Please consult Sybase if the need arises. Please refer to "Channel" section for detailed discussion of *channel*.

Once the workspace is created, users may be assigned to the workspace. A user is assignable to more than one workspaces. The default installation of the platform has a pre-defined workspace called "default", and a pre-defined user called "admin" who is initially assigned to this "default" workspace. The "admin" user has the "SUPER ADMIN" role that should be treated as the platform administrator. Both the "default" workspace and "admin" user should not be deleted, but please change the password immediately after installation.

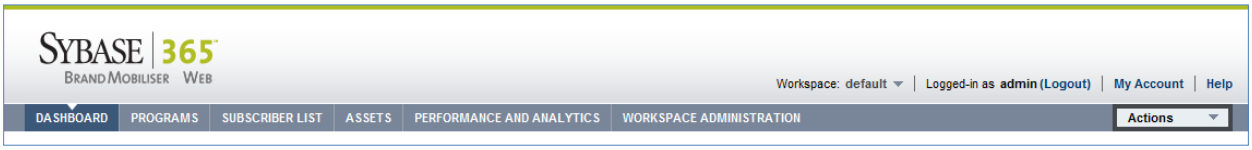
Note: it is possible during user creation to not assign a workspace to the user. However, during saving the application will automatically assign the user to a "default" workspace. Automatic assignment also occurs when modifying the user if no workspace is assigned.

Best Practice:

- Do not assign the pre-defined "default" workspace to any users including the newly created one. Preserve the initial setup of the "default" workspace assigned to the pre-defined "admin" user only. Treat the "default" workspace as a "*guest*" workspace so when users are in-advertently assigned to the "default" workspace, no damages can be done.
- Do not assign any short codes to the "default" workspace for reasons described above.
- Do not setup any channels on the "default" workspace for reasons described above.
- There are two methods to disable a user. The first method is to disable the user from the **Manage User** screen, and the user will not be able to login to the platform. The second method is to assign the user to the "default" workspace and remove all the assigned roles. In this case, the user is still be able to login but will not have access to the active workspaces. The latter method is acceptable provided the above recommended practices (items 1 to 3) are also adopted.

1.4.1 Change Workspace

As discussed earlier, a user may be assigned to multiple workspaces. After login, the user with multiple assigned workspaces will be placed automatically into one of the workspaces. The user may change to another workspace using the dropdown control (called Workspace) displayed on the "Login Status Bar", as shown below.



The workspace control shows the current workspace that the user is in, and when the down arrow (▼) is clicked it will show all the workspaces that are assigned to the user. Selecting a workspace from the dropdown list and clicking to accept it will invoke the change workspace action. The user will be placed in a new workspace and the screen will be navigated to the **Dashboard** screen. Since the workspace control is in the header, it is accessible from any screens. However, the user should be aware that when the workspace is changed the screen will always be navigated to the **Dashboard**. This is necessary because each workspaces have their unique artifacts, such as: short codes, channel, interactive applications, etc. So depending on the current displayed screen, when changing the workspace the displayed artifacts may no longer be valid. Therefore, the user is navigated to the **Dashboard** to simulate a successful login experience and hence avoid confusion.

By design, the user cannot work across workspaces. The user cannot see artifacts from multiple workspaces.

1.5 User

Brand Mobiliser expects a pre-defined user at startup. If the default database scripts is installed, the user is "admin", and the initial password is "brand". The "admin" user will have the "SUPER ADMIN" role which is referred to as the "***platform administrator***". **NOTE: the first thing a platform administrator should do is to change the password and to save the password in a safe location.** There are certain tasks that can only be performed by the platform administrator including: creating workspace, setting channel, and stopping or restarting the processing engine. In practice, the platform administrator creates and configures workspaces, assigns the "***workspace administrator***" role to user(s).

The workspace administrator is a user with an "ADMIN" role. The workspace administrator can create users and assign them to any workspaces that the workspace administrator have access to (or is assigned to). **Note: users with an "ADMIN" role and assigned to more than one workspaces, are automatically the workspace administrator for all those workspaces.** For role details, please refer to "Roles, Purposes and Accesses" section.

Brand Mobiliser does not allow deleting users. Users can be disabled using the **Manage User** screen. When attempt to login, the disabled user will see the invalid user id or password message.

Best Practice:

- Remember to change the password of the "admin" user
- For enhanced security, create a new user and assigned the "SUPER ADMIN" role and the "default" workspace. This user will be the new platform administrator. Then disabled the "admin" user.

1.6 Short or Long Code, and Keyword

As described in the "Workspace" section, workspace has unique short or long code(s). These codes are used by the Brand Mobiliser processing engine (BrandEngine) to identify the corresponding workspace. When the platform administrator enters a short or long code for a workspace on the **Manage Workspace** screen, the code will be validated to ensure that it has not been assigned to other workspaces.

For any incoming messages to Brand Mobiliser, the destination MSISDN will be used by the BrandEngine to find a match to the workspace using the workspace short or long code list. Once the workspace match is identified, the engine will use the keyword(s) from the text content of the incoming message to match the application. A workspace may contains many applications and they should have unique keyword(s). However, the unique application keyword requirement is not validated or enforced during keyword assignment to the application from the UI. This is due to the allowed flexibility to assign regular expression to the keyword. The engine will always use the first match application and ignore the rest.

The use of short or long code in Brand Mobiliser should not be confused with how it is used in the bigger picture of the mobile operator world. Short codes (as defined in the Appendix A2) are often associated with mobile services, such as Brand Mobiliser interactive applications, and they are assigned by the mobile operator to the owner of the services. A real example will make it clear.

For example, company XYZ is interested in providing a mobile service for paying street–side parking in San Francisco Financial District. XYZ will apply (with the help of Sybase 365) an assigned short code from a mobile operator. Typically, this assigned short code (let's say 9999) is advertise on the billboard in the San Francisco Financial District area, for example, "For paying parking with you mobile phone, text SFpay to 9999". When the mobile subscriber sends an SMS message to 9999 with SFpay in the content, the message will first reach the mobile operator and then the operator will route it to Brand Mobiliser server. So it is important to understand that the process of routing a short code (i.e., 9999) to Brand Mobiliser server is done by the operator, and has nothing to do with the Brand Mobiliser. When the message is received by the Brand Mobiliser server, the BrandEngine will map the destination MSISDN (*i.e.*, the short code) to a workspace. Once the workspace is identified, the engine will look at the keyword (*i.e.*, SFpay) and mapped it to the corresponding interactive application in that workspace.

Best Practice:

- Each interactive applications should have at least one assigned keyword
- Check that the keyword has not been used by other application in the workspace using the "Keyword Usage" tool provided on the **Keywords** screen. Note: Brand Mobiliser UI does not validate or prevent duplicate keyword usage. When a keyword is used by multiple applications, Brand Mobiliser does not guarantee which applications get dispatched. In fact, it may change on each occurrences.
- When a regular expression is used as a keyword, the application developer needs to ensure that the regular expression does not overlap with existing keywords already in use by other applications

1.7 Brand Mobiliser Applications

"***Application***" is defined as a process workflow that consists of a sequence of connected states. The application is executed by the Brand Mobiliser processing engine (BrandEngine) at runtime. There are two types of application: Interactive and Event, differ by how they are invoked and used. Interactive application is mainly used for providing mobile services with rich user interaction. This type of application is typically invoked by mobile customers sending a keyword to a pre-assigned short or long code.

The Event application was first introduced in Brand Mobiliser version 1.2. Unlike the interactive application, the event application is mainly designed for workflow or batch processing, with the ability to send outbound messages but no user interaction capability. The development of the application flow is the same as the interactive application, using the **Application Composer** screen. The event application is invoked by event, such as: scheduled times, system triggers or external triggers. So, unlike the interactive application, the event application is not associated with keyword(s).

Managing applications, including: create new application, importing an application from a file stored in the user's computer, and also import application(s) from the available templates, are performed using the **Assets** screen. All of these functionalities will be covered in details in the "Assets" section.

1.8 Brand Mobiliser Events

"***Event Model***", or "***Event***" for short, was introduced in Brand Mobiliser version 1.2 to support the Event type application. As described earlier, the event application is designed for workflow or batch processing that will be triggered by event, such as scheduled times. The Event model is a container for storing the configuration details and relationships including: active runtime, scheduled window (manual or recurring), an event application to trigger when the scheduled window is current, and all related interactive applications.

For example, you can create a "Promotional" event that is planned between Nov. 1 and Nov 30. Within this event "***Runtime***", you can define event "***Window***"(s) when the "***Trigger***" will occur. Window can be specified manually by specifying the start and stop date-and-time. Alternatively, a recurring window, for example, daily type with the start and stop times. In addition, an event application (e.g., Push discount code to Subscriber) is assigned to the event such that when the event trigger occurs the application is invoked. The message pushed to the subscribers may contain "call-to-action" for the subscriber to reply with, for example, the keyword "coupon" to receive the discount code. The "coupon" keyword will be handled by an interactive application, let's say, the "Promotional Discount Code" application. The interactive application can also be assigned to the event as a marker to prevent it from being deleted by others sharing the same workspace.

1.9 Brand Mobiliser Subscribers

The goal of introducing storage in Brand Mobiliser was driven by the need to upload and store Mobile Subscriber attributes that can be used to perform push campaigns, hence the storage is referred to as "Subscribers". However, you will realize as you become more familiar with the Brand Mobiliser Application model that the "Subscribers" storage is a general purpose storage available to any Brand Mobiliser Applications. NOTE: it is likely that in the future release of Brand Mobiliser the "Subscribers" term is modified.

Ideally, this storage should be used as a non-durable storage for purposes of: staging, or in transit storage pending batch transfer to the system of record. The reason is the database schema used by this storage was designed to be very generic, and therefore was not fully optimized for large scale and more domain specific purposes, such as for CRM, ERP, etc. However, it is perfectly fine to use the storage as the system of record for small scale implementations, and especially with regular and well housekeeping.

The subscriber storage is referred to as "Set". Set has a unique ID. The set name is not unique but to avoid confusion it should be kept unique. The set is made up of a list of rows with 21 fields. The first field (KEY) is the unique "key" while the remaining 20 fields are free form. The first free form field (ATTRIB1) can store up to 1000 characters while the remaining free form fields store up to 320 characters.

The design of the "Set" was heavily based on the need to use the storage for subscriber list. For example, when used for subscriber list, the "KEY" field will be used to store the mobile subscriber MSISDN. The first free form field (ATTRIB1) can be used to store a long message. Even though, each SMS message is limited to about 160 chars (varies based on the type of character encoding used), Brand Mobiliser will automatically send additional SMS messages when the message is longer. So ATTRIB1 can be used to store long messages. The remaining free form fields can be used to store additional attributes corresponding to the subscriber. These attributes can be used in many different ways, for example: to dynamically "mail-merge" into the SMS message template, to send customized SMS message based on the attribute, to filter the number of SMS messages sent based on certain attribute(s), etc.

The "Set" can easily be transferred to the durable or system of record storage. The simplest and manual mechanism is to export the "Set" to a comma separated value (CSV) file. The CSV file can easily be read by any spreadsheet programs, import to any reporting applications, upload to databases, process by custom build applications, etc. If an automated process of transferring the "Set" is needed, a Brand Mobiliser application can be composed using simple custom built plug-in states that batch sends or uploads the set to the target system. The batch process can easily be scheduled using the Brand Mobiliser Event system.

The "Set" can be populated by uploading a CSV file from the **Upload Subscriber** screen, or an empty "Set" can be created from the **New Subscriber Set** screen and automatically populated by the Brand Mobiliser Application(s).

For example, a Brand Mobiliser application can be composed to batch pull the subscribers from the system of record (i.e., CRM system, Twitter followers, Facebook friends, LinkedIn connections, etc.), or when a mobile subscriber opt-in by responding to a mobile interactive service created using the Brand Mobiliser Interactive application, or collecting data for reporting, etc.

1.10 Channel

Channel is the conduit in Brand Mobiliser for delivering inbound or outbound messages. There are three built-in channels: SmsOutDummy, SMPP and JMS. The SMPP and JMS channels are configured using the **Manage Channel Configurations** screen that is only accessible to the platform administrator. The SmsOutDummy is the loopback channel and it is used for simulation test in the development environment. When a new workspace is created, the outbound channel is automatically assigned to SmsOutDummy.

Channel configuration can only be performed by the platform administrator for security and system stability reasons. Channel can be configured with one or more connections. These connections can be set to "Active" or "Close". Active connections on the inbound channels will receive incoming messages. Active connections on the outbound channels will send outgoing message, and can be assigned to workspace(s) using the **Manage Workspace** screen. **NOTE**: channel connections are platform wide configurations, and therefore the connections can be assigned to more than one workspaces. Details on how to configure channels are presented in the "Workspace Administration – Manage Channel Configurations" section.

Note: Changes including addition, deletion or modification of channel connections will require the processing engine restart. This is done using the Manage Channel Configuration screen. Restarting the engine will impact the running applications in all workspaces but will preserve the hibernated session context.

1.11 Activation

Brand Mobiliser serves as both the development and runtime environments for applications. For this reason, it stores two different versions: "In-Review" and "Live". Any changes made on the Brand Mobiliser web UI are saved as "In-Review" version. The processing engine executes the "Live" version. The process of copying the "In-Review" version to the "Live" version is called "**Activation**" or *deployment.* Activated item will be "**In Service**" when the current date becomes equal to the start date of the item.

Activation process is used for the following items of Brand Mobiliser: Default Menu, Application and Event. Newly created item will need activation, and any changes made will require re-Activation. Changing "In-Review" version will not impact the "Live" version until after successful activation.

A good understanding of the activation concept is essential especially when there is a "Live" version running. Once the activation is completed, the changes are committed and it cannot be rolled back. **It is "Live"**. If mistakes were made and changes cannot be rectified quickly, it may be necessary to shutdown the application. There are many ways to accomplish this, as follows:

- Change the "Active From" date to a future date, and the re-activate. This is essentially making the item live but not active until the specified future date. This will work for: Application and Event.
- For interactive application, remove the keyword(s) and re-activate. This essentially prevents the application from being invoked. It is live but cannot be invoked.
- A more destructive method is to export the application, followed by deleting the application. This will removed both the "Live" as well as the "In-Review" versions. Prior to exporting the application, make sure that the "In-Review" version is the same as the "Live" version. If you are not sure, then first export the application and save it to a name like "appInReview". Then, go to the **Revert All Changes** screen. If the "In-Review" is different from the "Live" versions, the revert "Commit" button will be enabled. Click "Commit" to copy the "Live" back to the "In-Review" versions. Then export the application and save it to a name like "appLive". For more details, refer to the "Application" section.

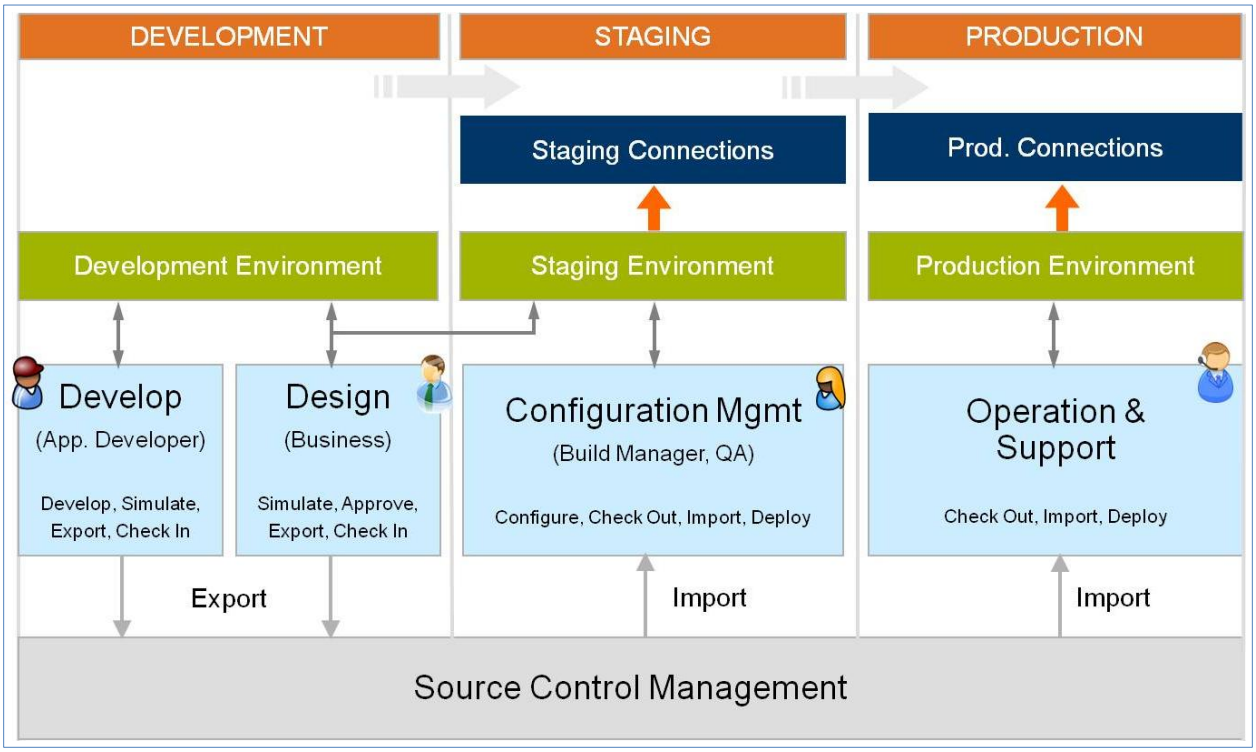
- There is no export capability for event so you just have to delete the active event. Note: if the event has a running window, you need to stop the window first before deleting the event. For more details, refer to the "Events" section.

Note: There is no mechanism to stop the Default Menu as it is designed to be live at all time. To stop the Default Menu can only be accomplished by disconnecting the outbound channel from the workspace (discussed above).

1.12 Development and Deployment Model

Brand Mobiliser platform provides both the development and deployment environments for Brand Mobiliser applications. The application development environment is the Web Application Composer. The deployment environment is the processing engine that provides the runtime container for the applications.

Although both the development and deployment can be done in the same environment, this practice is not recommended for the production environment especially those hosting mobile services that require high performance and availability. The following figure illustrates the recommended best practice development and deployment model.



Application requirements are typically captured and prepared by the business analysts. The analysts can take it a step further by modeling the requirements in the form of high-level business flow using the Brand Mobiliser Application Composer in the "**Development Environment**". When the high-level flow is approved, the analysts or another application developers can proceed to refine the application flow and configure each states in the application using the "**State Property Editor**" that is available in the Application Composer. This task is also performed in the development environment. The development environment will be setup such that both the analysts and developers will have access to the same workspace. So, they can collaborate on developing as well as testing the applications. Testing is done using the "**Application Simulator**". In addition, everyone should have the "ADMIN" roles in the development environment so that they can activate applications for testing. Note: applications need to be activated before they can be tested using the simulator. The development environment should not have the channel with active connections. The default SmsOutDummy will be used while performing

the simulation test in the development environment. Work in progress or completed applications should be "**Exported**" to an XML files, and these files should be checked-in to the source control management (SCM) system as necessary following the company policy.

When the application development is completed and checked-in to the SCM, the next step is performing integration testing in the "**Staging Environment**". The staging environment should have the active channel connection, and ideally it should be connecting to the corresponding staging environment as well. The QA testers can check out the applications from the SCM and then import it to Brand Mobiliser. The applications need to be activated prior to testing, either using the simulator or the real mobile handset. Since the staging environment has an active connection, it should be more restrictive in regards to user role assignment. For example, the "SUPER ADMIN" role should only be assigned to the platform administrator, who configures channel connections, and creates the QA workspace and assigns the QA workspace to the workspace administrator. The QA manager can be assigned as the workspace administrator. The QA manager can add users (i.e., testers) to the QA workspace. The testers should be assigned to the "APPLICATION ADMIN" role so that they can import, activate and test applications. The business analysts and developers could be given access to the staging environment with the "APPLICATION OWNER" role so that they can check the testing traffic report for troubleshooting reported bugs. They can create applications but are not able to activate them.

Finally, the "**Production Environment**" should be the most restrictive in terms of access. Applications should be checked out from SCM, imported into the production environment, activated and performed the final validation tests. Typically, to make the traffic reports from the production environment available to the business analyst or developer for troubleshooting, they should be exported into CSV file. In addition, they should be sanitized from sensitive information for security reasons. **Note: for security compliance, please refer to your corporate policy.**

The different environments discussed above can be setup in many different ways, such as, using different workspaces, using different instances of Brand Mobiliser on the same physical server, or using different physical servers or virtual machines (VMs). When using different workspaces, each environments will correspond to a workspace. User access to the workspace needs to be designed and managed. In addition to the above setups, Brand Mobiliser can also be installed on the developer's computer giving the developer full control and access to the platform. This setup is necessary for developers who are extending the capabilities by implementing new states using the "**Brand Mobiliser State SDK**".

NOTE to Application Developer: since there are only two versions (In Review and Live) provided by Brand Mobiliser, it is a good practice to export the application to your local file system when a certain milestone is reached during the development. The exported application file can be versioned and checked in to the source control management system for the version history. The checked in file also acts as a backup.

1.13 Date and Time

Brand Mobiliser provides runtime environment to execute brand mobiliser applications. The web application is the tool for configuring and deploying the applications to the runtime environment. Therefore, all the entered date & time need to be based on the server's date and time. For example, the application active from and to, the event runtime from and to, the window's date & time, etc. In most screens where the date and time entries are needed, the reference server current time is also displayed to aid in establishing the correct time.

2 Features and User Interfaces

The design of the Brand Mobiliser user interface focused on usability from the beginning. The UI was designed and reviewed by a team of professionals specializing in information architecture, interaction and visual design, and branding. The ultimate goal is to give our customers a user friendly tool to accomplish their main tasks quickly. Brand Mobiliser is truly a usability driven application and will continue to be enhanced with the same philosophy.

Brand Mobiliser user interfaces (BrandUI for short) provide the following functionalities on the top level: Dashboard, Events, Subscribers, Assets, Reports and Workspace Administration. They are accessible from the main Navigation Bar. Each functionalities will be described in great details on The following sections will explore each of the areas of the BrandUI in greater detail.

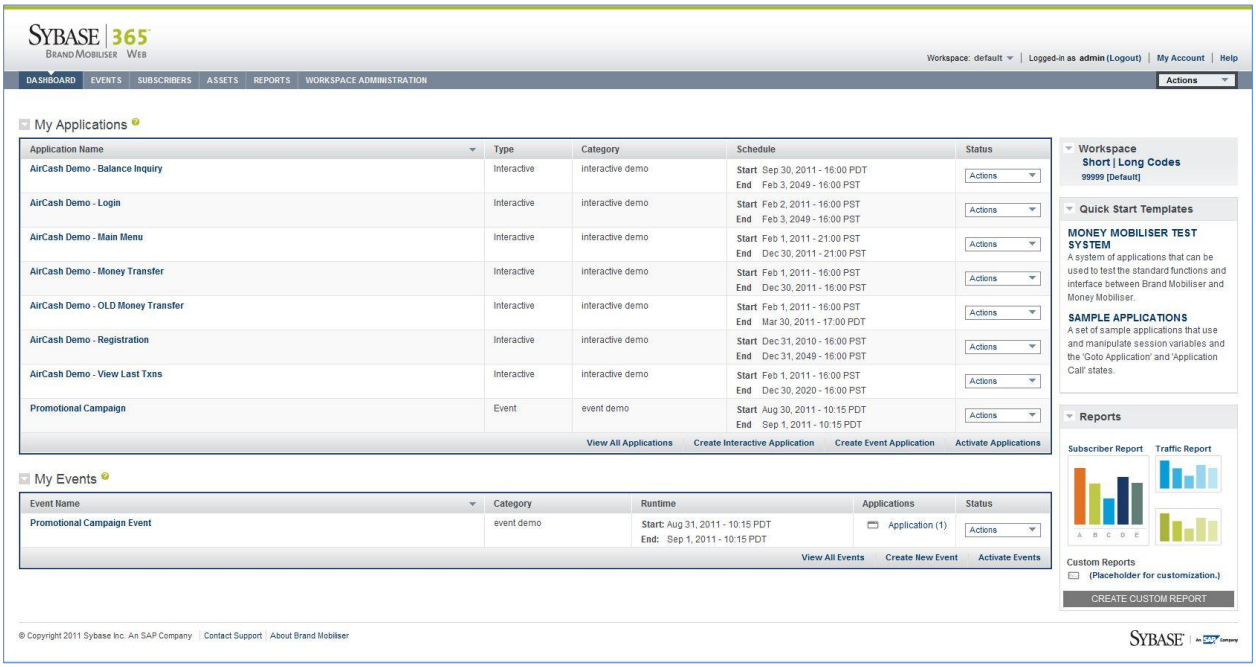
BrandUI has been tested and optimized for compatibility on the following browser versions:

- Microsoft Internet Explorer, version 8 or higher
- Mozilla Firefox, version 3.6 or higher
- Google Chrome, version 10 or higher

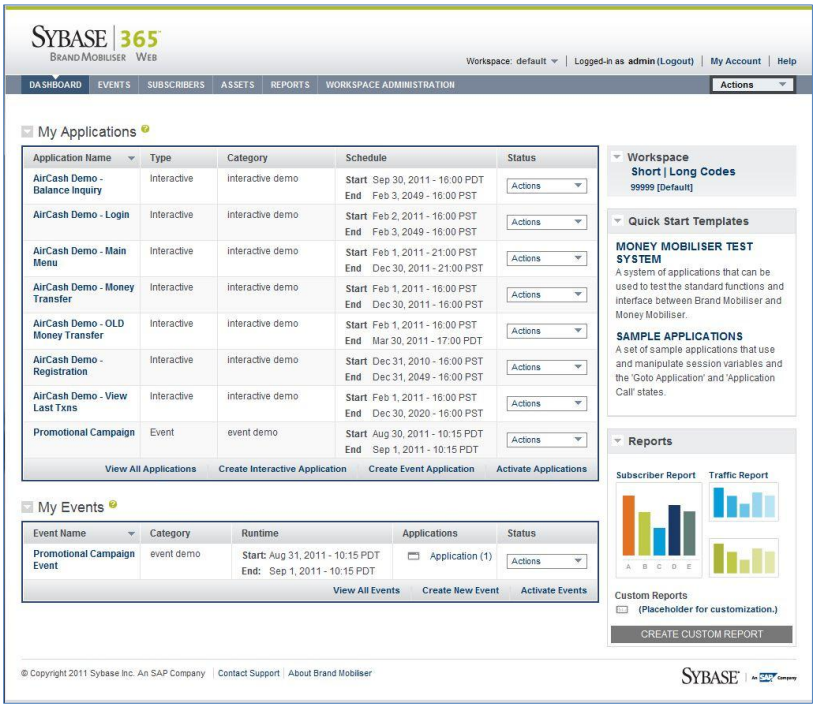
2.1 General User Interface Features

2.1.1 Screen Layout

BrandUI adopts a fluid page layout so that users with wider screen can benefit from it. Certain areas of the page will stretch according to the browser size. Users can adjust the browser size to reduce empty spaces. The following screenshots show the same **Dashboard** screen with different adjusted browser sizes on a 22-inch widescreen monitor. The top screenshot shows the browser in maximized mode, while the bottom shows a reduced browser screen.



Maximized Dashboard Screen



Reduced Dashboard Screen

2.1.2 Login Status Bar

The "***Login Status Bar***" displays information such as the login username, the current workspace, and provides links to **Logout**, **My Account** and **Help** screens.

Workspace – displays the current workspace that the login user is in. If the user has more than one assigned workspaces, the dropdown list will show all the assigned workspaces. The user can change to a different workspace by selecting from the dropdown list. The dropdown list is sorted alphabetically.

Note: The login status bar is on the header and shown on every screens allowing users to change workspace from any screens. Although changing workspace does not require the user to login again, but the user is moved to a different environment with completely different accessibility and artifacts including: application, events, subscribers, etc. Therefore, after changing the workspace the user is navigated to the Dashboard screen simulating a successful login experience, and with artifacts from the current workspace displayed.

After a successful login, the user is automatically placed into a workspace, the login workspace. Currently, the default login workspace is the first workspace assigned to the login user. For example, if the user were assigned to ws1 and then Aws3 later. After a successful login, the user will automatically be placed in workspace ws1 and the workspace dropdown list will be shown in the following order: Aws3 and ws1. The workspace administrator is able to change the default login workspace by reassigning the workspaces.

Logout – for user to logout.

My Account – for user to find out the account information including: assigned workspaces and roles, as well as the authentication source. If the authentication source is "Brand Mobiliser Database", the "Change Password" component will be displayed allowing the login user to change password.

Help – currently not active. This is a place holder for future online help.

2.1.3 Navigation Bar

The "***Navigation Bar***" stretches across the bottom of the header. The navigation bar is kept to a single level by design for ease of usability. Clicking the "***Navigation Item***" will navigate to the "***Main***" screen of the corresponding item. For example, when clicking the "Dashboard" the screen will navigate to the **Dashboard Main** screen which will be referred to as just **Dashboard** screen for short in the rest of this document. There are two types of "Main" screen design format: "***List Table***", and "***Module***", and they are described in details in the "Main Screen Layout" section. In addition, when selected the navigation item will be highlighted with a darker color and a downward pointing arrow head on the top center of the navigation item, as shown for the Dashboard in the figure below. The navigation item highlight will remain while navigating to the sub-screens or related screens of the selected item.

The navigation item will also be highlighted when the mouse pointer is hovered over it. The visibility and availability of the navigation item is governed by the user's role. For example, the Workspace Administration is available to users with SUPER ADMIN and ADMIN roles only. Currently, the navigation items include:

- Dashboard
- Events
- Subscribers
- Assets
- Reports, and
- Workspace Administration

Dashboard – when clicked the screen will display the **Dashboard** screen, that is also shown immediately after a successful login. **Dashboard** screen provides summaries of most commonly or recently used information and artifacts. See "Dashboard" section for details.

Events – the **Events** screen displays the list of available events, as well as links to other Event features. See "Events" section for details.

Subscribers – displays the list of available subscriber sets, as well as links to other Subscriber Set features. See "Subscribers" section for details.

Assets – displays the list of available assets, as well as link to other Assets features. Currently, Brand Mobiliser applications are the only asset types supported. System templates can be considered as assets as well as but they are essentially collections of applications. See "Assets" section for details.

Reports – displays the available reporting modules. Currently, there is only one module, Standard Reports. See "Reports" section for details.

Workspace Administration – displays the administration, management & monitoring modules. This administration item is available to users with SUPER ADMIN and ADMIN roles only. See "Workspace Administration" section for details.

On the right hand end of the navigation bar, the "***Navigation Actions***" is a dropdown list of action links, navigating to the setup or frequently used screens, The action links are included:

- Setup Default Menu – See "Default Menu" section

- Manage Categories – See "Manage Categories" section
- Simulate Application – See "Simulate Application" section

2.1.4 Main Screen Layout

The navigation item will display the "Main" screen of the corresponding feature when clicked. The main screen acts as a portal screen to the feature, providing links to other screens related to the feature. The main screen currently has two types of layout: "***List Table***" and "***Module***". For example, the **Assets** screen adopts the table list layout while the **Dashboard** screen adopts the module layout. See "List Table Layout" and "Module Layout" below for details.

2.1.4.1 List Table Layout

"List Table" layout screen is used when the feature has list of items to choose from, such as: Events, Subscribers and Assets. The screen layout consists of three sections: Header, Filters, and List Table. The filters section is optional so it is only displayed when supported.

The Header section displays the feature name (for example, Assets) on the left-aligned side, and action links on the right-aligned side (for example, Create Interactive Application, Create Event Application, etc.) The feature name may be followed by the help icon and when the mouse pointer is hovered over additional information about the feature will be displayed.

The Filters section displays filter attributes, and search keyword that can be used to narrow the displayed items. The filters section has two modes: minimize or advanced. The minimize mode is the default. The advanced mode shows additional filter attributes when the "Advanced" button is clicked, as shown below.

Assets Filters – Minimize Mode

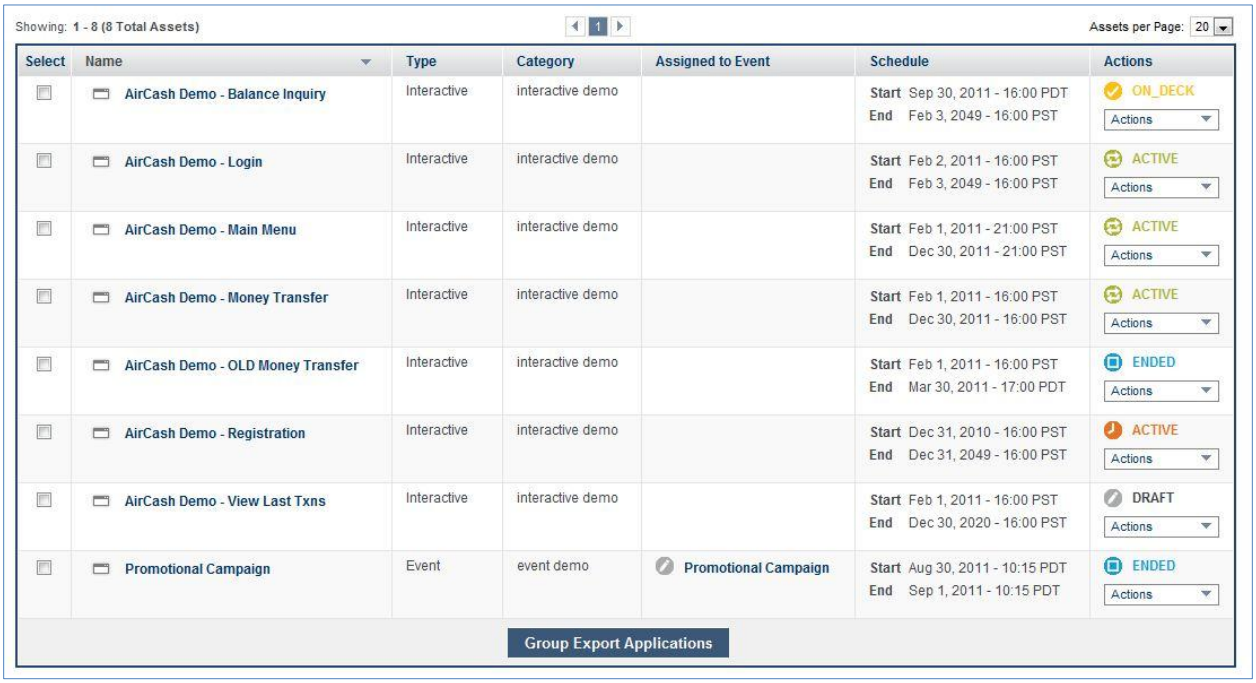
Assets Filters – Advanced Mode

The List Table section displays the list items in table form with the following common features: currently shown items, total items, page navigator, and number of items shown per page, as shown below. If the column header has a downward pointing arrow-head, it indicates that the table is sort able by the column. Sorting is done by toggling between ascending or descending order. The table footer may show one or more action buttons.

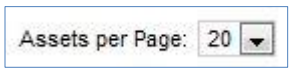
Typically, entries in the table list contains the item's information. For example, the Assets table list shown below displays the list of applications. Entries in the list may have a hyperlink, typically the name. When clicked the screen will navigate to the detail screen. Using the application list example shown below, the Name entry is a hyperlink to the **Application Details** screen.

In some cases like the following example, the "Actions" column (typically the last column) shows the "Status" and "Actions" dropdown menu. The Status may be color coded for better identification, and when the mouse pointer hovers over the status additional information may be displayed. The entries in the Actions menu is context aware

implemented using two mechanisms: the entry can be enabled or disabled based on the entry current status, or an alert popup is displayed confirming or declining the action with reasons.

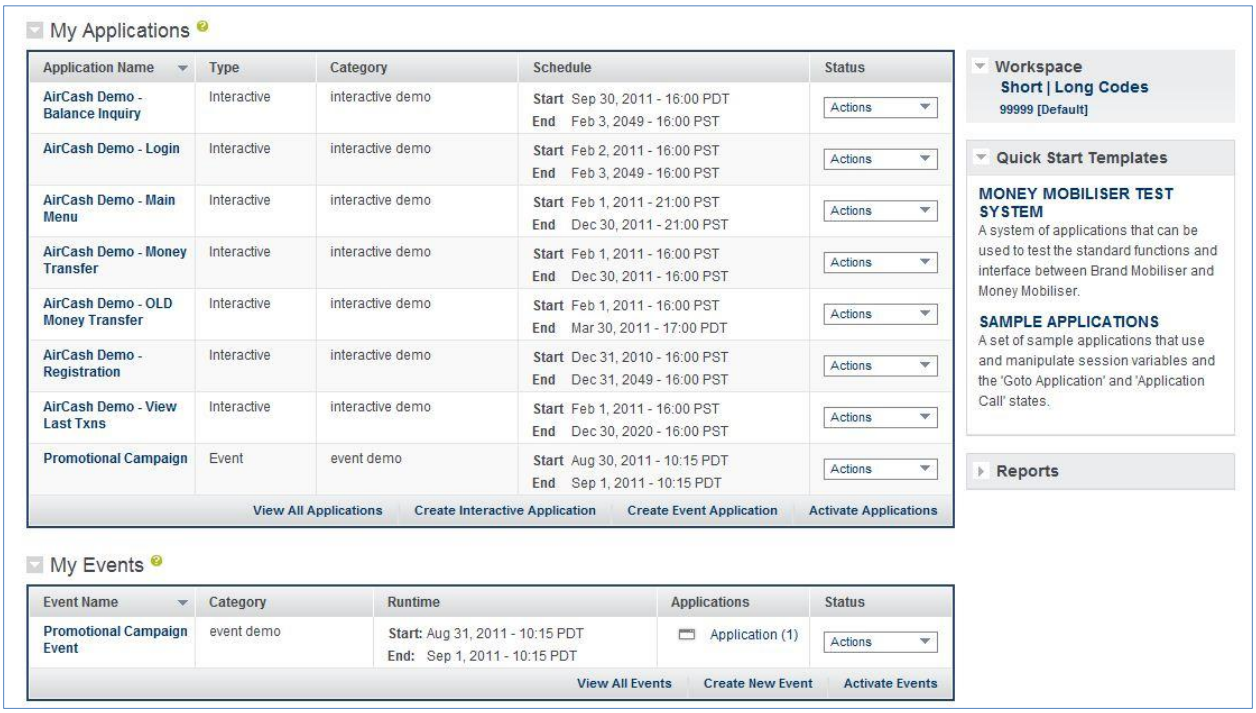


List Table Legend:

	The displayed entries are from 11 to 20 out 229 total programs
	Page navigator; Select the page or click the left-arrow or right-arrow for pagination
	Number of Asset items shown per page; Use the dropdown list to choose a different value. The selected value will be sticky until changed or user logout.
	Header name followed by a downward arrow head indicates sort able column; Sort by clicking on the header name
	Color coded status, and context aware Actions dropdown menu

2.1.4.2 Module Layout

The "Module" layout screen enables future UI extension via modules, and currently used in the **Dashboard**, **Reports, My Account, Create Asset,** and Workspace Administration screens. The current layout is two columns: wide left and narrow right columns, as shown below for the Dashboard screen. Modules that need more width are docked to the left, while narrow modules docked on the right. Modules can be maximized or minimized by clicking button. The Reports module shown on the bottom of right hand column is in minimized mode.



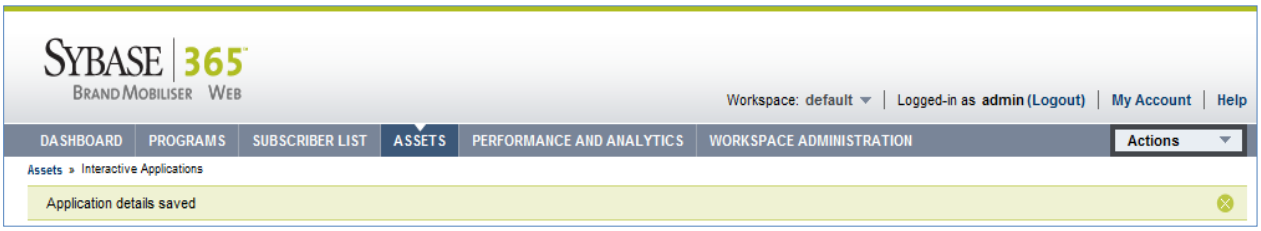
2.1.5 Action Buttons and Icons

Buttons have unique icons with assigned functionality as described below. Icons are also used to show status.

	Save settings
	Add settings
	Delete settings
	Settings are in used (cannot be deleted)
	Feature is activated
	Feature is NOT activated or has expired
	Cancel
Save	Enabled button
Export	Disabled button

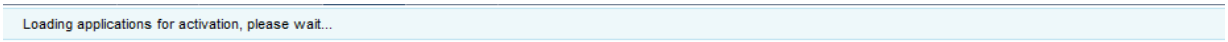
2.1.6 Feedback Panel

Feedback panel is used to inform user the result of the action whether it is success, warning, error, etc. Feedback panel is displayed below the navigation bar and the breadcrumb (if shown). The background color of the feedback panel is adapted to the feedback status. The feedback panel shown below is for a success status.



Note: The feedback panel can be closed by clicking on the icon when available.

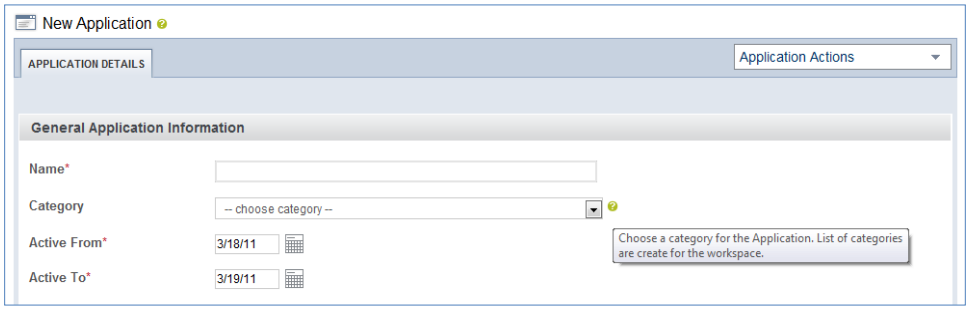
The feedback panel for the in progress status is shown below.



The feedback panel for the error status is shown below.

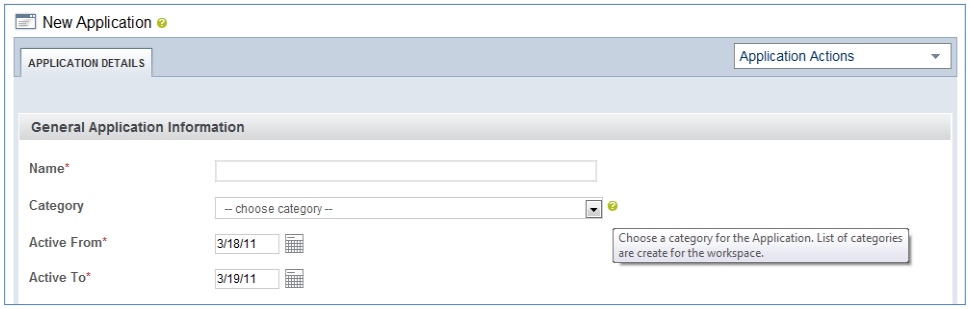
2.1.7 Help Tooltip

Tooltip is provided in some screens as a less intrusive mechanism to provide additional help to the user. The tooltip is typically shown using the icon. To display the additional help information, place the mouse cursor over the tooltip and a moment later a help box should appear.



2.1.8 Mandatory Form Field

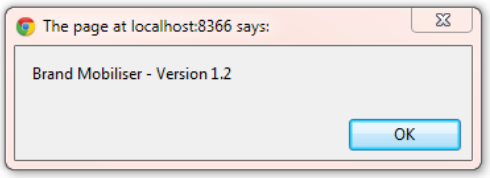
The mandatory form field is designated with a red * following the field name as shown below. The mandatory fields on the following screen are: Name, Active From, and Active To.



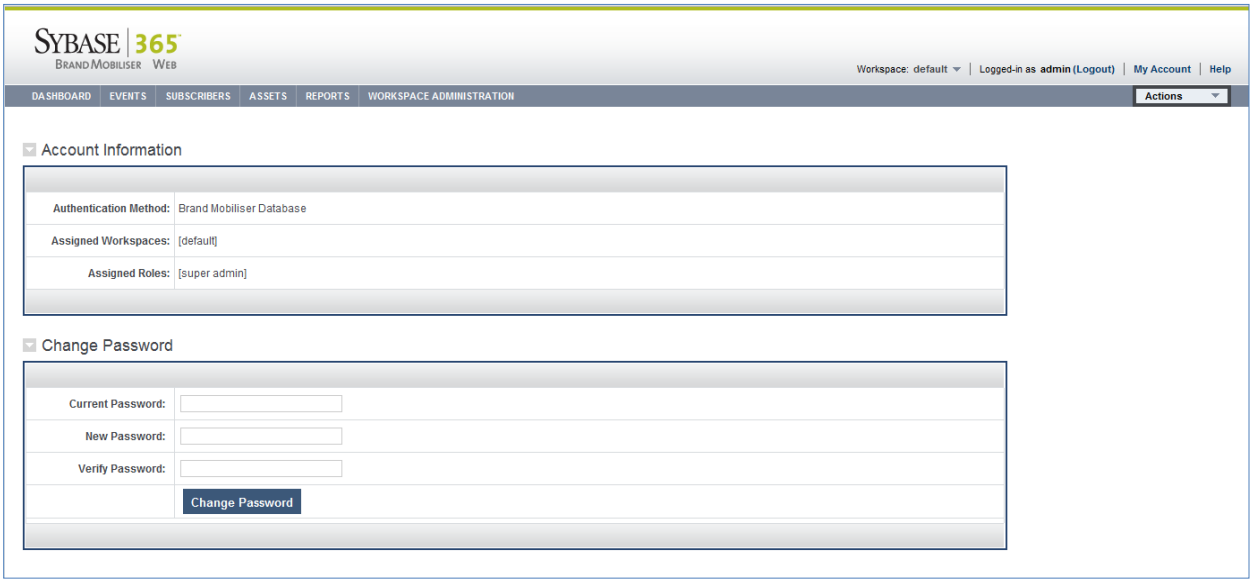
2.2 Login

The Login screen. For security purposes, the only feedback message is “Invalid username/password” regardless of the main reasons. For example, when the username is disabled it will also show the same feedback message.

Clicking the “About Brand Mobiliser” link at the footer to display the Brand Mobiliser version number.



2.3 My Account



My Account screen is reachable from the Login Status Bar – My Account. There are two modules on the screen: Account Information and Change Password. The Account Information module displays information pertaining to the login user such as: assigned workspaces and roles, and the authentication source. If the authentication source is "Brand Mobiliser Database", the "Change Password" module will be displayed allowing the login user to change password. Other supported authentication source is LDAP but it's a platform configuration change discussed in "Installation and Configuration" section.

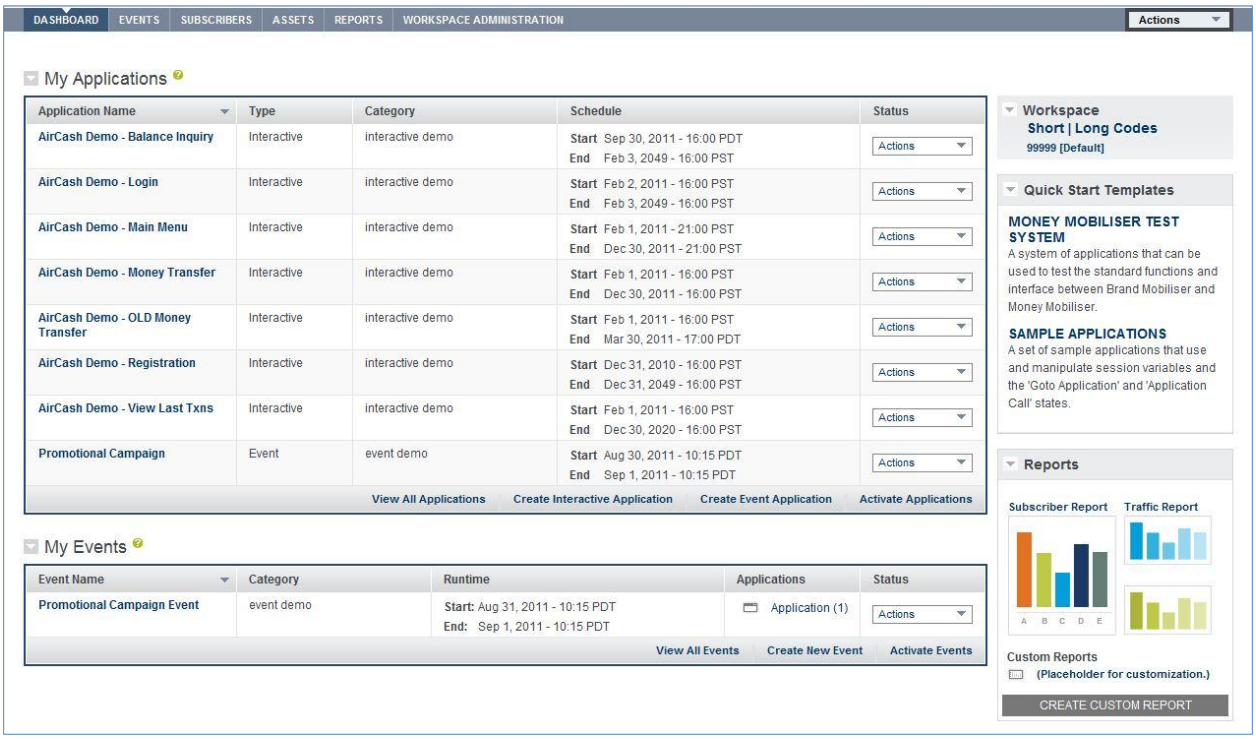
2.4 Dashboard

Dashboard is the first screen that an authenticated user sees and it is the most frequently used screen. As users become more familiar with BrandUI, they will find that the **Dashboard** is a useful and also the "go to" screen when lost in navigation. The **Dashboard** has been carefully designed with strong emphasis on usability. As indicated earlier, the ultimate goal is to make it easy and quick for the user to accomplish their intended tasks.

Dashboard screen uses the "Module" layout. Some modules are displayed conditionally. For example, in future releases, users will see the notification module if there are applications or events that needs their approval.

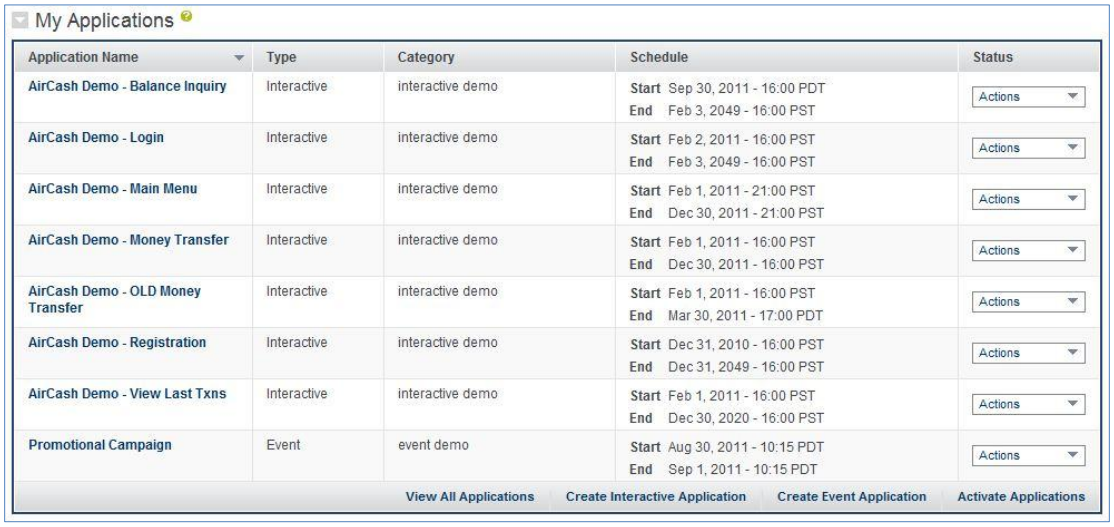
The following modules exist on the **Dashboard** screen. The text inside the square bracket indicates whether the module is displayed on the left or right column.

- My Applications [Left]
- My Events [Left]
- Workspace [Right]
- Quick Start Templates [Right]
- Reports [Right]



2.4.1 My Applications

My Applications module displays a short list of applications belonging to the current user. For a complete list, click **View All Applications** located at the list footer, or click **Assets** on the navigation bar. In either cases, the screen will navigate to the **Assets** screen. By default, the list is sorted by the "*Application Name*" – first column.



Other columns include:

Type – the application type (i.e., interactive or event). Set automatically during create new application.

Category – the application category. Selected from the pre-configured category list by the application developer at creation or modification. The category list is maintained using the **Manage Category** screen.

Schedule – the active from and to dates of the application. Created or managed by the application developer

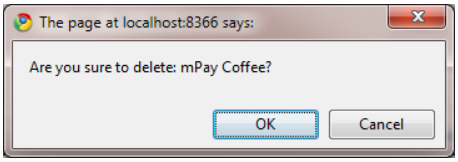
The following functionalities are found within this module.

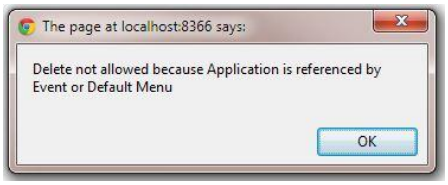
Sort by Application Name – click on the "*Application Name*" in the header to toggle the sorting.

Actions – each row has a context aware **Actions** dropdown menu on the "*Status*" column. The **Actions** dropdown menu has the following options: Show Details and Delete actions.

Actions - *Show Details* navigates to the **Application Details** screen, showing details of the selected application. Please refer to the "Application Details" section for details.

Actions – *Delete* is for deleting the application. A confirmation popup dialog will be displayed showing the application name and the "OK" or "Cancel" buttons. Clicking OK will result in deleting the application, while Cancel will just close the dialog. Upon a successful delete, the list is refreshed with updated entries. When the application is in use (i.e., assigned to Default Menu or Event), a warning popup will be displayed with only the "OK" button, as shown below. Clicking OK will close the popup dialog and cancel the delete.





View All Applications – located at the list footer, navigates to the **Assets** screen. Please refer to the "Assets" section for details.

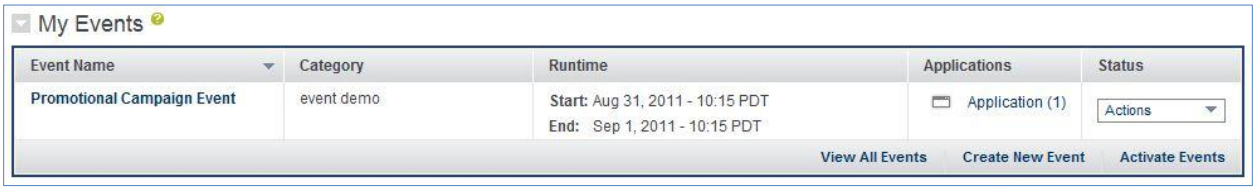
Create Interactive Application – located at the list footer, navigates to the **New Application** screen of Interactive type. Please refer to "Application Creation" section for details.

Create Event Application – located at the list footer, navigates to the **New Application** screen of Event type. Please refer to "Application Creation" section for details.

Activate Applications – located at the list footer, navigates to the **Activate Applications** screen. Please refer to "Activate Applications" section for details.

2.4.2 My Events

My Events module displays a short list of events belonging to the current user. For a complete list, click **View All Events** located at the footer, or click **Events** on the navigation bar. In either cases, the screen will navigate to the **Events** screen. By default, the list is sorted by the "*Event Name*" – first column.



Other columns include:

Category – the event category. Selected from the pre-configured category list by the application developer at creation or modification. The category list is maintained using the **Manage Category** screen. Category list is also used for application category.

Runtime – the runtime from and to dates of the event. Created or managed by the application developer

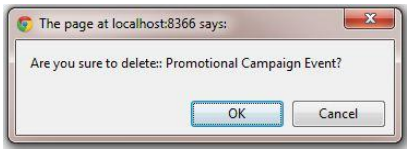
Applications – displays the number of applications assigned to the event. Links to the **Event** screen – Event Application tab to show the application list

The following functionalities are found within this module.

Sort by Event Name – click on the "*Event Name*" in the header to toggle the sorting.

Actions – each row has a context aware **Actions** dropdown menu on the "*Status*" column. The **Actions** dropdown menu has the following options: Show Details and Delete actions.

> **Actions - *Show Details*** navigates to the **Event Details** screen, showing details of the selected event. Please refer to the "Events" section for details.
>
> **Actions – *Delete*** is for deleting the event. A confirmation popup dialog will be displayed showing the event name and the "OK" or "Cancel" buttons. Clicking OK will result in deleting the event, while Cancel will just close the dialog. Upon a successful delete, the list is refreshed with updated entries.



View All Events – located at the list footer, navigates to the **Events** screen. Please refer to the "Events" section for details.

Create New Event – located at the list footer, is the shortcut to the **New Event** screen. Please refer to "Events" section for details.

Activate Events – located at the list footer, is the shortcut to the **Activate Events** screen. Please refer to "Activate Events Screen" section for details.

2.4.3 Workspace Module

The Workspace module displays the short or long codes assigned to the current workspace.

The **[Default]** indicator next to the code indicates that the code is used as default for outbound message in the absent of explicit configuration. For more details discussion, please refer to Workspace Administration – Manage "Workspaces – Workspace Short codes" sections.

If the user has a SUPER ADMIN role, the workspace title becomes a hyperlink to the **Workspace Configurations** screen as explained in details in the "Workspace Administration – Manage Workspaces – Workspace Configurations" section.

2.4.4 Quick Start Templates Module

The Quick Start Templates module provides commonly used templates that can easily be imported into Brand Mobiliser. A template is a collection of one-to-many applications. These applications may be standalone individual applications or linked together forming a system. The template can also be a collection of individual sample applications that are used for demo or tutorial.

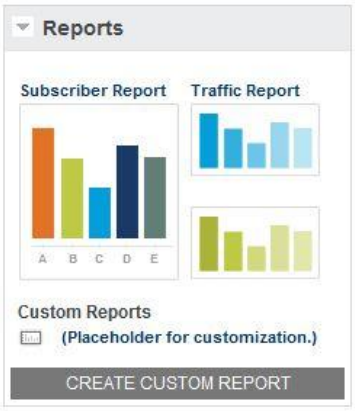
The Quick Start Templates module displays template list as shown below. Templates can be added by the platform administrator and they will show up dynamically. Please refer to "Create Quick Start Template" section on how to create template and add to Brand Mobiliser.

Brand Mobiliser version 1.2 comes with the following built-in templates:

- Money Mobiliser Test System - A system of applications that can be used to test the standard functions and interface between Brand Mobiliser and Money Mobiliser.
- Sample Applications - A set of sample applications that use and manipulate session variables and the 'Goto Application' and 'Application Call' states.

2.4.5 Report Module

The Report module displays shortcuts to commonly used reports: Traffic Report and Subscriber Report. The traffic report shortcut linked to the **Traffic Report** screen discussed in the "Reports" section. The subscriber report shortcut linked to the **Subscribers** screen discussed in the "Subscribers" section.



2.5 Assets

The **Assets** screen lists applications in the current workspace that are authorized to the user. Users assigned to "APPLICATION OWNER" role will be able to see their own applications only. All other roles will be able to see all applications including those created by other users in the current workspace.

Filters can be used to narrow the application list. Advanced filters are displayed by clicking the "Advanced" button. The filters attributes are as follows.

Display – filters by application types, or show both.

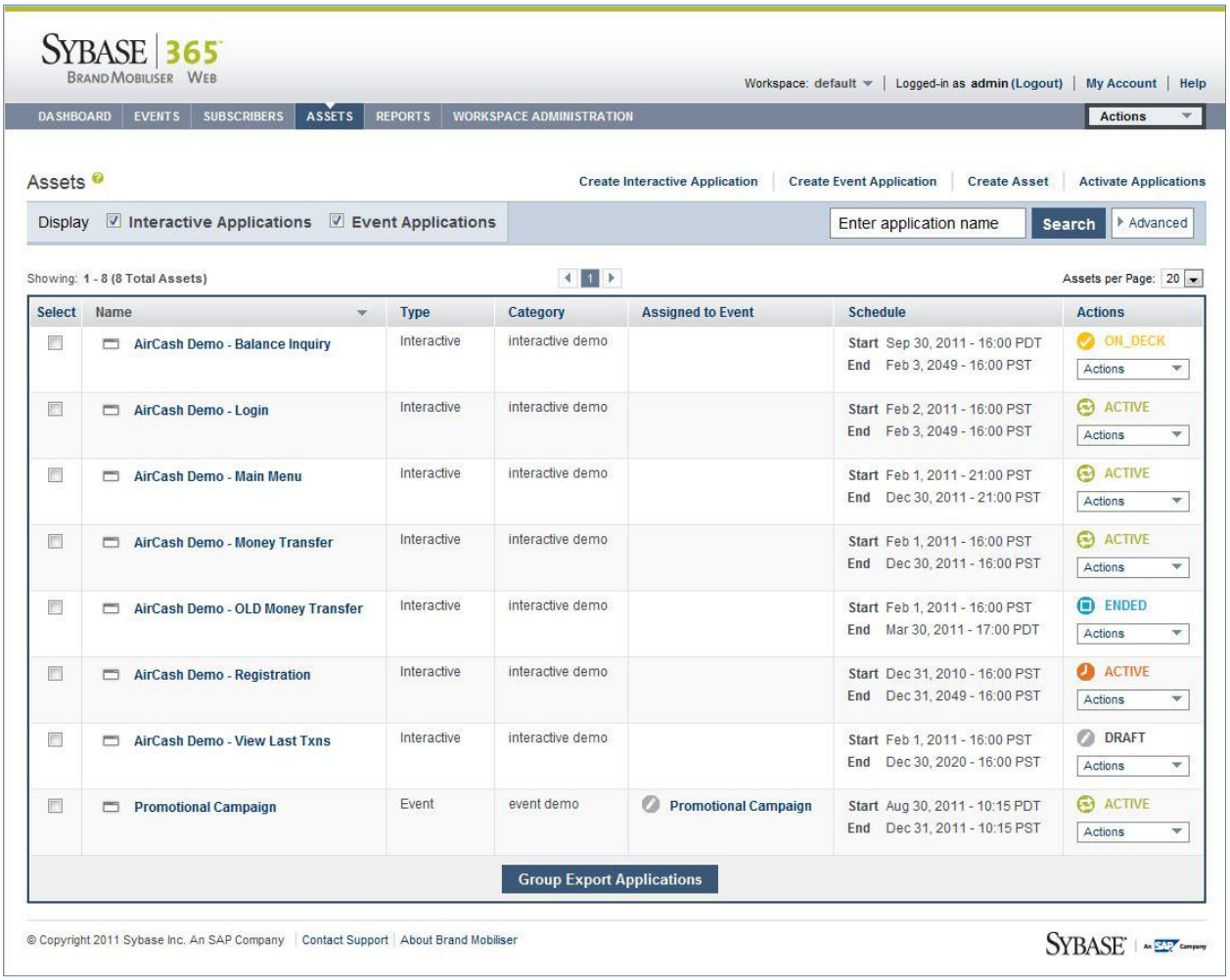
Enter application name – filters or searches by the application name. The character is not case sensitive. The following wildcard characters are supported.

Wildcard	Description
%	Substitute for zero or more characters
_	Substitute for exactly one character

Category – filters by the category.

The **Assets** screen provides access link to other application related functionalities as shown in the screenshot below, such as:

- Create Interactive Application – navigates to the **New Application** screen of interactive type. See "Create New Application" section for details.
- Create Event Application – navigates to the **New Application** screen of event type. See "Create New Application" section for details.
- Create Asset – navigates to **Create Asset** screen that provides the ability to create new application, load Quick Start Template, and upload application(s) from an existing application file.
- Activate Applications – navigate the **Activate Applications** screen. See "Activate Applications" section for details.



The application list is sorted by the Name column (i.e., Application Name) and can be toggle for ascending and descending order by clicking the header. The followings describe each columns in details.

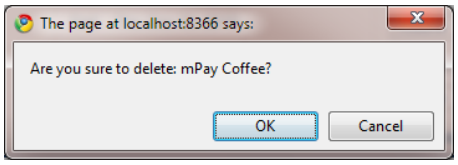
- Select column provides checkbox for selecting application(s). Selected applications can be exported into a single "***Brand Mobiliser Application File***" in xml format, by clicking the "Group Export Applications" button located at the table footer
- The Name column displays the Application Name that is also a hyperlink to the **Application** screen
- Type column displays the Application type: Interactive or Event
- Category column displays the category assigned to this application. The benefit for assigning a category to an application is for filtering of the asset list, as described earlier in this section. New categories can be created using the **Manage Categories** screen, accessible from "Navigation – Actions – Manage Categories"
- Assigned to Event column lists the events that this application is assigned to. Please refer to "Events" section on how to assign an application to an event
- Schedule displays the Start and End date/time of the application
- Actions column displays the application status, and provides the context aware Actions menu, both are described in details below.

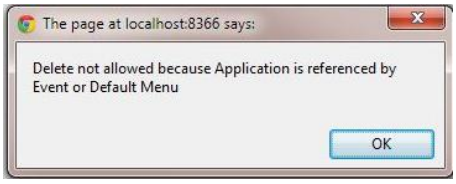
  The application status is shown as text and color, as follows.

DRAFT	corresponds to newly created application, and has not been activated
ON_DECK	corresponds to activated application, and pending start date
ACTIVE	corresponds to Live application
ACTIVE	corresponds to Live application but has been revised pending review and activation
ENDED	corresponds to terminated application because the end date has occurred

  The Actions menu items include:

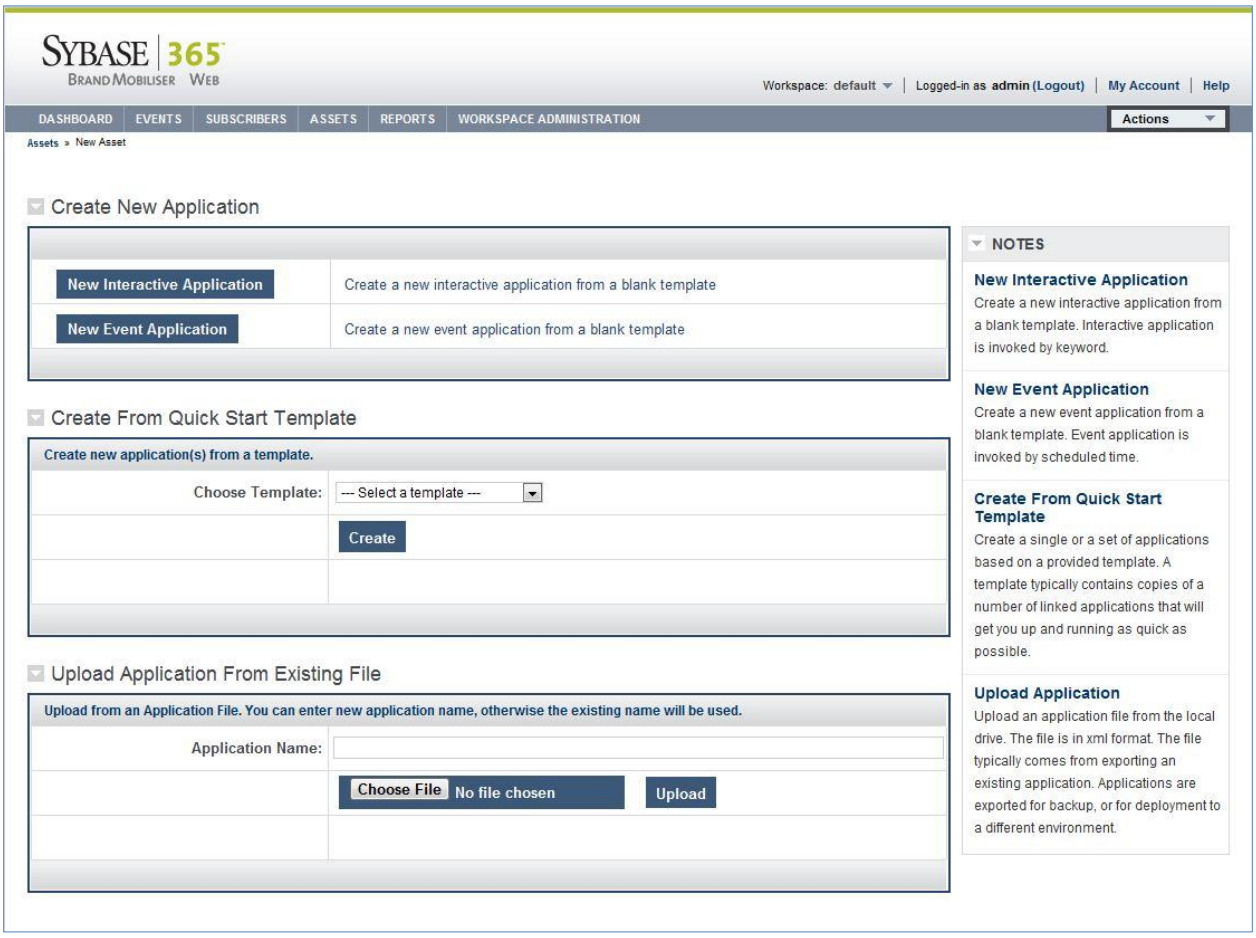
  - **Actions - *Show Details*** – links to the **Application** screen showing the details of the selected application. Please refer to the "Application Details" section for details.
  - **Actions - *Delete*** - deletes the corresponding application. A confirmation dialog showing the application name and the "OK" or "Cancel" buttons. Clicking OK will result in deleting the application, while Cancel will just close the dialog. Upon a successful delete, the list is refreshed with updated entries. When the application is in use (i.e., assigned to Default Menu or Event), a warning popup will be displayed with only the "OK" button, as shown below. Clicking OK will close the popup dialog and cancel the delete.





2.5.1 Create Asset

The Create Asset link navigates to the **New Asset** screen. There are three modules available on this screen including: Create New Application, Create From Quick Start Template, and Upload Application From Existing File, as shown on the following screenshot.



"***Create New Application***" module provides two buttons for creating: "New Interactive Application" and "New Event Application". This is the same as clicking the "Create Interactive Application" and "Create Event Application" links, respectively, on the **Assets** screen. Both navigates to the **New Application** screen with the application type preset. Please refer to "Create New Application" section for details.

"***Create From Quick Start Template***" module lists all the available pre-installed templates. Templates can be added by platform administrator and they will show up dynamically in this list. A template is a collection of one-to-many applications. These applications may be standalone individual application or linked together forming a system. The template can also be a collection of individual sample applications that are used for demo or tutorial. Please refer to "Brand Mobiliser Development Manual" for how to create Quick Start template.

Brand Mobiliser version 1.2 comes with the following templates:

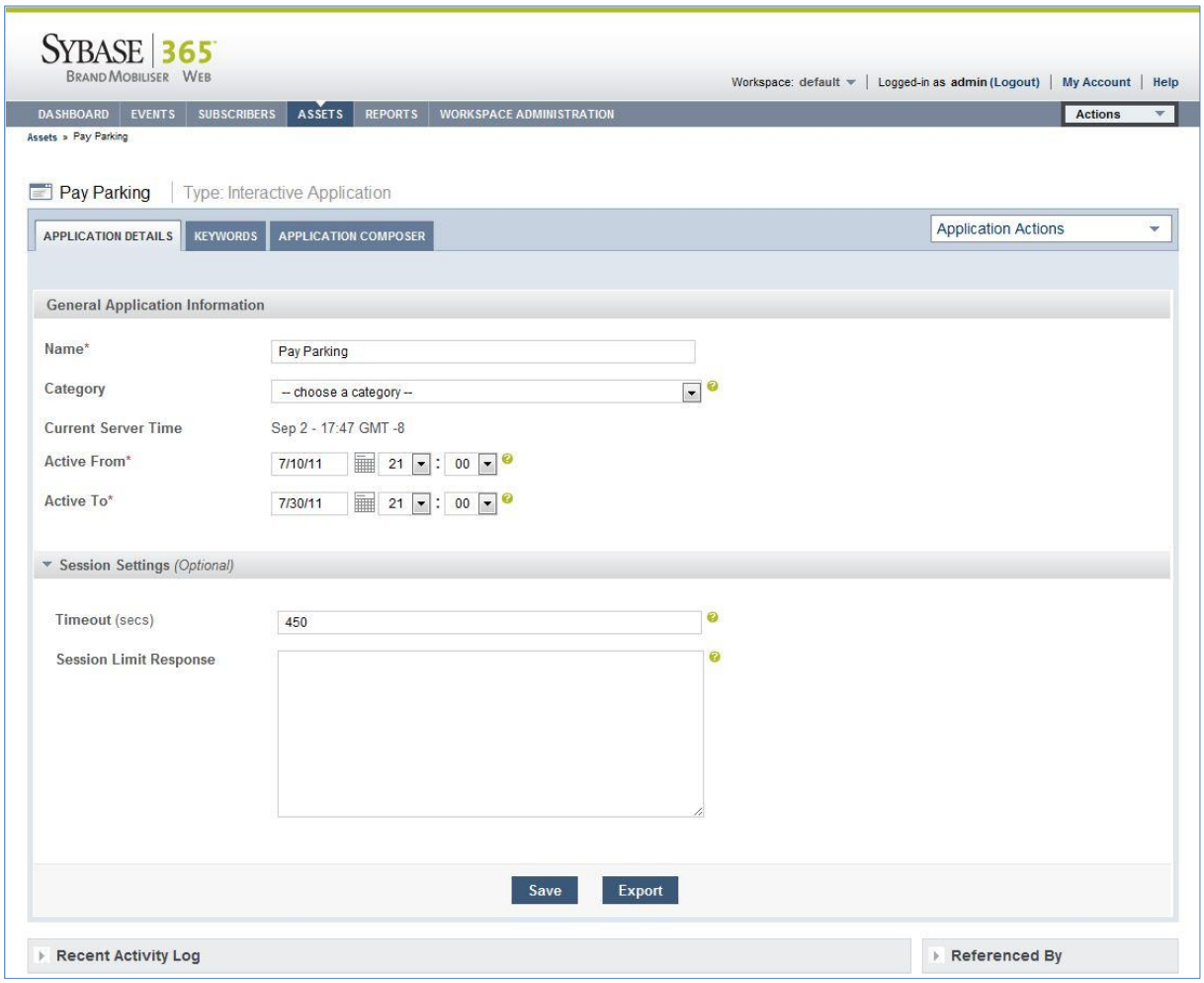
- Money Mobiliser Test System - A system of applications that can be used to test the standard functions and interface between Brand Mobiliser and Money Mobiliser.
- Sample Applications - A set of sample applications that use and manipulate session variables and the 'Goto Application' and 'Application Call' states.

Choose the template from the dropdown list and click “Create” button. Once the template is installed, go to the **Assets** screen to find all the installed applications, and click the application name or use the **Actions** – **Show Details** to view the application details.

“***Upload Application From Existing File***” module provides the “Choose File” component for choosing application file from the local file system. The application file typically comes from a previously saved interactive application for the purpose of backup or export to another environments or systems.

To upload the file, select the “Choose File” button to find and select the file on your computer and click the “Upload” button. The “Choose File” button shown in the screen above was from the Google Chrome browser. If you are using other browser, the button may look slightly different. The existing application name in the file will be used, or enter a new name in the “Application Name” field to overwrite. A confirmation or error message is displayed below the file name field, as shown below.

Clicking on the “View Application Details” button navigates to the **Application** screen with details of the uploaded application, as shown below. Please refer to “Application Details” section for details.



2.6 Application

Brand Mobiliser Application, or Application for short, is a process workflow that is constructed from a sequence of connected states. States are the basic building block with a specific capability. State's capability can be implemented natively within the state ("***stand-alone state"***) or it can be "proxy" to an internal or external web service that is exposed through an SOA layer ("***service state"***). State can also be an aggregation of internal corporate and external services.

Brand Mobiliser provides several basic stand-alone states including: states for composing workflow that are referred to as "***Base State"***, and states for performing operations on the Subscriber storage known as the "***Subscriber State***". In addition, also included out of the box the service states for integrating with the Money Mobiliser platform. Additional custom states can be developed using the State SDK and added dynamically using the plug-in mechanism enabled by the OSGi™ Services Registry. Please consult the reference "Brand Mobiliser – State Developer's Guide" on how to develop and deploy plug-in states for use within the Brand Mobiliser. The custom states enrich the application capability.

Brand Mobiliser provides a rich web-based tool for developing application, called Application Composer. The basic capabilities of the composer that are used to develop a simple application will be discussed below. For more details and advanced features, please refer to the "Brand Mobiliser – Development Manual".

There are two types of application: "***Interactive"*** and "***Event"***, differ by how they are used and invoked. Interactive application is used for providing rich user interaction mobile service. This type of application is typically invoke by mobile customers sending a keyword to a pre-assigned short code. The Event application is designed for workflow or batch processing. The workflow or batch type of application is typically invoked by event, such as: scheduled time(s), system trigger or external trigger events. So, unlike the interactive application, the event application is not associated with keyword(s).

2.6.1 Create New Application

To create a new application, start from the **New Application** screen. For convenience, this screen is accessible from several locations as follows:

- From the **Dashboard** screen – My Applications module – Create Interactive Application or Create Event Application links at the footer of the module
- From the **Assets** screen – Create Interactive Application or Create Event Application links located at the header section on the right hand side.
- From the **Create Asset** screen – Create New Application module – New Interactive Application or New Event Application buttons.

Depending on which you clicked, the **New Application** screen is customized to the application type as shown in the title.

2.6.2 Application Details

The new application screen starts with a single "Application Details" tab as shown below, containing the basic information needed to create a new application. Once the mandatory fields are provided and saved, additional tabs will be displayed. The screen below is for creating a new interactive application as shown by the type.

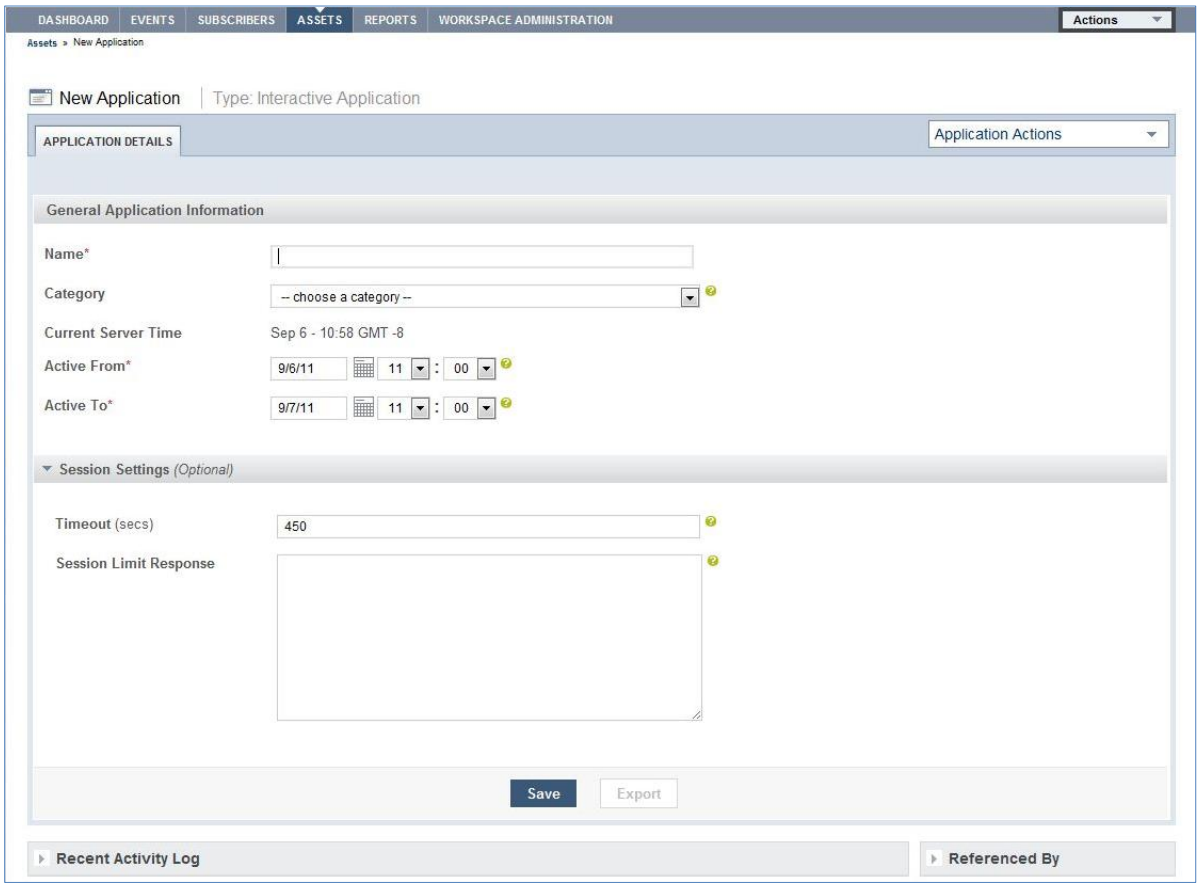
The fields, action links and buttons on the Application Details tab includes:

Mandatory fields:

- Name – name of the application. The name is the main identifier for referring to the application. Duplicate name within the workspace is allowed but not recommended. No validation or warning is provided for

occurrence of duplicate name within the workspace. The application developer is responsible for duplicate checking prior to assigning the name.

- Active From – the start date of service and is based on the server date and time. Note: the **Current Server Time** is shown to aid in establishing the correct date and time. Activated application will not be in service until the start date.
- Active To – the end date of service and is based on the server date and time.



Optional Field that appears on both application types (Interactive and Event):

- Category – information only; used for easily grouping applications in the **Assets** screen. New category may be added using the **Manage Categories** screen, accessible from either Actions – Manage Categories or from Application Actions – Manage Categories links. The dropdown lists provides the available categories. Please refer to "Manage Categories" section for details.

Optional Fields that appear on the Interactive Application type only:

- Timeout – in seconds. An interactive application establishes conversation with the mobile subscribers. When the conversation starts, a unique session is established for the conversation. By default, this session is terminated (or timeout) when there is no conversation for more than 7:30 minutes or 450 seconds. This timeout field is used to adjust the timeout.
- Session Limit Response – the default message sent to the mobile subscriber when the application is not able to start or carry-on a conversation for various reasons. Typically, the reason is because there are too many conversations already taking place beyond the capacity of the system. In this case the default message provided in this field is sent to the mobile subscriber immediately. For example, the message can say: "Sorry, the service is busy, please try again in few minutes".

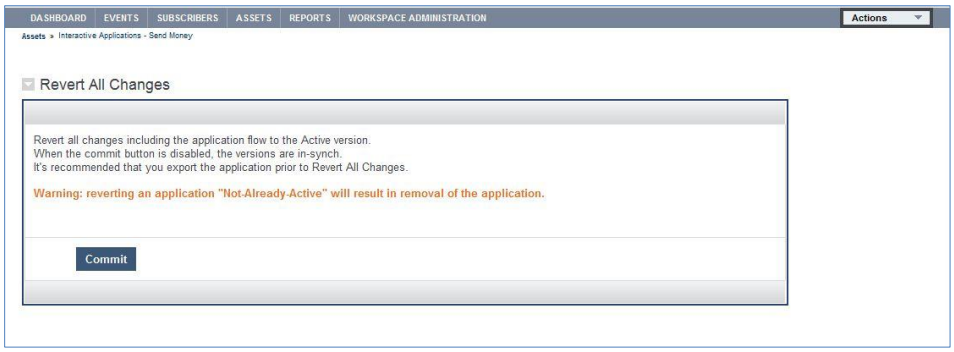
Buttons:

- Save – save the application. Once the application is saved successfully, the Application Composer tab will be displayed. The Keywords tab will also be displayed for interactive application. See below for details on each tab.
- Export - exports the application in a portable XML format to the user's local file system. The XML format is transferrable to other Brand Mobiliser workspaces or installations. You can also use the feature to safely backup an application in a portable way. Please refer to "Development and Deployment Model" section for a complete discussion on establishing the best practice development and deployment environment.

  Note: The Export button is disabled prior to saving a new application.

Application Actions is a dropdown list that provides commonly used links (shortcuts) while working on an application. The content of the list is universal across all three tabs: Application Details, Keywords, and Application Composer.

- Activate Application – once an application creation is complete, it needs to be activated before it become "***Live***" pending start date. The same applies when modification is made to application information. For further discussions on Activation, see the "Activation" section. Please also refer to "Activate Applications" section on how to activate applications.
- Simulate Application – once an application is activated, it starts to be "In-Service" when the current date is equal to the start date. The in-service application can be tested using the application simulator, or directly using the mobile handset if the channel has been setup as well. The application simulator is available on the **Simulate Application** screen. It enables testing without the need to have a real channel or mobile handset. This is very useful in the development and/or QA environment. Please refer to "Simulate Application" section for details on how to test using the **Simulation** screen.
- Manage Categories – adds new or modifies categories. Please refer to "Manage Categories" section for details.
- Revert All Changes – navigates to the **Revert All Changes** screen. The revert action can be viewed as the opposite to the activate action, whereas the Live version is copied back to the In-Review version, overwriting all the changes made in the In-Review version.



An example when the revert action is needed is as follow. After making changes to the In-Review version using the Application Composer, the changes do not work as expected or the changes are not required any longer. Instead of leaving the changes in the In-Review version, causing potential confusions or activation of the unintended version, the revert action can be used to synchronize the Live version back to the In-Review version.

WARNING: reverting a newly created application will result in complete deletion of the application. Therefore, it is highly recommended as a best practice that the application is exported to the local file system prior to the revert action. In this case, the application can be re-import back.

Recent Activity Log Module – Not available in this version.

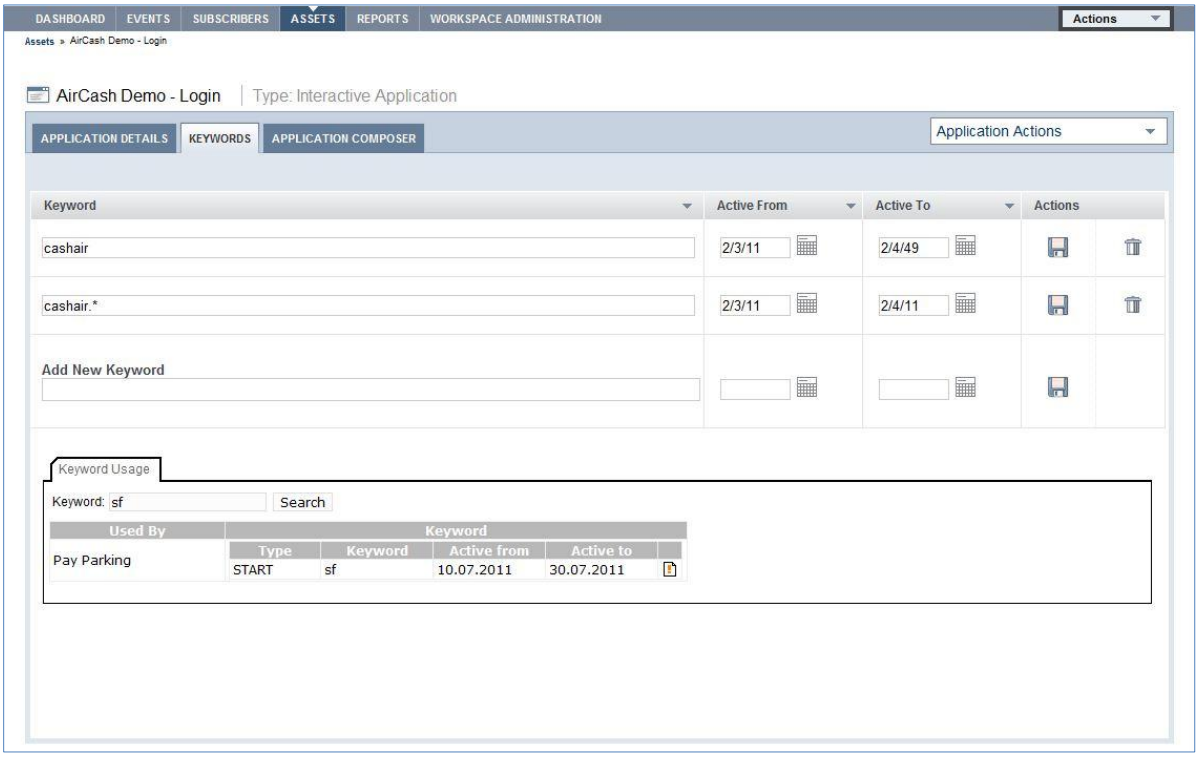
Related Items Module – Not available in this version.

2.6.3 Keywords

Keywords tab appears for Interactive applications only, since interactive applications are invoked by keyword. When an incoming message arrives, the engine extracts the destination MSISDN and keyword. The destination MSISDN is matched to the short/long code of the workspace, and the keyword is matched to the applications within that workspace. The keywords tab is where the keyword(s) for the application is specified. Keyword is the mechanism to share a short/long code. Keywords can be a simple string like "coupon" or regular expressions. The *use of highly complex regular expressions is not recommended*. Please refer to Appendix Regular Expression for discussions on regular expressions. There is a date range associated with the keyword so you can control how long the keyword will remain active independent of the Application. Keywords should be uniquely assigned to one application, but an application can have more than one keywords.

The dates (Active From and To) are optional and can be left empty or null. When left empty, the Application date range as specified in the Application Details tab will be the governing dates. The keyword dates, when specified, will override the active from and to dates. Consequently, **it is possible to have an expired Application Details date range, but a non-expired Keyword date range. In this case, the application will be active under the keywords with a non-expired date range only.**

Even though keyword should be unique for each application within the workspace, BrandUI does not check or prevent a keyword to be assigned to more than one applications. This is due to the fact that keyword can also take regular expressions so it is difficult to detect duplicates. A simple tool called "**Keyword Usage**" is provided on the same screen to enable the application developer a quick search to see if the keyword has been used by other applications within the same workspace. This is not a duplicate-proof tool though, as it cannot test against the regular expressions. To use the tool, enter a keyword in the field and click Search. List of application using the entered keyword will be displayed, as shown below. The keyword "sf" is used an application called Pay Parking, and it has the Active From and To date of between July 10 and 30, 2011, hence the application has expired.



Note: the application developer is responsible to ensure that no application keyword overlapped within the same workspace. When overlapped keyword occurs within the same workspace, there is no mechanism to ensure a specific application is triggered.

The followings describe fields, action buttons on the Keywords tab.

Fields:

- Keyword column lists the assigned keywords for this application. Keyword field is mandatory.
- Active From displays the active from date of the keyword. Optional field.
- Active To displays the active to date of the keyword. Optional field.

"Add New Keyword" row can be used to add new keyword, followed by clicking the save button  .

To modify an existing keyword, place the cursor at the keyword field and start modification followed by clicking the save button when done.

Changes made on the keyword tab require re-activation before the changes become Live.

Action buttons and Statuses:

	Save settings
	Delete settings
Search	Search keyword usage
	Application is activated
	Application is NOT activated or has expired

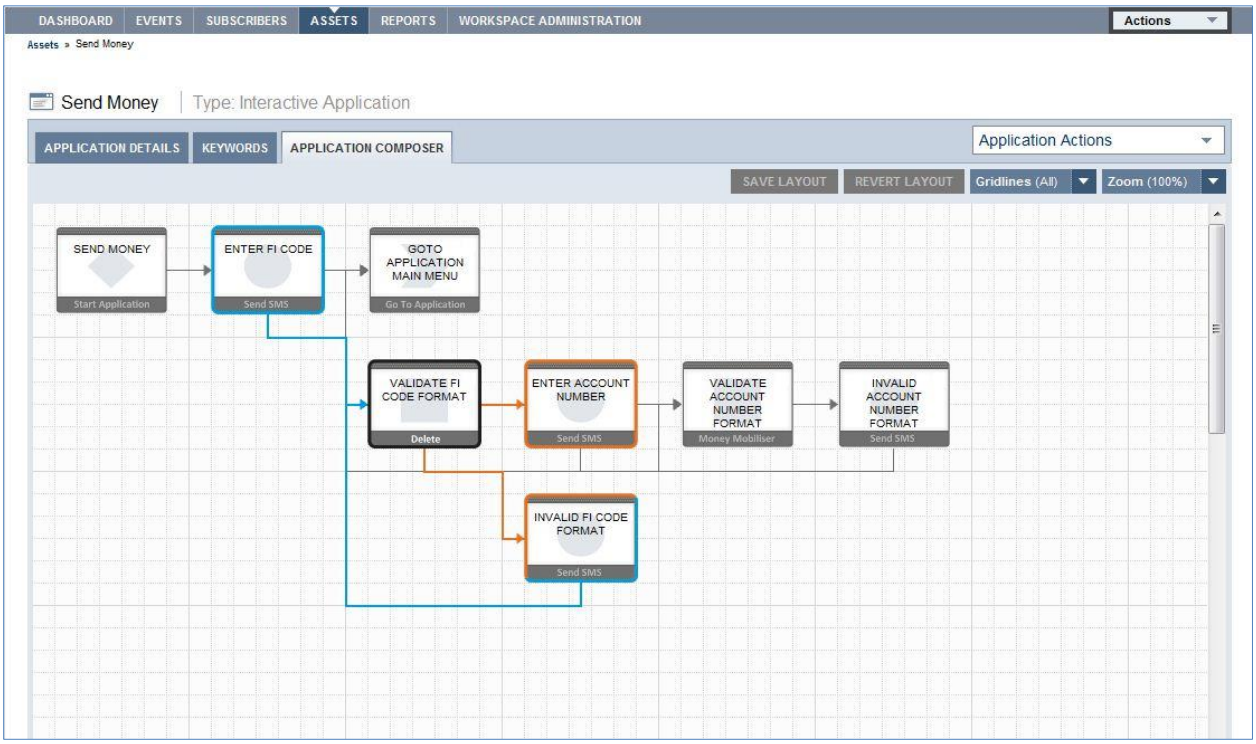
2.6.4 Application Composer

The basis of Application Composer is to enable our customers to visually compose Brand Mobiliser application and configure the states. The key to effective development of applications is the easy-to-use visualization of the workflows involved in modeling the business processes. This is primary purpose of the web-based visualization interface.

The Application Composer state layout view is the primary way of visualising the processing steps of the application workflow, by visualising the states and drawing the transitions between these states. Much as a generalised visual workflow tool would do, the Application Composer allows the application developer the ability to;

- visualise states in the application using an automatic layout,
- dragging and dropping of states to re-arrange layout and to save the modified layout,
- highlight the context and dependencies, the transitions of states, and
- zooming in and out of the application composer view to see a complete or partial application layout.
- Changing to the preferred gridline type

The Application Composer also provides complete facilities to traverse through the workflow through a "***State Property Editor***" popup window or "***State Editor***" for short. The purpose of state editor popup is to allow configuration of the state's settings and values. This is described separately in the next section.



We will only introduce features of Application Composer in this document. For in-depth discussions, please refer to the "Brand Mobiliser – Development Manual".

Application Composer features include:

Layout Canvas shows the application flow on the gridline background. The flow is from left-to-right. The flow consists of states (icon shown as box) and transitions connecting two states (shown as line). The state icon shows the name of this state instance on top of the icon, the type of the state in the bottom bar of the icon and a watermark pattern to also help signify the type of the state. The transition between one state and the next is shown as a directional arrow between the related states. For example using the above screen, the transition from the "Send Money" state to the "Enter FI Code" state is shown as the directional grey arrow with the arrow head on the "Enter FI Code" side. In more complex applications, transition lines may overlap others. For example, the transition between "Validate FI Code Format" to "Invalid FI Code Format", shown as orange line, overlaps with the grey transitional lines.

Context and Dependency Highlighting is essential for more complex flow. Any state can be 'highlighted'. Highlighting a state allows all of its transition lines that "***go to it***" and "***go from it***" to also be highlighted, along with the states they come from or go to.

To select a state, move the mouse cursor over the state icon and left-click. The dependent states and transition lines are then highlighted in different colors:

- The highlighted state is shown with a dark grey surround. For example, the "*Validate FI Code Format*" state on the above screen. Also notice that when a state is a selected state, the state type icon is replaces with the delete icon. Please refer to "State Property Editor" discussion below.
- Any states that transition TO the highlighted state have a blue surround, with the transition line emboldened in blue. For example, the "Enter FI Code" state on the above screen.
- Any states that are transitioned FROM the highlighted state have an orange surround, with the transition line emboldened in orange. For example, the "Enter Account Number" state on the above screen.

- Any states that have both transition TO and transition FROM the highlighted state have both blue and orange surrounds (dual mode). For example, the “Invalid FI Code Format” state in the above example.
- The popup State Editor window will open automatically. The State Editor will be discussed in details in the “State Property Editor” section.

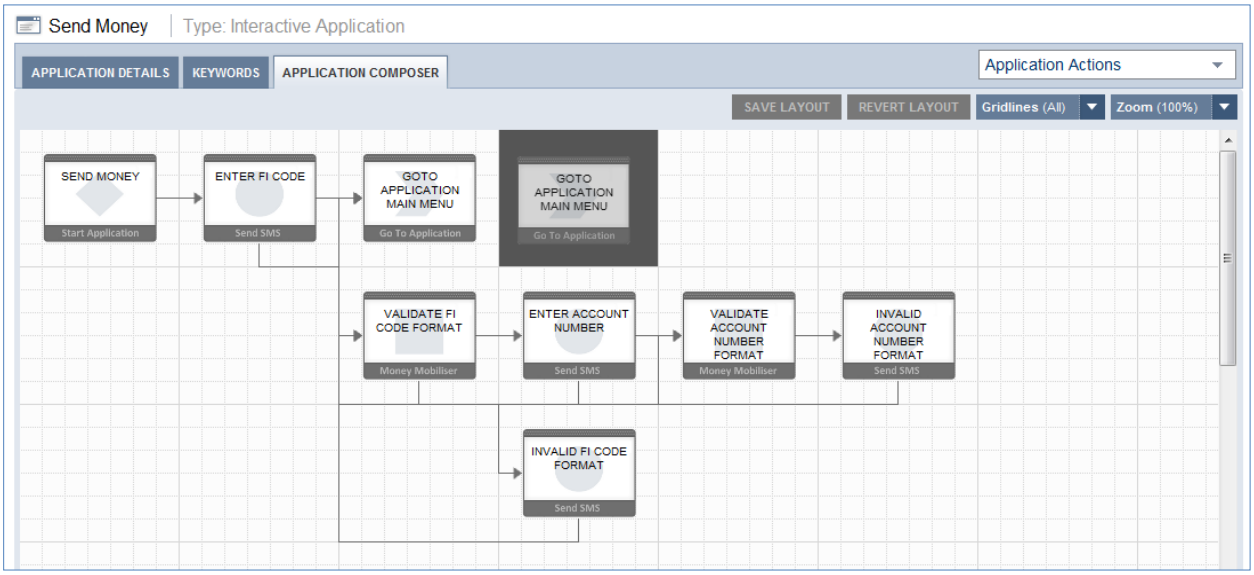
Note: the state icon is a toggle button. When you click a selected state again, the state will be de-selected and all the highlighting will disappear.

Re-arrange Layout can be accomplished by “click-and-drag” the state icon. On first entry to the Application Composer view any states that have been added to the application are laid out automatically on the canvas. Occasionally, it is worth re-arrange the layout of the states, to get a better view of the transitions between the states, particularly when transition lines are overlapping. The state icons on the canvas are sensitive to drag-and-drop into fixed grid positions on the canvas.

To start a drag;

- move the mouse over the state icon you want to move,
- left-click on the mouse and hold the click down, and
- drag to an alternative grid position

The figure below shows that when being dragged, the state icon will become transparent and target grid positions will be highlighted when the mouse enters that grid area.

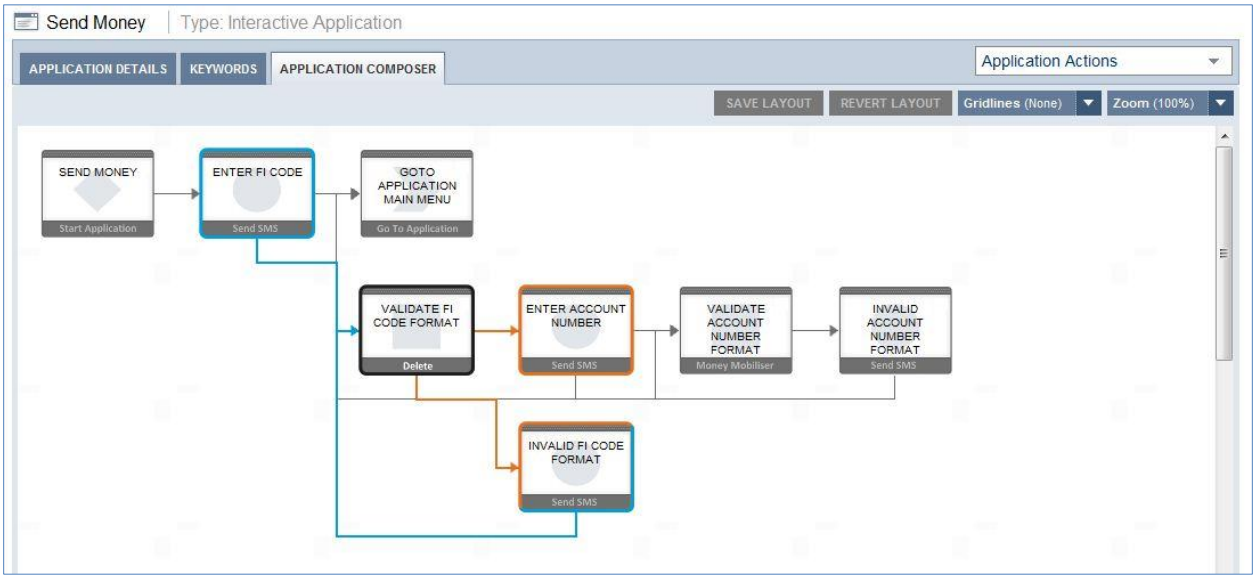


Note: The canvas will not allow absolute free-form positions and each state will be snapped-to a specific grid position as highlighted when being dragged. The transition lines are always automatically positioned and there is no way to move a transition line itself.

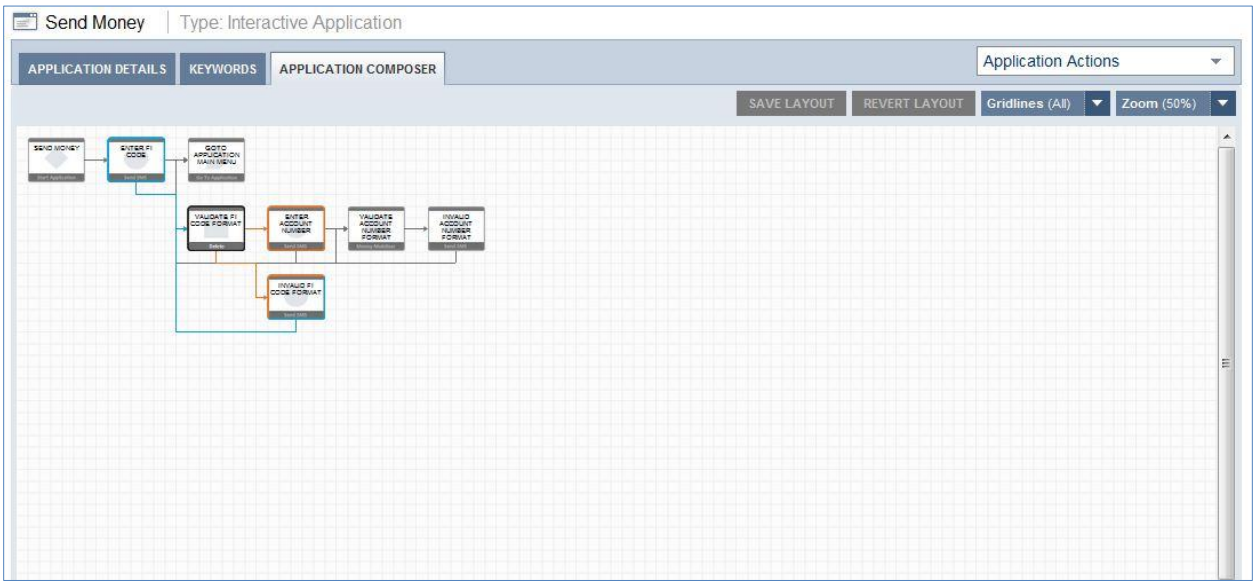
SAVE LAYOUT – the re-arranged layout can be saved for future display, by clicking on this button which saves the layout to the database. This button is enabled only when the layout has been modified.

REVERT LAYOUT – reverts the application layout to the last saved layout from the database. This button is enabled only when the layout has been modified.

Gridlines – Allows the option to display all the grid-lines, a partial grid line or none at all. The none gridline is shown below.



Zoom – Allows the option to zoom the Application Composer view panel both out and in so that applications with a large number of states can show the complete workflow on one screen (see example below). **Note: If you have zoomed out from the default 100% view, you must reset the zoom level back to 100% before making any layout changes.**

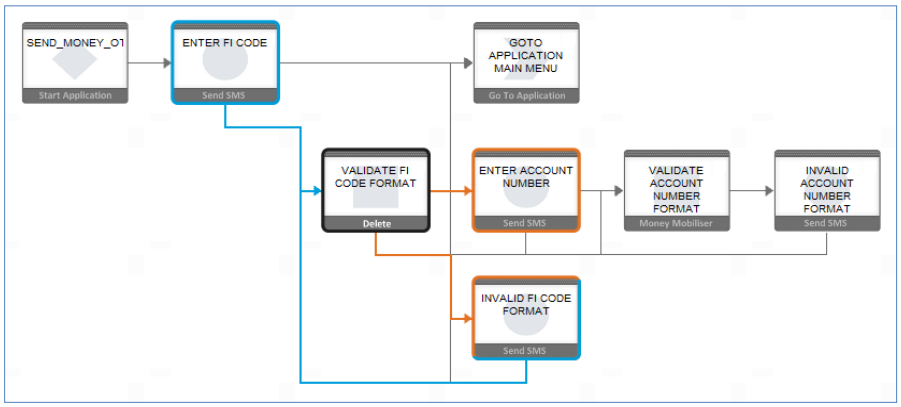


2.6.4.1 State Property Editor

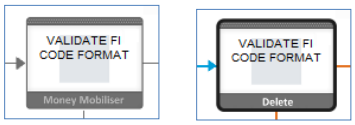
The Application Composer also provides facilities to manage the states displayed on the layout canvas including: select the state, delete the state and configure the state using the State Property Editor popup or State Editor for short. Please refer to the "Brand Mobiliser Development Manual" for more detailed information on how to use the state editor.

SELECT State When the application layout is displayed for the first time, there will be no state selected. To select a state, move the mouse over the state icon and left-click. Three things will happen after clicked:

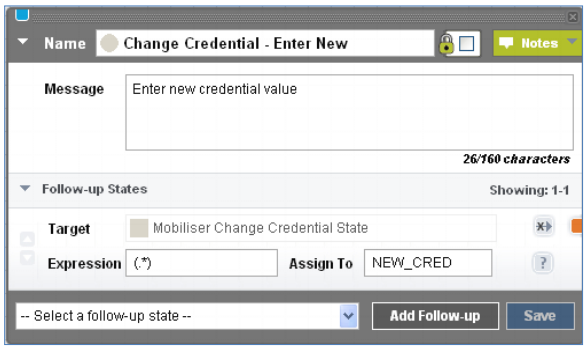
- The state will be selected and highlighted as explained above. The "Validate PI Code Format" is the selected state, shown below.



- The bottom portion of the state icon on the layout will change to display the “Delete” icon. The following figure shows the non-selected and selected state on the left and right, respectively.

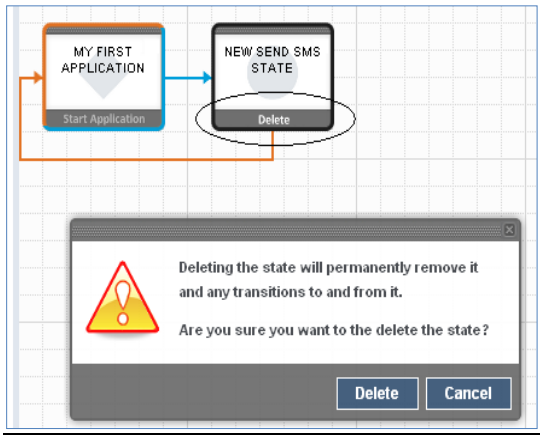


- The popup State Editor window will open automatically showing the state configurations. The following figure shows the state editor for Send SMS state.

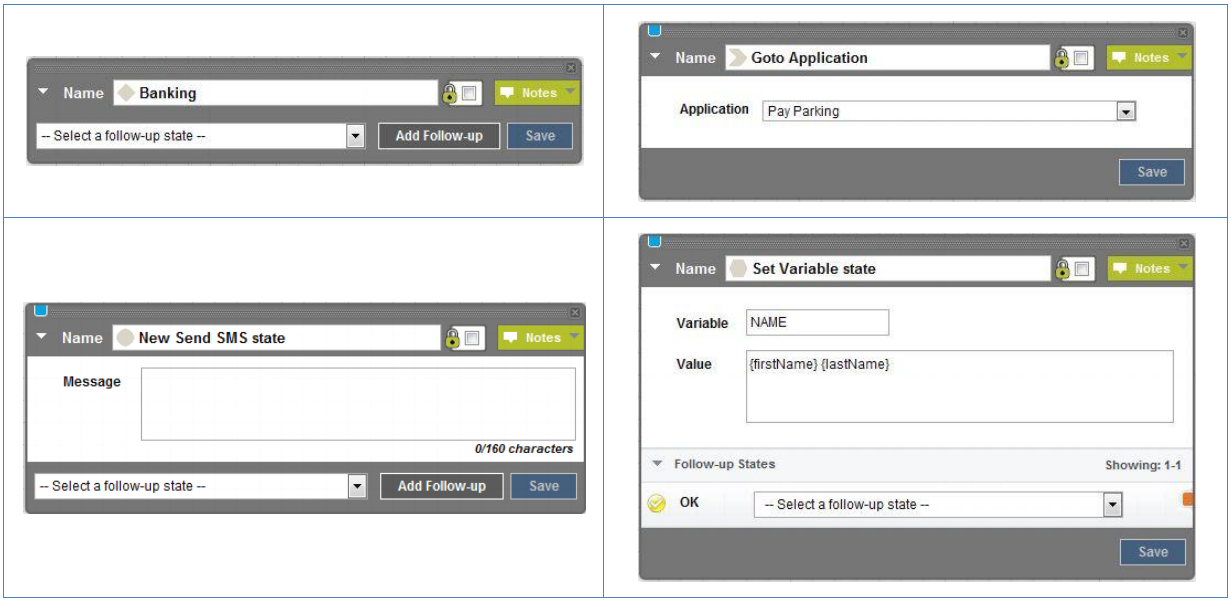


The state icon on the layout is a toggle button. When clicked again, the state will be de-selected and all the highlighting will disappear. The popup state editor window will remain open because you can move to other states connected to or from the current states.

DELETE State can be accomplished by clicking on the “Delete” icon of the selected state. Deleting a state will permanently delete that state and any transitions that were associated with it. You are prompted to ensure you know the implications of deleting the state, and can continue or cancel the delete action.



CONFIGURE State is done using the State Editor. The following figures show the state editor for a sample of some base states: Start Application, Goto Application, Send SMS, and Set Variable.



2.6.5 *Interactive versus Event Applications*

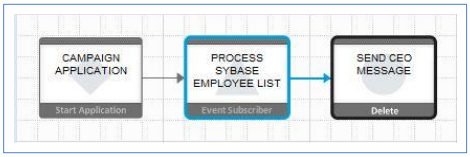
In general, interactive and event applications are the same. Both are modeled using the Application Composer. Listing the differences may help explain better the detailed features of each types.

- Interactive application is invoked by keyword; Event application is invoked by event; Therefore, there is a Keyword tab on the interactive Application screen but not for event application.
- When invoked, interactive application creates a session based on the mobile subscriber MSISDN (or phone number). The session context is persisted when the application reached the "***wait-for-response***" state (like Send SMS state), and the session context expiration will be managed automatically by the BrandEngine. The session context will be terminated when the session expiration time is reached. The session context enables "***interaction***" with the consumer (such as, mobile handset). By contrast, event applications create a session based on the system unique ID but the session context is terminated when the *wait-for-response* state is reached. So the event application is not able to interact with the consumer.
- Some states are available only to an application type. The "Process Subscriber" state is available to the event application only. The "Application Call" and "Application Call Return" states are available to the interactive application only.

- The “Goto Application” state does not display on the Application dropdown list the event application(s) in the workspace because routing to event application is not supported due to the difference in session context described above. The “Goto Application” state can be routed to an interactive application, but in this release there are limitations when used in the event application that pretty much invalidate the benefits of using the “Goto Application” state in the event application. The limitations are as follows. The “Goto Application” state cannot be used in conjunction with the “Process Subscriber” state because it prevents the “Process Subscriber” state from fetching the next subscriber resulting in processing one subscriber and then stop. Secondly, the “Goto” interactive application does not support *interaction* with the consumer because again of the difference in session context mechanism. Hence, the “Goto” interactive application flow needs to end when the *wait-for-response* state is reached because the rest of the flow is not reachable. In addition, when there is at least one state connecting to the *wait-for-response* state, the session context will be persisted while waiting for the response or until session expiration. This session persistence adds un-necessary workload to BrandEngine.
- Event application is invoked by event (time-based, system, etc.) but interactive application cannot. However, the work around with some limitations is to have an event application with a “Goto Application” state routing to the interactive application. Please refer to the discussions on above bullet item, regarding the limitation of the “Goto Application” when used in the event application before using this approach.
- One of the important feature of the event application is its built-in callback mechanism. New states can be developed to take advantage of this callback for an automatic application loop-back. The “Process Subscriber” state is an example state that uses this callback mechanism. Please refer to the “Process Subscriber State” section below for details.

2.6.6 Process Subscriber State

The Process Subscriber state is only available to the event application because it relies on the event application “***callback***” mechanism for handling an automatic application loop-back. The Process Subscriber state was added to the Brand Mobiliser for the ability to retrieve subscribers, one at a time, and feed it into downstream application flow. Let’s take look at the following event application flow, called Campaign Application.

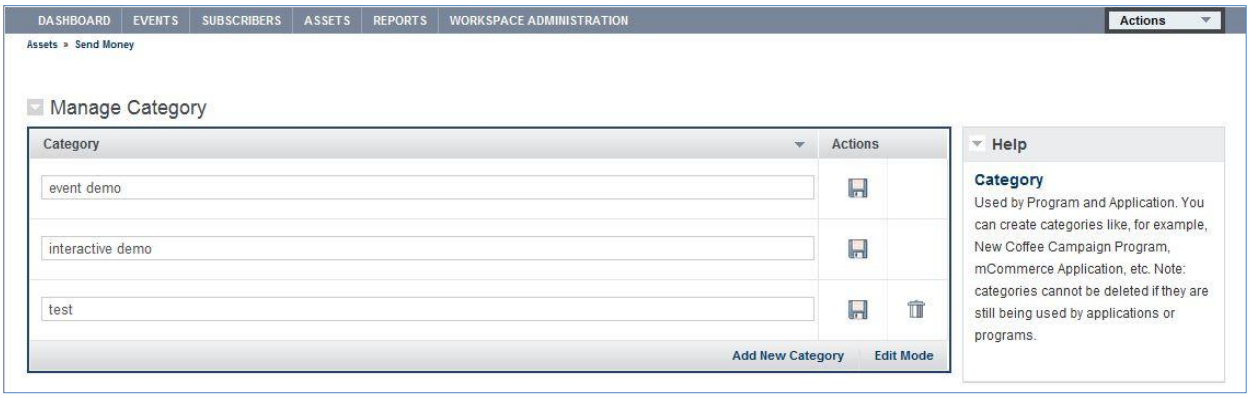


This is a very basic event application but it is used quite often to perform SMS campaign, such as broadcasting the CEO message to employees. This application, although looks very simple, actually performs quite a lot and it could execute for a long time. The application enters from the “Start Application” state (called Campaign Application) and moves to the “Process Subscriber” state. The state is named “Process Sybase Employee List” because it will retrieve subscribers one at a time from the Brand Mobiliser “Subscribers”, and feed to the “Send CEO Message” state which is really a renamed of the “Send SMS” state containing the CEO message. When the message is successfully sent, a loop-back to the “Start Application” will be performed automatically by the processing engine on behalf of the event application. The loop-back enables the “Process Subscriber” state to retrieve the next subscriber and again feed it to the “Send CEO Message” state downstream. The stopping criteria is determined by the “Process Subscriber” state based on, for example, the Sybase employee list has been exhausted. There are other control parameters, available via the State SDK, that can be used to determine the stopping criteria or other control functions. There are three control parameters: Limit, Throttle and Resume, that can be configured from the **Event** screen - Event Windows tab. Please refer to “Event Windows” section for discussion on these three control parameters.

2.7 Manage Categories

Manage Categories screen is used to add new categories or manage existing categories, as shown below. Category can be assigned to Application and Event. Currently, it is mainly used for filtering in the **Assets** and **Events** screens.

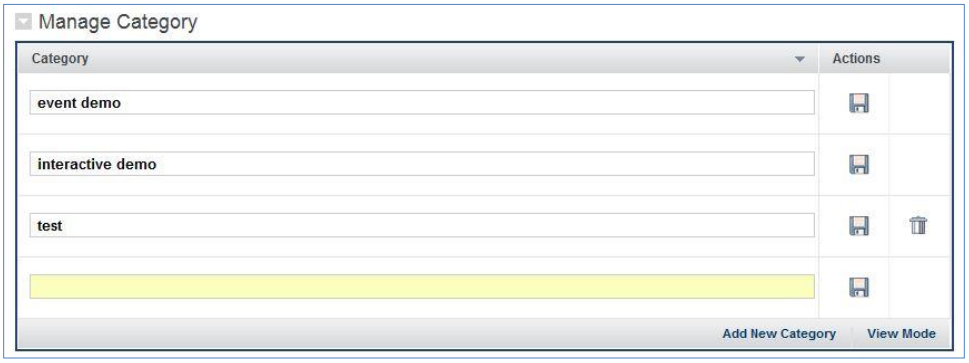
The screen has two modes: View or Edit modes. The screen below is in View mode showing a protected category textbox, and an "Edit Mode" action link at the footer.



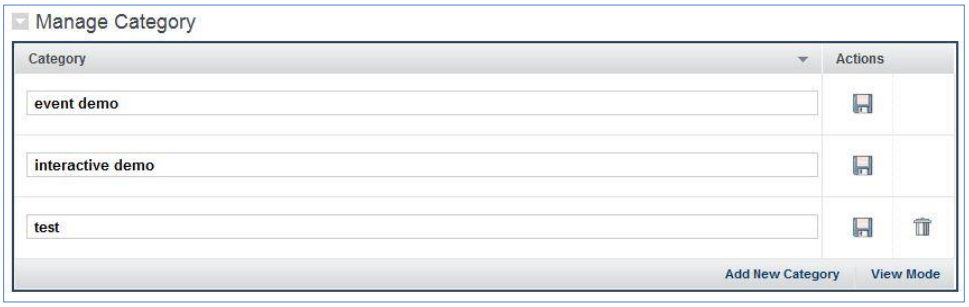
Category displays available categories. The textbox is protected while in View mode.

Actions provides Save and Delete functionalities. The delete button is made available when the category is not assigned. The Save button is disabled while in View mode. The Delete button is active in any modes.

Add New Category is used to create a new placeholder for adding a new category, followed by clicking the Save button to add the new category, as shown below.

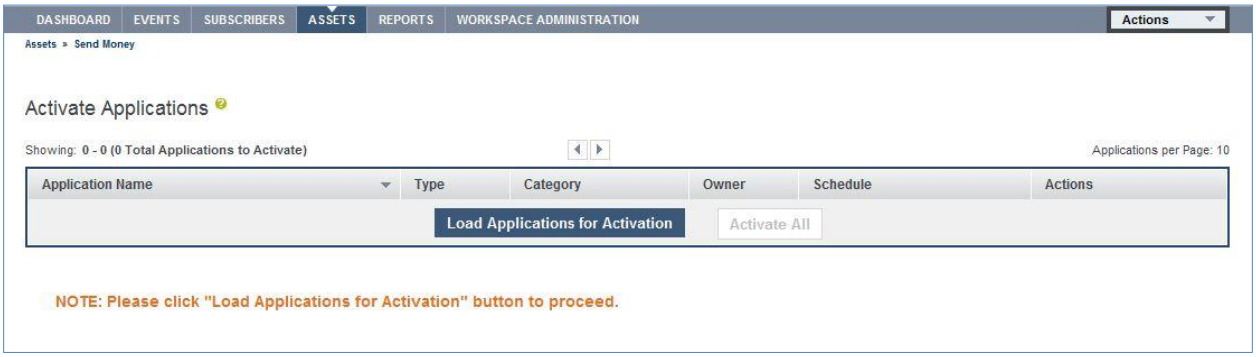


Edit Mode or View Mode is a toggle between the Edit and View Modes. The link shows "Edit Mode" when the screen is in View mode, vice versa. In addition, when in edit mode the category textbox is editable and the text is in **bold**. Changes need to be followed by clicking the Save button for the changes to be permanent.

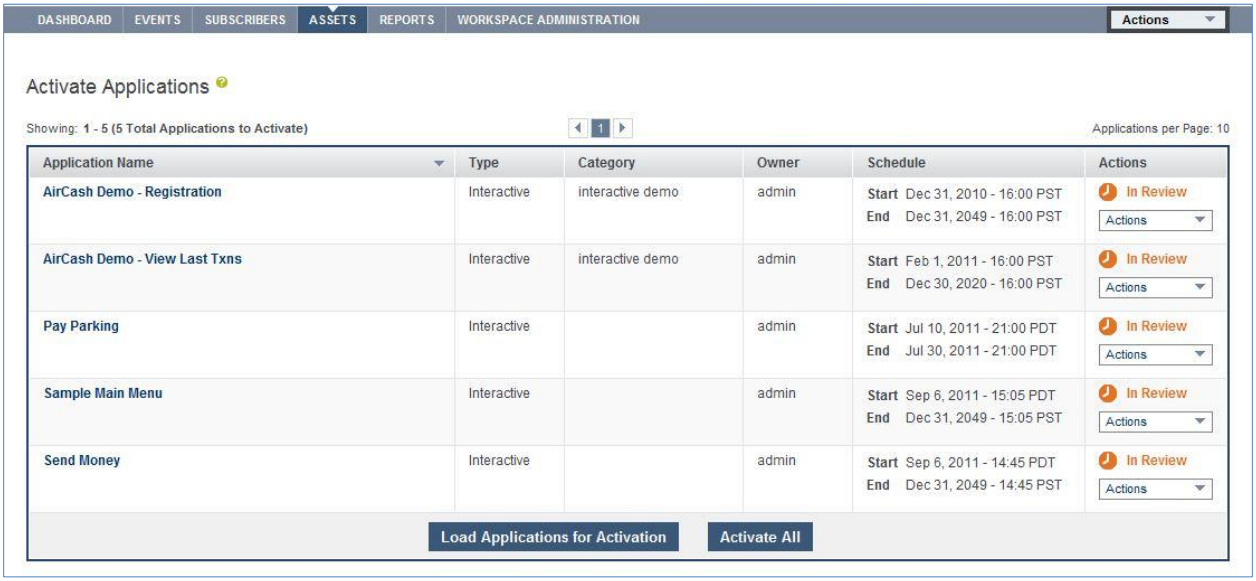


2.8 Activate Applications

The **Activate Applications** screen is accessible from several screens including: **Dashboard** – My Applications module, **Assets**, and **Application –** Application Actions – Activate Applications. The screen has the List Table layout. The list is not loaded by default.

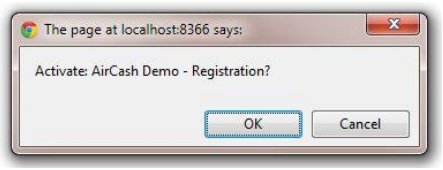


Click the "load Applications for Activation" button to load the applications. Only the applications that needs activation are shown in the table.

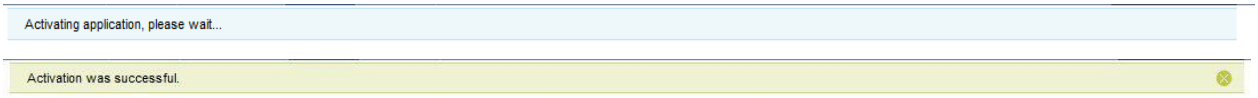


Newly created applications or applications that have their settings changed (including: name, dates, application flow, state configurations, etc.) needs to be activated to become "***Live***". For new applications, becoming "Live" means that the application will automatically be "In Service" when the current date is equal to the start date. In addition, when in-service these applications can be tested using the simulator, and if the workspace have a real channel then messages can be sent from or delivered to the mobile handset. For modified applications, being live means that the changes have been propagated and they can be tested from the simulator or mobile handset.

There are two ways to activate applications: Actions – Activate link, or Activate All button. The former is recommended method especially in the production environment. The "Activate All" button is useful for the development and test environments, or for the production environment when the operator is absolutely certain that all listed applications need to be activated. Both mechanisms provide an extra confirmation popup dialog, as shown below, that enables the operator to cancel the action by clicking "Cancel", or clicking the close button.



The activated application will disappear from the list. When "Activate All" is clicked, the table should be cleared. The feedback panel provides additional information during the activation process, as shown below.



When in doubt, check the application details by clicking on the application name, or use the Actions – Show Details link. This action will navigate to the **Application** screen.

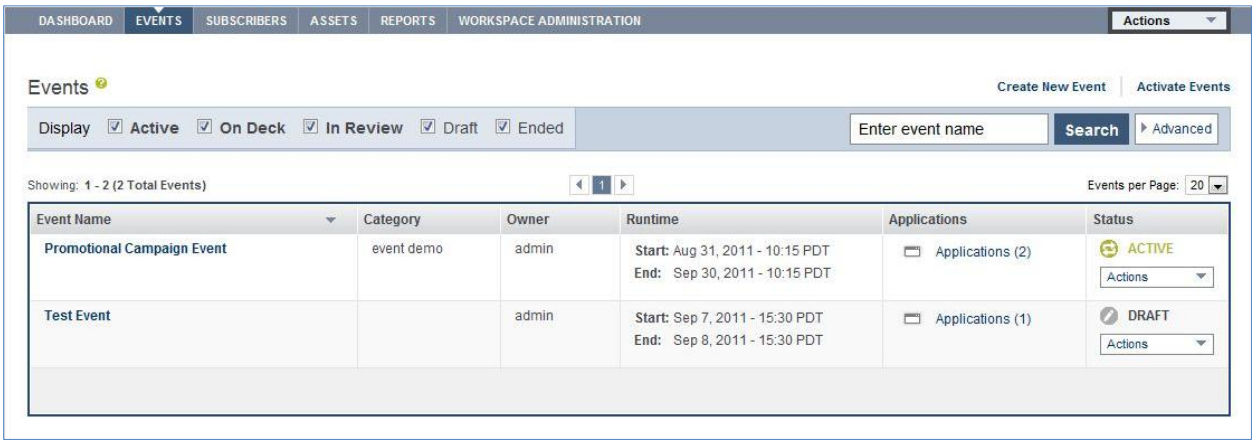
2.9 Events

The "***Event Model***" or "***Event***" for short is a container for storing the event configurations including event schedules and event handler. This event is used by the Brand Mobiliser processing engine to trigger the event handler when the schedule is current. The event handler is an event application that is designed for workflow or batch processing and is triggered by event, such as: scheduled time or other system events. Event configurations include: event runtime, scheduled window (manual or recurring), an event application to invoke when the trigger occurs, and related interactive applications for marker.

An example will better illustrate what event is. An event is created to host a "New Product Campaign" that has a planned "***Runtime***" of between Nov. 1 and Nov 30 timeframe. Within this timeframe, multiple time segments or "***Window***"(s) are scheduled to send out the promotional discount codes. The schedules can be defined using the "***Manual***" or "***Recurring***" window modes. An event "***Trigger***" will occur for each of these schedules to start the task of sending out the discount codes. In addition to the discount codes, a reply "keyword" is included so that subscribers can opt-in to receive more discount codes in the future.

Event adopts the same versioning model as application: In-Review and Live. Therefore, event needs to be activated to become live. In addition, an event needs to have an assigned "in-service" event application before it can be activated. Note: "in-service" application is a live application and has started. A live application may not be "in-service" due to the current date is outside the Active From and To dates. The process of activating an event does not include activating the assigned event application. Therefore, an event application needs to be activated and in-service first before it can be assigned to an event.

The **Events** screen displays the list of events and is accessible by clicking the Events navigation item. Depending on the role(s) of the login user, the display events may vary. For the SUPER ADMIN, ADMIN and APPLICATION ADMIN roles the display events include all the events in the workspace, otherwise (the APPLICATION OWNER role) it will only include events that belong to the login user. Filters are not yet supported in the current version.



Events screen also provides links to other events related functionalities, such as:

- Create New Event – navigates to the **New Event** screen. See "Create New Event" section for details.
- Activate Events – navigate the **Activate Events** screen. See "Activate Events" section for details.

The event list is sorted by the Event Name column, and can be toggle for ascending and descending order by clicking the column header. The followings describe each columns in details.

- The Name column displays the Event Name that is also a hyperlink to the **Event** screen
- Category column displays the category assigned to this event. The benefit for assigning a category to an event is for filtering. New categories can be created using the **Manage Categories** screen, accessible from "Navigation – Actions – Manage Categories"
- Owner is the user who create the event
- Runtime displays the Start and End date and time of the event
- Application displays the number of applications associated with this event. When clicked, the screen navigates to the Event screen - Event Application tab
- Actions column displays the event status and provides the context aware Actions menu, both are described in details below.

  The event status is shown as text and color, as follows.

DRAFT	corresponds to newly created event, and has not been activated
ON_DECK	corresponds to activated event, and pending start date
ACTIVE	corresponds to Live event
ACTIVE	corresponds to Live event but has been revised pending review and activation
ENDED	corresponds to terminated event because the end date has occurred

  The Actions menu items include:

  - **Actions - *Show Details*** – link to the **Event** screen showing details of the corresponding event. Please refer to the "Event Details" section.
  - **Actions - *Delete*** - is for deleting the event in the selected row. A confirmation dialog showing the event name and the "OK" or "Cancel" buttons. Clicking OK will result in deleting the event, while Cancel will just close the dialog. Upon a successful delete, the list is refreshed with updated entries.

2.9.1 Create New Event

To create a new event, start from the **Add New Event** screen. For convenience, this screen is accessible from several locations including:

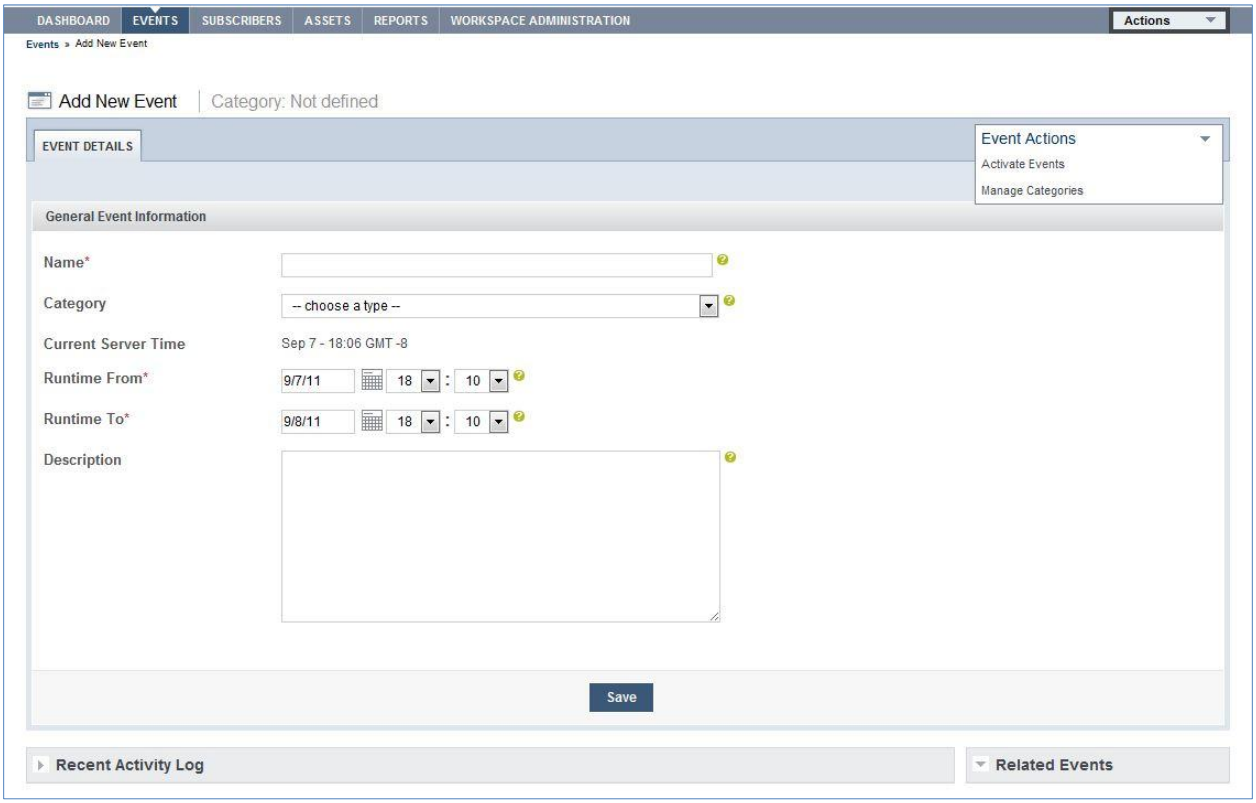
- From the **Dashboard** screen – My Events module – Create New Event link at the footer of the module
- From the **Events** screen – Create New Event link.

Clicking the Create New Event link navigates to the **Add New Event** screen showing the "Event Details" tab, as discussed on the next section.

2.9.2 Event Details

The new event screen starts with a single "Event Details" tab, showing the information needed to create a new event. Once the mandatory fields are provided and saved, additional tabs will be displayed. For existing events, the event details can be used to modify existing settings.

The followings describe fields, action links and buttons on the Event Details tab, shown below.



Mandatory fields:

- Name – name of the event, and is used to uniquely identify the event. Duplicate name within the workspace is allowed but not recommended. The developer is responsible to check prior to assigning the name.

- Runtime From – the date & time when the event will be in service.
- Runtime To – the date & time when the event will stop being in service.

  Both from and to dates are based on the server date and time. The "Current Server Time" is displayed to aid in establishing the correct date and time.

Optional Field that appears on both application types (Interactive and Event):

- Category – used for easily grouping events in the **Events** screen. The dropdown list displays the available categories. New category may be added using the **Manage Categories** screen, that is accessible from either Actions – Manage Categories or from Application Actions – Manage Categories links. Please refer to "Manage Categories" section for details.
- Description – additional descriptions for the event, for user viewed only.

Buttons:

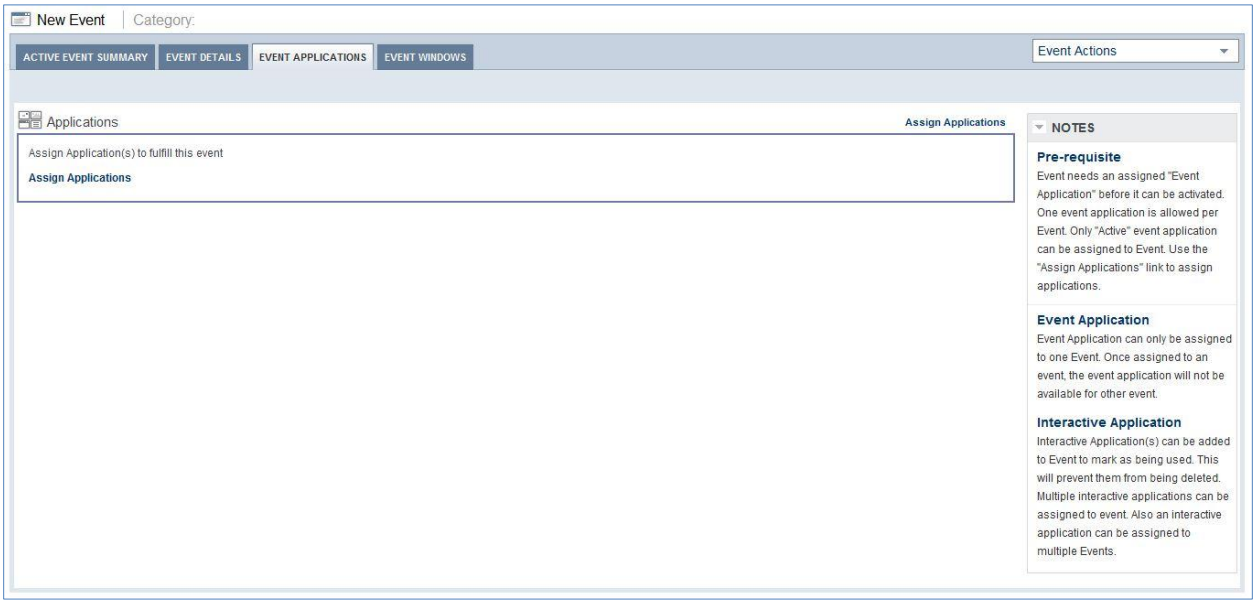
- Save – save the event. Once the event is saved successfully, three additional tabs will appear: Active Event Summary, Event Applications and Event Windows. Details on each tabs is discussed below.

Event Actions is a dropdown list that contains commonly used links (shortcuts) related to event. The content of the dropdown list is universal across all four tabs: Active Event Summary, Event Details, Event Applications and Event Windows.

- Activate Event – once an event creation is complete, it needs to be activated before it become "***Live***". The same applies when modification is made to event settings, or when changing the assigned event application. For general discussions on Activation, refer to the "Activation" section. Please refer to "Activate Events" section for detailed discussions on how to activate event.
- Manage Categories – adds new or modifies categories. Please refer to "Manage Categories" section for details.

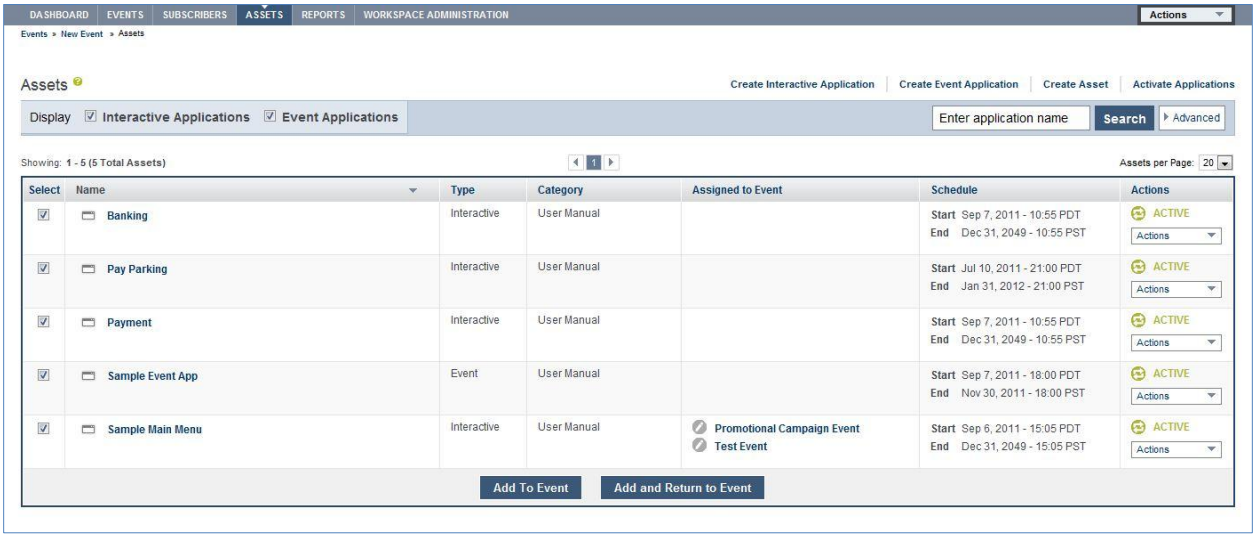
2.9.3 Event Applications

The "Event Applications" tab provides the tool for assigning applications to the event. For a newly created event with no assigned application, the screen looks like the following.



Each event can only have one assigned event application, and it will be the application that gets invoked when the trigger occurs. However, multiple interactive applications can be assigned to the event for the purpose of locking

the applications so that they are not deleted while still in use. A warning popup dialog will be shown when trying to delete a locked application, for example, from the **Assets** screen. To assign applications to the event, click the "Assign Applications" link, and the screen will be navigated to the **Assets** screen as shown below.

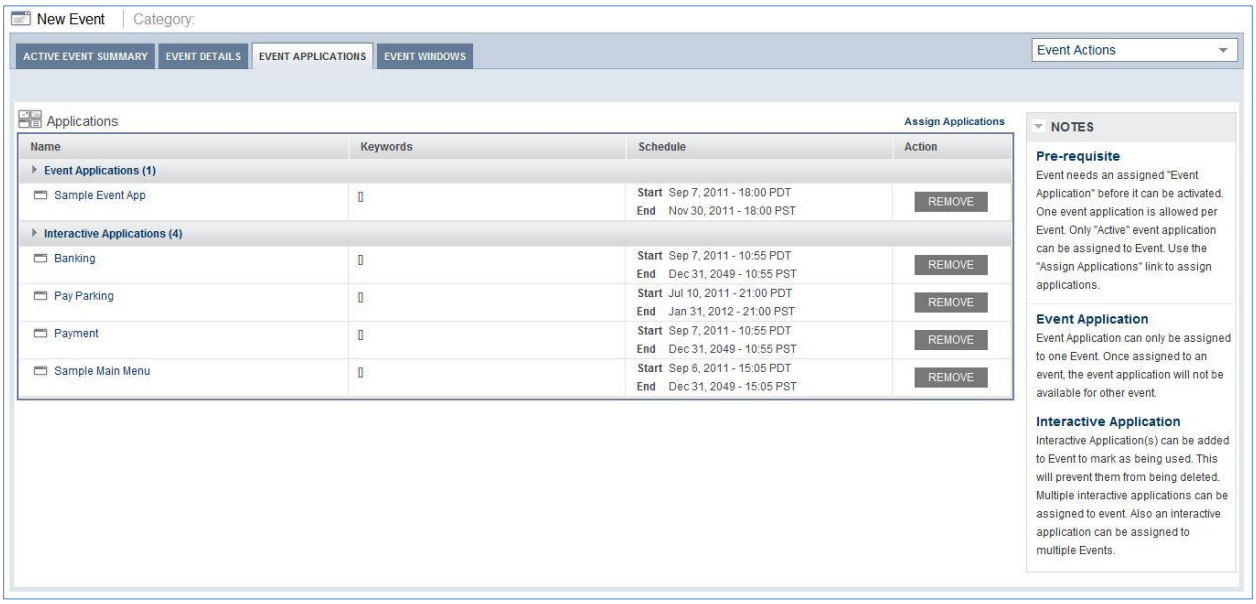


This **Assets** screen has been customized with buttons necessary for assigning application(s) to the corresponding event. In addition, the **Assets** screen has been associated with the corresponding event, as shown in the breadcrumb [Events > New Event > Assets]. The associated event name is "New Event". The name is also a hyperlink that navigates back to the **Event** screen. However, clicking the breadcrumb Events link navigates to the **Events** list screen and clear the event association.

To assign application(s) to the associated event, click the checkbox to select the application(s). Then click the "Add To Event" or "Add and Return to Event" buttons to save them to the event with the following difference. Clicking the "Add To Event" assigns the selected application(s) to the event but remain on the **Assets** screen, while the later navigates back to the **Event** screen showing Event Applications tab. Note: since only one event application can be assigned to an event, if more than one event applications are selected, the first event application in the list will be assigned to the event. If prior to navigating to the **Assets** screen the Event already has an assigned event application, the application list will have been filtered to not showing any event applications. So, the assigned event application needs to be deleted prior to clicking the Assign Applications link for changing the assigned event application.

The "Assigned to Event" column on the **Assets** screen shows the list of events that the application has been assigned to. The name is also a link that navigates to the **Event** screen showing the corresponding event. Filters can be used to narrow the list. Note: clicking any links on the **Assets** screen (other than the "Add To Event", "Add and Return to Event" and "Search" buttons) will detach the linked event.

The following screen shows the list of applications assigned to the event. There are one event application and four interactive applications. To remove an assigned application, click the "Remove" button. Like other list screens, the application name is a hyperlink that navigates to the **Application** screen. Event application does not have assigned keyword but interactive application may have assigned keyword(s) and they will be shown in the Keywords column. The schedule column corresponds to the Active From and To dates of the Application.

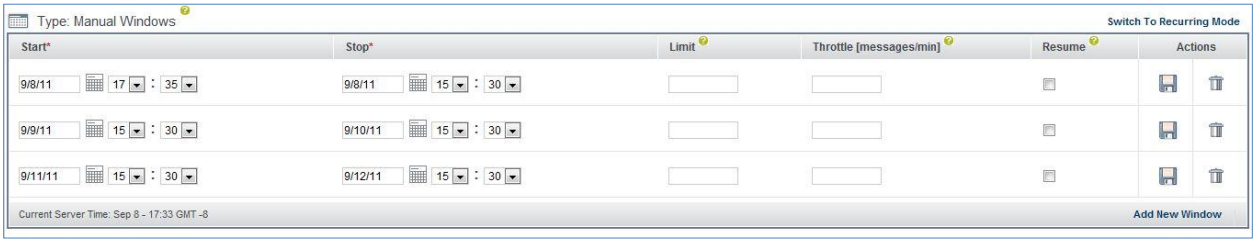


The above screen shows a commonly implemented scenario in the mobile commerce industry, as follows. The Mobile Commerce 365 Inc. is launching new mobile services (i.e., Mobile Banking, Mobile Pay Parking, and Mobile Payment). On the launch date, the marketing department will conduct SMS campaign to notify their subscribers of the new services (using the Sample Event App). The SMS message that will be sent out to their subscribers reads "New mobile services for your convenient. Reply menu365 to start. $50 will be credited to your mobile wallet". When the subscriber reply with the keyword "menu365", the "Sample Main Menu" interactive application will be invoked and the main menu will be sent to the subscriber as follows: "Choose: 1 – Banking 2 – Pay Parking 3 – Other Payment". The menu items are linked to the interactive applications: Banking, Pay Parking and Payment, respectively.

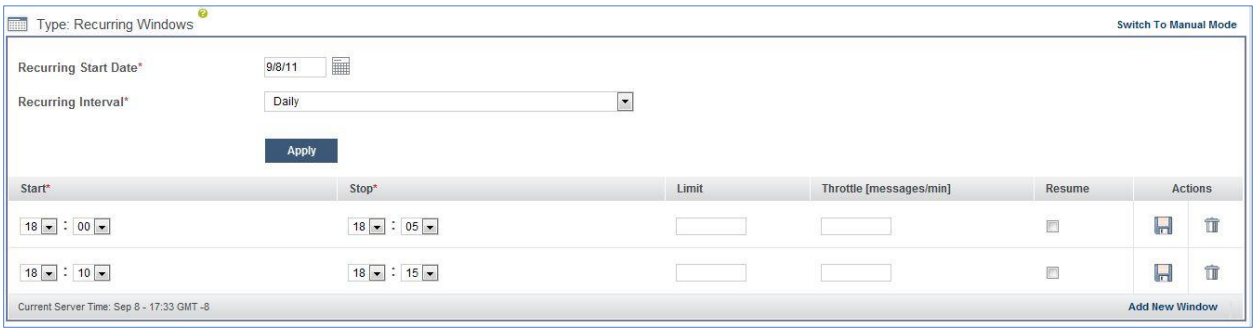
The event application is assigned to the event so that it gets invoked when the event windows occurs (see Event Windows section for how to setup the windows). The interactive applications are also assigned to the event as a marker so that they are traceable to a specific purpose or business event. The event name in this case is "New Event" but could as well be called "New Mobile Services 2011 Event" to be specific. So when the business group refers to the "New Mobile Services 2011 Event", it refers to "New Event" and all the associated SMS campaign (Sample Event App) and interactive applications (Sample Main Menu, Banking, Pay Parking, and Payment).

2.9.4 Event Windows

The Event Windows tab provides the tool to set up the scheduled time-based events, called the "***Event Window***". When a window becomes current, an "***Event Trigger***" will be generated that will invoke the assigned event handler (i.e., event application). There are two types of window: manual or recurring, shown below. In the recurring mode, the recurring type recurring start date are specified first, and then the "***Event Timeslot***" is added manually.



Manual Windows Setup



Recurring Windows Setup

The setup for the above *manual window* translates to three Event Triggers occurring on the Start (date & time): 9/8/11 at 17:35, 9/9/11 at 15:30, and 9/11/11 at 15:30. Once triggered, the processing will continue until the Stop (date & time): 9/8/11 at 15:30, 9/10/11 at 15:30, and 9/12/11 at 15:30, respectively.

The setup for the above *recurring windows* translates to an Event Trigger that will start to occur from the "Recurring Start Date" of 9/8/11, and the trigger will occur on the daily-basis at the Start (time): 18:00 and 18:10, and the processing will continue until the Stop (time): 18:05 and 18:15, respectively. The Event Trigger will stop occurring on the "Runtime To" date defined in the Event Details tab.

Switching between modes is accomplished using the toggle links: Switch To Recurring Mode and Switch To Manual Mode. Switching is allowed only when there are no entries; i.e., no manual windows, or no timeslots. Existing entries need to be deleted to enable switching.

The event parameters include:

- **Start** – configures when (in date & time) the event trigger will occur; Date & time fields for event window; Time field only for event timeslot, because the date is calculated automatically by the scheduler engine based on the "Recurring Start Date" and "Recurring Interval".

  Using the above screen, the first event trigger will occur on 9/8/11 at two timeslots: 18:00 and 18:10. The next event trigger will be calculated by the scheduler and it should occur a day later (from 9/8/11) at the same timeslots: 18:00 and 18:10, because the recurring interval is "Daily". The subsequent event triggers will be calculated the same way and it will continue until the end of the event runtime as configured in the Event Details tab – Runtime To field.

  Note: the time field is defined in "hour:minute" with the minute resolution of 5 minutes.

- **Stop** – configures when the processing should stop. Date & time fields for event window; Time field only for event timeslot.

- **Limit** – configures the maximum number of loop-back to process. When used in conjunction with the throttle, it should be specified as a multiple of throttle. For example, throttle = 60 messages/min; specify limit of 60, 120, or 180, etc.

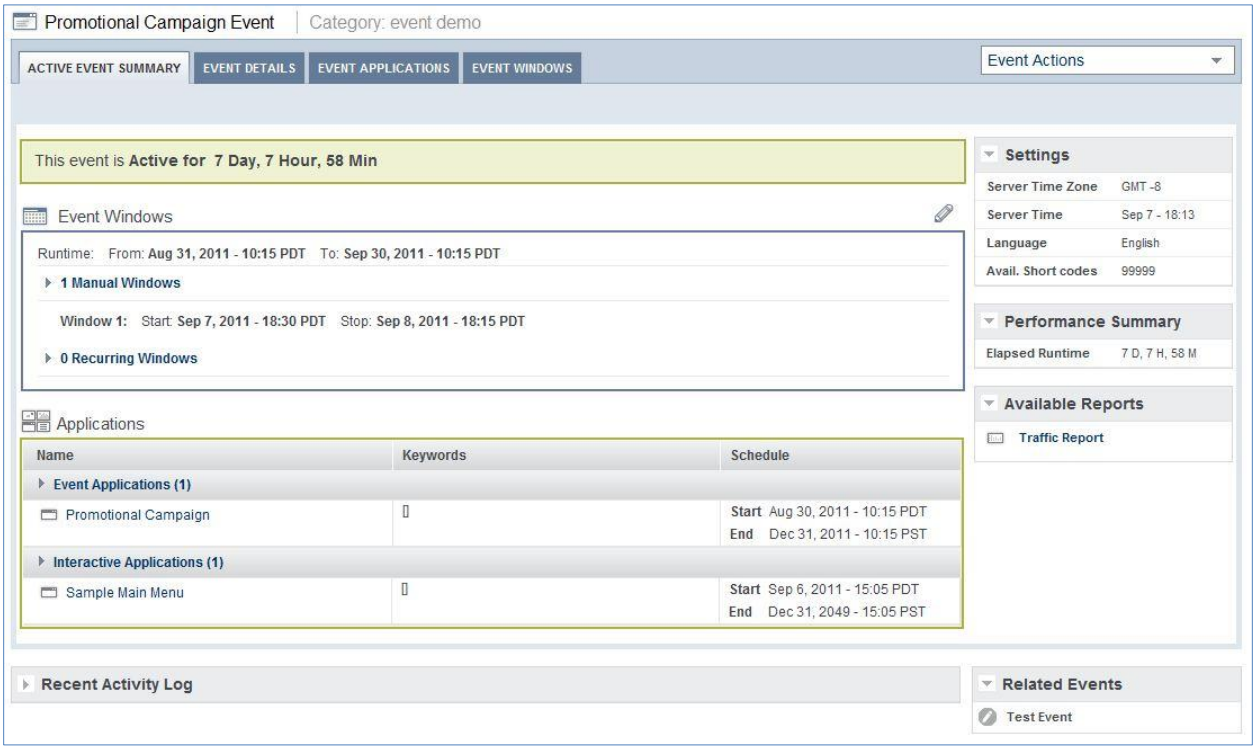
- **Throttle** – configures the maximum processing rate. **NOTE**: throttle is not a Service Level Agreement (SLA) on sending rate. Throttle is a control to NOT exceed the specified rate. The actual sending rate is governed by the rate of the channel and not by the throttle. In fact, throttle, if specified, should always be set to a lower value than the channel rate so that it does not overflow the internal queue resulting in messages being dropped.

- **Resume** – configure the resume flag. An extra parameter for the state that processes list, such as Subscriber list, to indicate whether to restart from the beginning of the list or to resume from the last processed item.

The three control parameters, Limit, Throttle and Resume, apply to event applications that have a state that uses these parameters to control the automatic application loop-back. For example, the "Process Subscriber" state uses these control parameters as described in "Process Subscriber State" section.

2.9.5 Active Event Summary

The Active Event Summary tab summarizes, in view-only mode, the Live version of the event configurations. The other three tabs display the In-Review version, therefore they can be modified. The Active Event Summary tab is based on the "Module" layout, as shown below. Each modules will be discussed in more details in the subsections.



2.9.5.1 Event Status Module

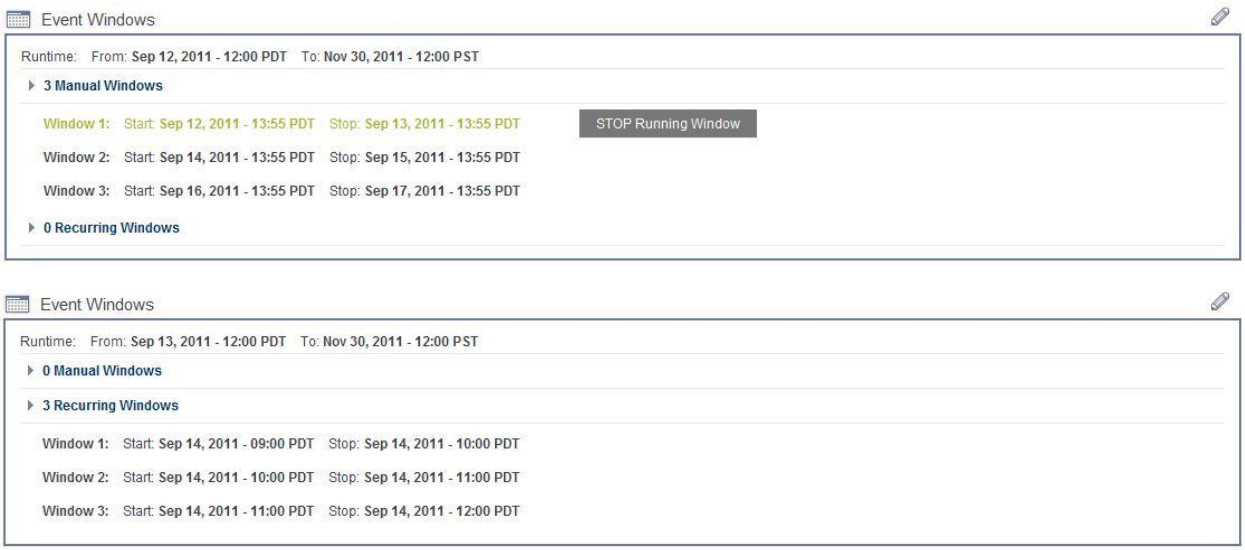
The first module on the top left hand side shows the current status of the event. When the event is in DRAFT mode, the module displays buttons for assign applications and create event windows, that navigate to the event applications and event windows tabs, respectively, as shown below.

For an "in-service" event, the event status module shows how long the event has been in service as shown below.

2.9.5.2 Event Windows Module

The event windows module displays the event runtime, and the windows grouped by types. Only one type will be in used at a time. In addition to displaying the event runtime, it also displays the list of manual or recurring windows. **Note**: the list displayed on the event windows module (live version) may be different from the list shown on the event windows tab (in-review version) if changes have been made on the event windows tab but have not been activated yet.

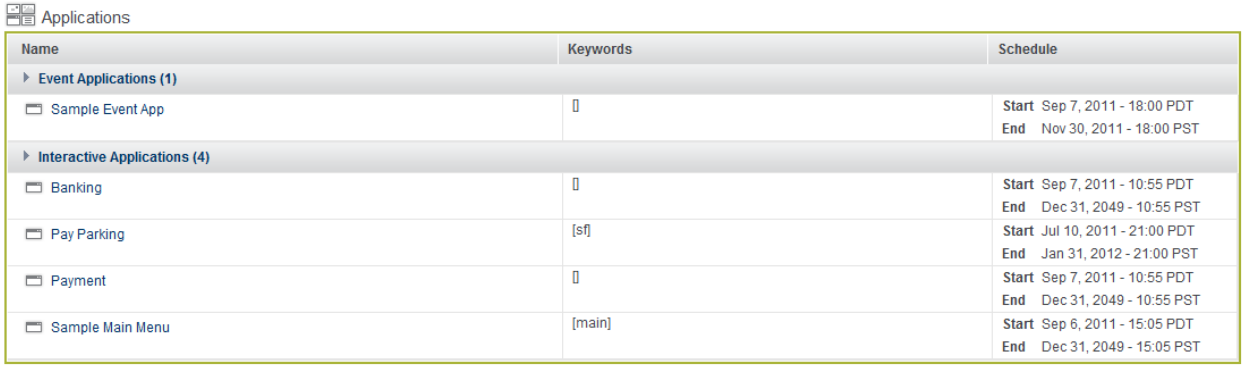
Use the ▸ to expand and show the list. The screen below shows the manual and recurring list, respectively.



The "*current*" window will be displayed in green, and if the process is running a "STOP Running Window" button will also be displayed to allow stopping the process. There are several possible scenarios when the window is green but the button is not shown including: the processing has not started, has completed earlier, or has encountered error and terminated, or the processing is idle because it is setup to run at the throttling rate and in this case refreshing the screen multiple times may show the button.

2.9.5.3 Applications Module

The Applications module is displayed below the event windows module, showing the required event application and the optional interactive applications that are assigned to the corresponding event. Also displayed are the keyword(s) and the schedule dates. Note: event application does not have keyword. Use the ▸ to expand or minimize the list. The applications module could be used to confirm that the event runtime is **within** the schedules of all assigned applications for proper operation. Currently, this is not validated by the tool while assigning the application or approving the application.



2.9.5.4 Settings Module

The Settings module displayed: server time zone, server time, language and available short codes. Note: all the date and time setup is based on the server time, and not the user's computer date and time.

2.9.5.5 Performance Summary Module

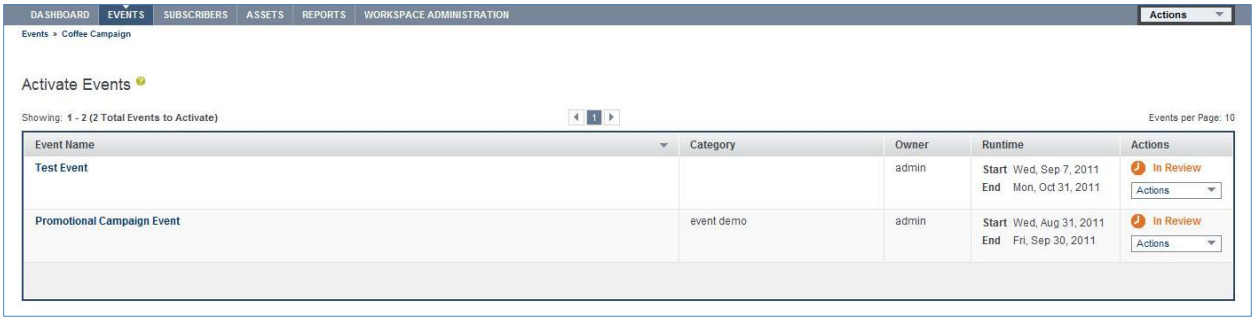
The Performance Summary module currently displays the total elapse runtime of the event. More information will be added to this module in the future release.

2.9.5.6 Available Reports Module

In the current version, there is one link to the **Traffic Report** screen.

2.10 Activate Events

The **Activate Events** screen is accessible from several screens including: **Dashboard** – My Events module, **Events**, and **Event** – Event Actions – Activate Events. The screen has the List Table layout as shown below.



Newly created events or modified events needs to be activated for the changes to become "***Live***". Like applications, events need to be activated before it can be tested with the simulator. The functionalities on this screen is the same as the **Activate Applications** screen.

2.11 Subscribers

For background and introduction on Subscriber, please refer to the "Brand Mobiliser Subscribers" section. The **Subscribers** screen list all the subscriber sets in the workspace. A subscriber set is the grouping of subscribers. Usage includes: notification set containing list for subscribers for notification, opt-in/opt-out set, subscription set, etc. A set has a unique ID and a name. The name does not have to be unique but since it is used to identify the set and to avoid confusion, it should be kept unique. The sets are listed by name when they displayed on the State Property Editor, such as, Process Subscriber, Get Subscriber, Add Subscriber and Update Subscriber states., even though the unique ID is stored when selected.

There are three mechanisms to create Subscriber Set and to populate it:

- Create Empty Set – create an empty subscriber set that is later populated by: Upload Merge, or Populated by "Add Subscriber" state in the Event or Interactive application. Use the Create Empty Set link to navigate to the **New Subscriber Set** screen. See "Create Empty Set" section below for details.
- Upload Subscriber – create a subscriber set and populate it with subscribers from the uploaded file. Use the Upload Subscriber link to navigate to the **Upload Subscriber File** screen. See "Upload Subscriber" section below for details.

The following columns are displayed for each subscriber sets:

- Set Name – the Subscriber Set name. Clicking the name will navigate to the **Set Preview** screen
- Attributes – User defined attribute name for each free form columns. There are a total of 20 free form columns. Only columns with attribute name will be displayed in the **Set Preview** screen.
- No. of Subscribers – the total number of subscribers in the set. Note: when creating the subscriber using the Upload Subscriber, the total number of subscribers will change while the set is being populated. The final number is when the "Load Success" status has been reached.
- Source Filename – the filename of the uploaded subscriber file with unique ID extension; Supported files are: CSV and compressed CSV (zip). When uploaded filename has a ".zip" extension, the file will be treated as compressed file so it will be uncompressed prior to processing. Once uncompressed, the file is treated as a normal CSV file.
- Status – status of the subscriber set. For newly created empty set, the status is "Not Started". For successful upload, the status is "Load Success". Other statuses include: "Load In Progress", "Load Failed", "Purge In Progress", "Purge Success", and "Purge Failed".
- Create Date – date when the subscriber set was created
- Actions – includes Preview, Upload Merge, Purge and Delete.

  Preview – navigates to the **Set Preview** screen to display a preview of up to 2000 subscribers

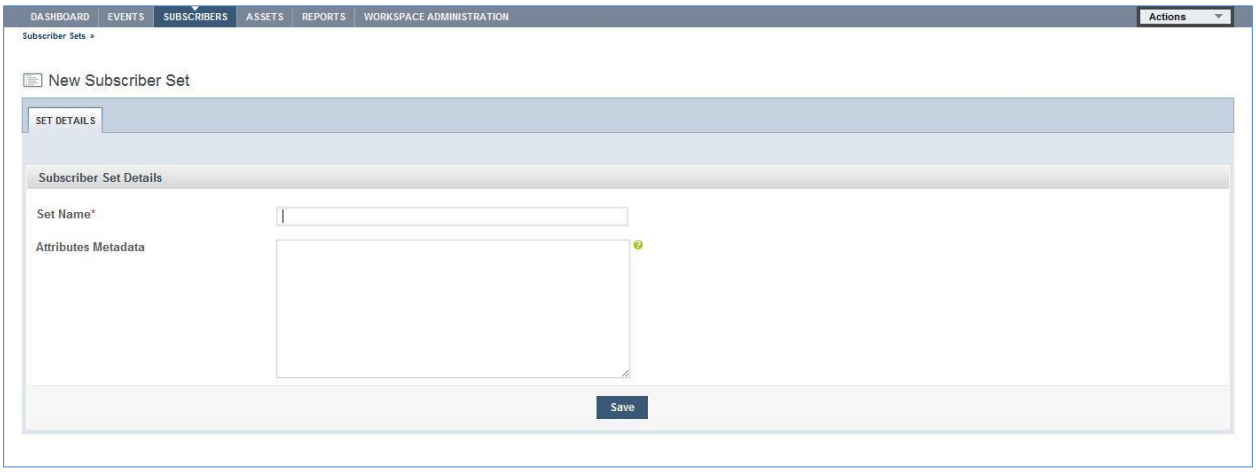
  Upload Merge – for uploading a subscriber file and merge it into the corresponding subscriber set. The link navigates to the **Upload Subscriber** screen with the pre-configured subscriber set. On the **Upload Subscriber** screen, the Set Name will be populated and not editable.

  Purge – for purging the subscriber set. Note: subscriber set needs to be purged before it can be deleted. In other words, only empty subscriber set can be deleted. The purge process will occur in the background and depending on the number of subscribers in the set it may take some times to complete. Immediately after clicking the Purge, the Subscriber Set list will be refreshed and the status will show "Purge In Progress". When the purge is complete, the status will show "Purge Success" or "Purge Failed".

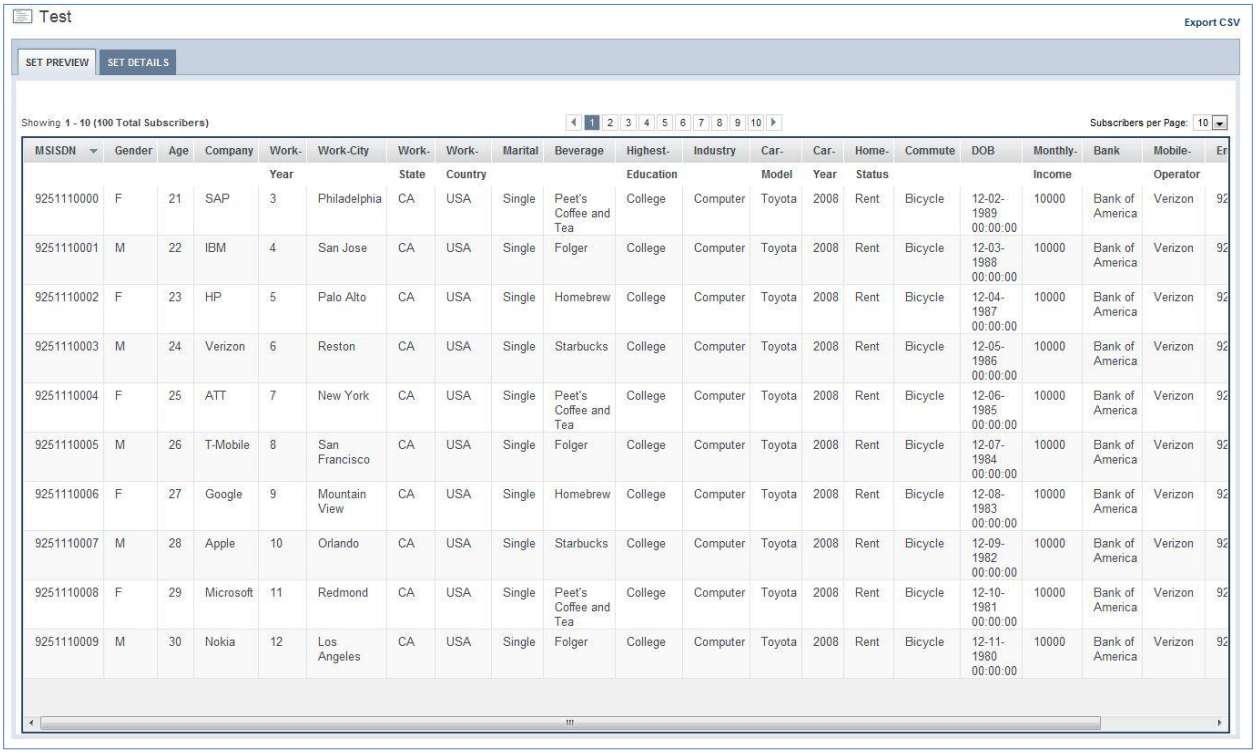
  Delete – to delete an empty subscriber set. If the set is not empty the Delete link will be disabled. First, click the Purge link to delete all the subscribers in the set. If the subscriber set contains a large number of subscribers, you may have to wait and then refresh the screen until the Status column shows "Purge Success" before proceeding with the delete of the subscriber set.

2.11.1 Create Empty Set – Set Details Tab

To create an empty set, click the Create Empty Set link displayed on the top right hand side of the Subscribers screen and the screen will navigates the New Subscriber Set screen shown below. Currently, there are two fields to populate: set name (required) and attributes metadata. The attribute metadata is a comma separated attribute names that will be assigned to the subscriber free-form fields (up to 20). This is an optional field.



The attribute metadata is used as the column header name in the **Set Preview** screen, also shown below.



MSISDN	Gender	Age	Company	Work-Year	Work-City	Work-State	Work-Country	Marital	Beverage	Highest-Education	Industry	Car-Model	Car-Year	Home-Status	Commute	DOB	Monthly-Income	Bank	Mobile-Operator	Er
9251110000	F	21	SAP	3	Philadelphia	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-02-1989 00:00:00	10000	Bank of America	Verizon	92
9251110001	M	22	IBM	4	San Jose	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-03-1988 00:00:00	10000	Bank of America	Verizon	92
9251110002	F	23	HP	5	Palo Alto	CA	USA	Single	Homebrew	College	Computer	Toyota	2008	Rent	Bicycle	12-04-1987 00:00:00	10000	Bank of America	Verizon	92
9251110003	M	24	Verizon	6	Reston	CA	USA	Single	Starbucks	College	Computer	Toyota	2008	Rent	Bicycle	12-05-1986 00:00:00	10000	Bank of America	Verizon	92
9251110004	F	25	ATT	7	New York	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-06-1985 00:00:00	10000	Bank of America	Verizon	92
9251110005	M	26	T-Mobile	8	San Francisco	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-07-1984 00:00:00	10000	Bank of America	Verizon	92
9251110006	F	27	Google	9	Mountain View	CA	USA	Single	Homebrew	College	Computer	Toyota	2008	Rent	Bicycle	12-08-1983 00:00:00	10000	Bank of America	Verizon	92
9251110007	M	28	Apple	10	Orlando	CA	USA	Single	Starbucks	College	Computer	Toyota	2008	Rent	Bicycle	12-09-1982 00:00:00	10000	Bank of America	Verizon	92
9251110008	F	29	Microsoft	11	Redmond	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-10-1981 00:00:00	10000	Bank of America	Verizon	92
9251110009	M	30	Nokia	12	Los Angeles	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-11-1980 00:00:00	10000	Bank of America	Verizon	92

Set Preview with Attribute Metadata

If the Attribute Metadata is left empty, there will be no attribute columns displayed on the **Set Preview** screen of the corresponding subscriber set, as shown below. Note: removing the attribute metadata does not delete the data but only change the set preview column display. The attribute metadata can be added at later point to redisplay the attribute columns.

Set Preview with No Attribute Metadata

2.11.2 Upload Subscriber

The Subscriber Set can be created by populating it with subscribers from the uploaded CSV file using the **Upload Subscriber** screen shown below. File types and format are discussed below. To upload the file, enter the set name and then select the file to be uploaded from the local drive, and click the Upload button. The file will be uploaded to the server using the streaming mechanism. The browser needs to wait until the streaming is complete and a success message is displayed beneath the "Upload" button before navigating to the **Subscribers** screen, by clicking the Subscribers on the main navigation, to check the upload status and preview the data.

Two file types are supported: Comma Separated Value (CSV) and Compressed CSV (ZIP). The CSV file contains comma separated attribute names on the first row, and comma separated values on the remaining rows. The ZIP file is the compressed CSV file with "zip" extension. Uploading the ZIP file is preferred method especially for file containing a large number of subscribers. For file containing comma separated values, the compression can reduce the file size by as much as a factor of 10, reducing the upload streaming significantly and the user wait time.

The first row of the CSV file should be the attribute names with the first attribute called MSISDN. The maximum number of attributes is 21 including the MSISDN. The remaining rows are the values rows. The first field (KEY) is the unique "key" typically used to store the subscriber's phone number (or MSISDN). The remaining 20 fields have free form format. The first free form field (ATTRIB1) can store up to 1000 characters while the remaining free form fields store up to 320 characters. Example of the CSV file format is shown below.

```
MSISDN,FirstName,LastName,Gender,Age,HomeState
+19991111111,John,Doe,M,23,CA
+19991112222,John,Black,M,33,WA
+19991113333,Pat,Doe,F,43,IA
```

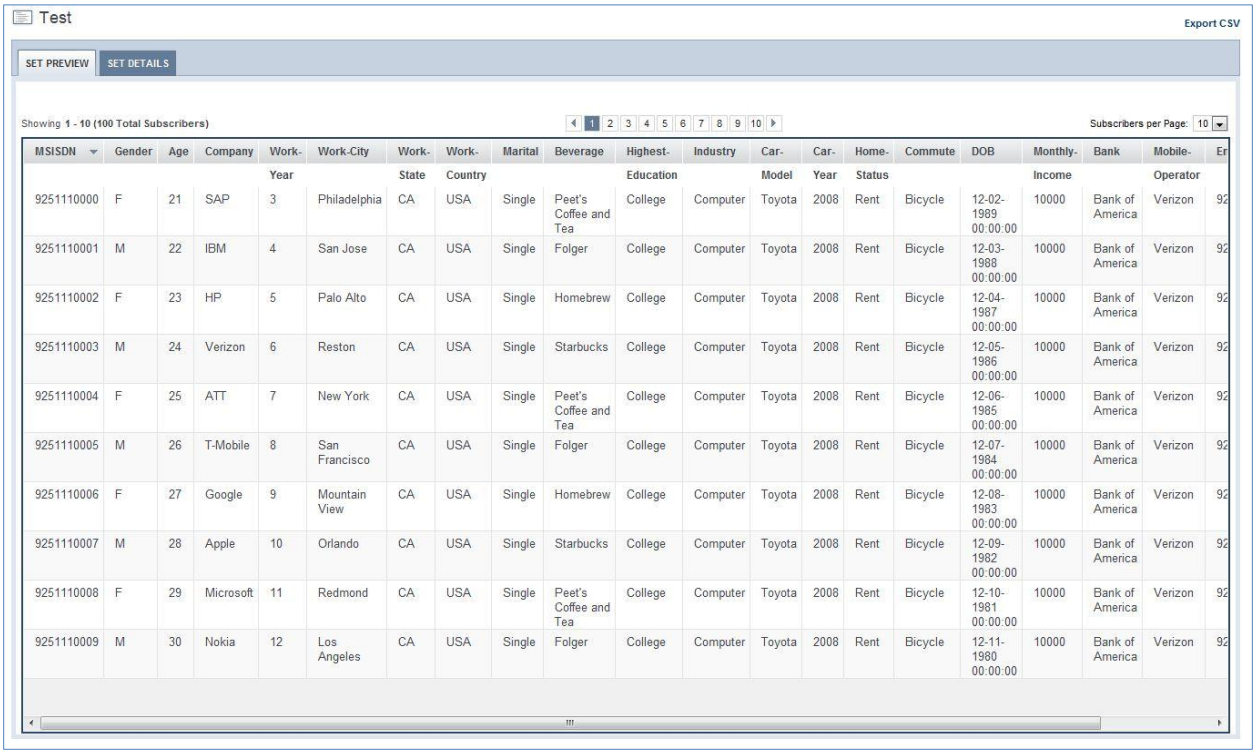
NOTE: if the MSISDN column contains subscriber's phone number and they are to be used to send SMS message, you need to consult the SMS provider on the format. For example in United States, it should be like +19992224444 where 999 is the area code.

The file upload process is a two-step process: streaming the file to the server, then processing and uploading the file content to the database. Depending on the uploaded file size and the system utilizations, the latter may take some times. However, once the file is stored on the server the subscriber set will be created enabling users to track the latter process from the **Subscribers** screen. The upload process life cycles changes the status in the following order: Not Started, Load In Progress, Load Success. While in the Load In Progress status, users may be able to see the number of subscriber column changes.

2.11.3 Set Preview

Set Preview screen is accessible from the **Subscribers** screen by clicking on the subscriber set name. The displayed columns are defined by the attribute metadata in the "Set Details" tab. The total number of rows displayed on the set preview screen is limited to 2000. The functionalities provided on the set preview screen is intentionally limited because, as discussed earlier, the set should be used as a temporary (staging) storage and therefore they should modified to prevent out-of-synch. All the necessary data manipulations should be done at the system of record or the data pre-processor stage prior to uploading to subscriber set.

The Export CSV link can be used to export the complete subscriber set to the CSV file. The export functionality is useful in many cases. For example, some of the subscriber sets were created from the Upload Merge of multiple files, so the export will result in the combined. The subscriber set may have been created as an empty set in the beginning, and later populated by the application through the options like: opt-in, subscribe, log for reporting, etc. The export provides the mechanism to transfer the data to the system of record, or reporting. Alternatively, an automatic mechanism can be developed to perform the transfer using application and custom state.

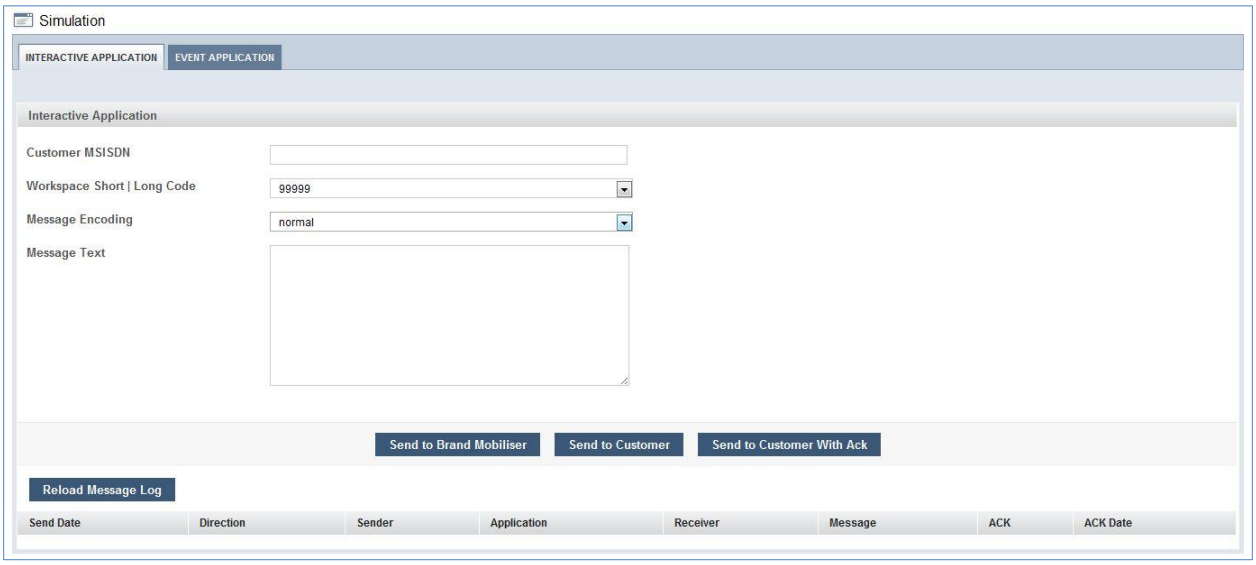


MSISDN	Gender	Age	Company	Work-Year	Work-City	Work-State	Work-Country	Marital	Beverage	Highest-Education	Industry	Car-Model	Car-Year	Home-Status	Commute	DOB	Monthly-Income	Bank	Mobile-Operator	Er
9251110000	F	21	SAP	3	Philadelphia	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-02-1989 00:00:00	10000	Bank of America	Verizon	92
9251110001	M	22	IBM	4	San Jose	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-03-1988 00:00:00	10000	Bank of America	Verizon	92
9251110002	F	23	HP	5	Palo Alto	CA	USA	Single	Homebrew	College	Computer	Toyota	2008	Rent	Bicycle	12-04-1987 00:00:00	10000	Bank of America	Verizon	92
9251110003	M	24	Verizon	6	Reston	CA	USA	Single	Starbucks	College	Computer	Toyota	2008	Rent	Bicycle	12-05-1986 00:00:00	10000	Bank of America	Verizon	92
9251110004	F	25	ATT	7	New York	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-06-1985 00:00:00	10000	Bank of America	Verizon	92
9251110005	M	26	T-Mobile	8	San Francisco	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-07-1984 00:00:00	10000	Bank of America	Verizon	92
9251110006	F	27	Google	9	Mountain View	CA	USA	Single	Homebrew	College	Computer	Toyota	2008	Rent	Bicycle	12-08-1983 00:00:00	10000	Bank of America	Verizon	92
9251110007	M	28	Apple	10	Orlando	CA	USA	Single	Starbucks	College	Computer	Toyota	2008	Rent	Bicycle	12-09-1982 00:00:00	10000	Bank of America	Verizon	92
9251110008	F	29	Microsoft	11	Redmond	CA	USA	Single	Peet's Coffee and Tea	College	Computer	Toyota	2008	Rent	Bicycle	12-10-1981 00:00:00	10000	Bank of America	Verizon	92
9251110009	M	30	Nokia	12	Los Angeles	CA	USA	Single	Folger	College	Computer	Toyota	2008	Rent	Bicycle	12-11-1980 00:00:00	10000	Bank of America	Verizon	92

Note: to preserve data integrity, there are no data manipulation functionalities provided in the **Set Preview** screen including no mechanism to delete row.

2.12 Simulate Application

Simulation screen enables testing of applications without using the active channel to the mobile handset. Simulation can be performed on interactive or event applications by using the "Interactive Application" and "Event Application" tabs, respectively, as shown on the screen below.

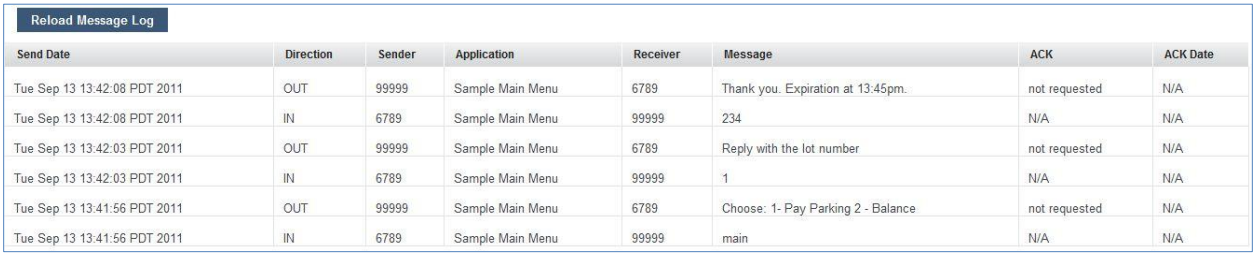


CAUTION: when performing simulation in the workspace with the active channel, the message will be sent to that channel even when the customer MSISDN is a dummy one. For example, if the simulation is performed in the workspace with real SMPP connection and the customer MSISDN is the real one, the response message will be displayed on the message log but it will also be sent out to the corresponding mobile handset. So, please use it with caution in the workspace with live channel(s).

2.12.1 Simulate Interactive Application

Interactive applications are invoked using a short or long code, a keyword, and the customer phone number (MSISDN). The simulation works for any in-service applications in the current workspace. The short or long codes provisioned for the workspace need to be used. The choice of which short or long codes to use is not really important for the purpose of simulation. In real scenario, the customer MSISDN is provided automatically by the mobile handset. In simulation, it needs to be provided manually but it could be any numbers as long as the same number is used throughout the entire simulation. The MSISDN is used by the processing engine to create session context for the interaction. The Message Text field is used to enter the keyword to start the interaction, and any subsequent reply texts during the interaction. To send the message, click the “Send to Brand Mobiliser” button.

All messages including those send to the application and responded by the application are displayed on the interaction table (also known as Message Log) with direction of IN and OUT, respectively. Sometimes, the response message arrives after the page has been refreshed. In this case, wait few seconds then click the “Reload Message Log” button to refresh the interaction table. The following screen shows the interaction with parking payment application.



Send Date	Direction	Sender	Application	Receiver	Message	ACK	ACK Date
Tue Sep 13 13:42:08 PDT 2011	OUT	99999	Sample Main Menu	6789	Thank you. Expiration at 13:45pm.	not requested	N/A
Tue Sep 13 13:42:08 PDT 2011	IN	6789	Sample Main Menu	99999	234	N/A	N/A
Tue Sep 13 13:42:03 PDT 2011	OUT	99999	Sample Main Menu	6789	Reply with the lot number	not requested	N/A
Tue Sep 13 13:42:03 PDT 2011	IN	6789	Sample Main Menu	99999	1	N/A	N/A
Tue Sep 13 13:41:56 PDT 2011	OUT	99999	Sample Main Menu	6789	Choose: 1- Pay Parking 2 - Balance	not requested	N/A
Tue Sep 13 13:41:56 PDT 2011	IN	6789	Sample Main Menu	99999	main	N/A	N/A

There are two additional buttons: “Send to Customer” and “Send to Customer with Ack” that sends message to the customer. The latter also requests acknowledgement from the SMS gateway. This functionality is not very useful in the simulation but it is used in production to test the active outbound channel. To use this functionality, the real customer MSISDN, message encoding and message text need to be provided. The message sent out will also be displayed on the interaction table.

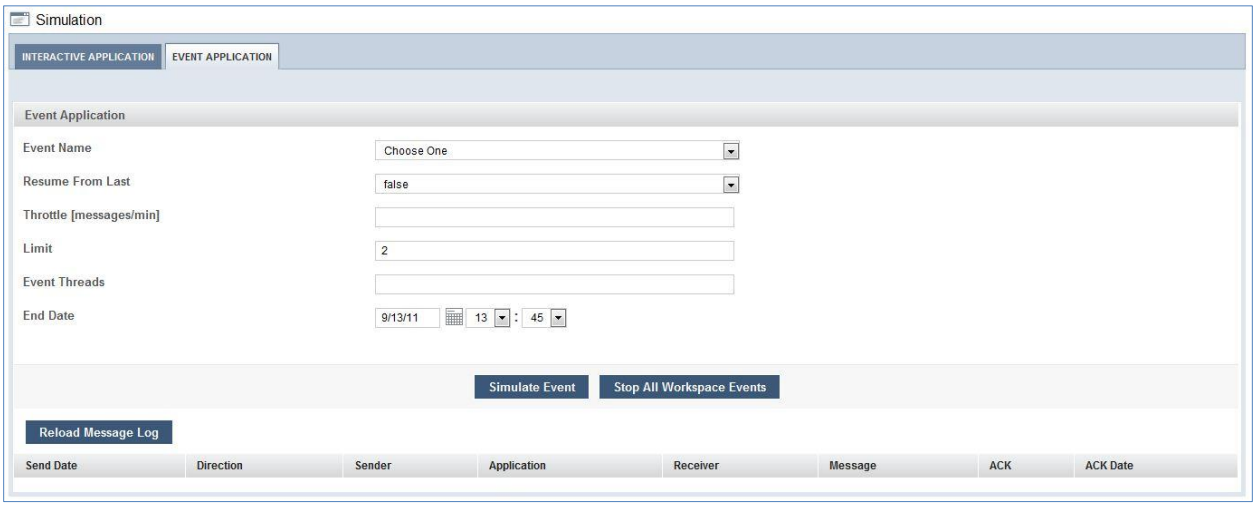
Step By Step Example

An example to illustrate how to setup and simulate an interactive application is presented on the following.

- Start by creating an interactive application. Any interactive applications that need to be initiated by incoming message need to have keyword(s). Some interactive applications are the "goto" applications so they do not need keyword. Compose the application based on the business requirements.
- When completed, activate the application. Make sure the Active From date is set prior to the current date, because only the in-service application can be simulated. At this point, the application should be in-service ready for simulation.
- On the simulation screen, go to the Interactive Application tab. The customer MSISDN is a required field but any number should work, as long as the same number is used throughout the entire interactive simulation.
- Choose any entries from the Workspace Short | Long Code field because it is a required field but the choice is not important for the purpose of the simulation.
- The default message encoding is Normal but choose Unicode if necessary.
- Enter a keyword assigned to the application earlier in the Message Text field. If there are more than one keywords are assigned to the application, choose any one.
- Next, click the "Send to Brand Mobiliser" button to start the simulation. A minimum of two new entries (IN and OUT) should appear in the message log (interaction) table. If there is only one entry (i.e., IN) appears on the table then click the "Reload Message Log" button to refresh the table. In some cases when the application or system is slow, wait few seconds before clicking again to refresh the table and the OUT message should appear. It is also possible that more one OUT messages appear on the table. This occurs when the response message generated by the application is longer than 160 characters, but the application uses the "Send SMS" state that limits the message length to 160 characters. In this case, a multiple number of messages are generated and sent resulting in multiple OUT messages.
- Typically at the beginning stage of an interactive application, the response will be a request for additional information. In this case, provide the requested information on the Message Text field. All other fields should still have the value entered earlier. If the values has been removed for some reasons, they need to be re-populated. It is essential to use the same customer MSISDN, otherwise the interaction will restart from the beginning with a new session. Click the "Send to Brand Mobiliser" to submit the requested information.
- Again new entries (IN and OUT) will appear in the tables. Continue the interactions until the end of the application. NOTE: if you need to temporarily navigate away from the Simulation screen for some reasons, make sure you note the customer MSISDN and the last OUT message because you need them to continue the interaction when returning to the **Simulation** screen.

2.12.2 Simulate Event Application

Event applications are triggered by events that are configured using the Brand Mobiliser Event based on schedule time(s). The simulation screen (event application tab) provides the mechanism to trigger the configured brand mobiliser event immediately, by submitting the event window settings. Any in-service events listed on the **Events** screen (Event Name dropdown field) can be simulated. Since the event windows are by passed, the three control parameters are made available from the simulation screen. These control parameters work exactly as in the event window setup. Please refer to the "Event Windows" section for details. Note: the limit is defaulted to 2 so that it does not process the complete list specified in the event application.



The Event Threads parameter is an advanced feature that is currently not very useful in the simulation. It creates multi-processing based on the specified number of threads.

The End Date allows specifying a specific time for the simulation to stop, in case the event application is processing a large subscriber list.

Step By Step Example

An example to illustrate how to setup and simulate an event application is presented on the following.

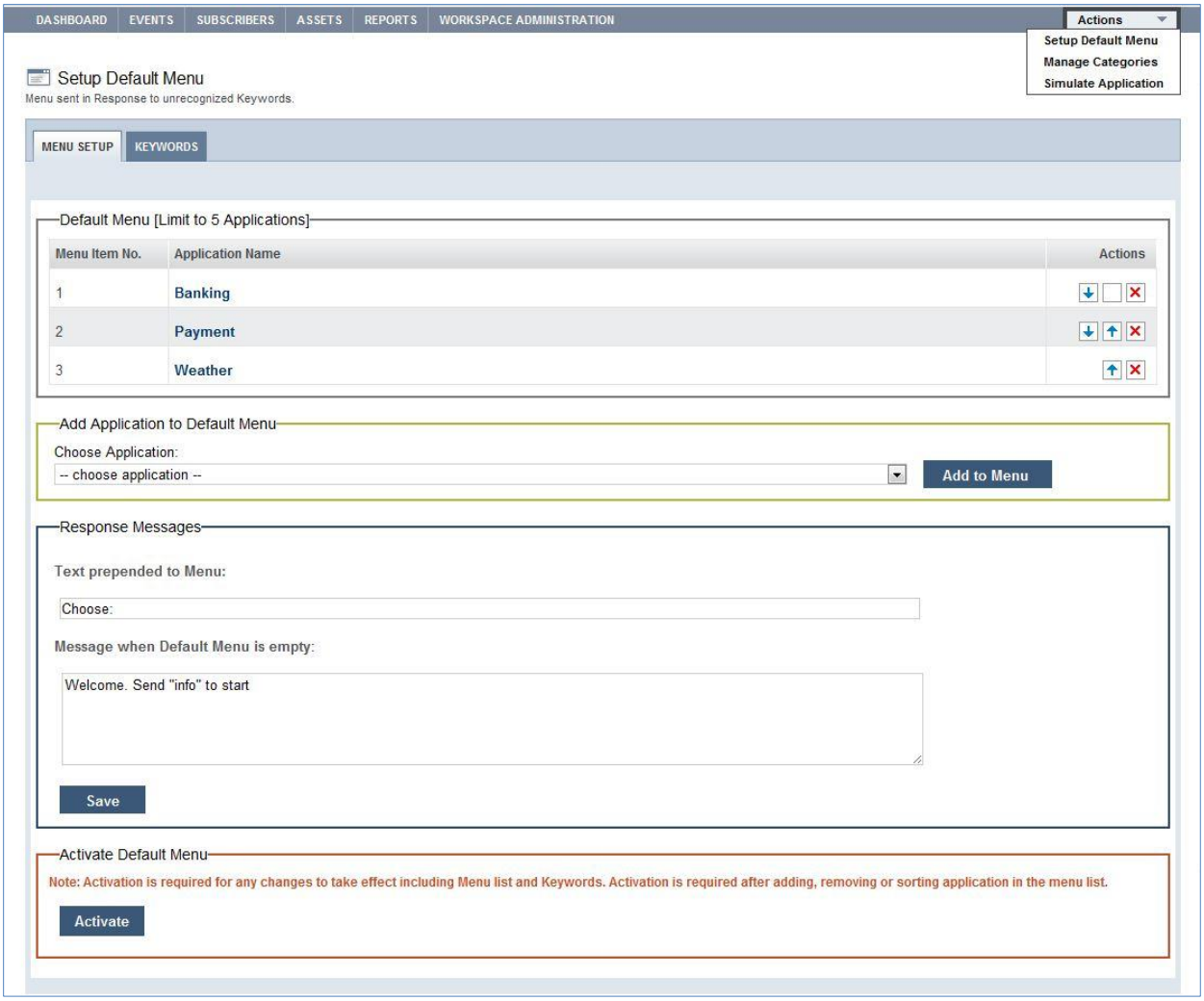
- Start by creating an event application. Compose the application based on the business requirements.
- When completed, activate the application. Make sure the Active From date is set prior to the current date, because only the in-service application can be assigned to an event. Unlike simulating application, we need to create an event and assign the event application to the event before it can be simulated.
- Go to the **Events** screen, and create a new event using the Create New Event link. See "Create New Event" section for details. Assign an event application to the event. Since we are simulating the event application, it is not necessary to specify the event windows. These settings will be provided from the event simulation screen. After activating the event, it is ready for simulation. Make sure the Event Runtime From date (on the Event Details tab) is set prior to the current date, because only the in-service event can be simulated.
- On the simulation screen, go to the Event Application tab. Choose the event to simulate based on the Event Name. Only in-service event will be displayed in the list. Keep the default control parameters until you are comfortable with the simulation. For best practice, always specify the End Date to make sure that the application does not overrun. Click the "Simulate Event" button to start the simulation. **NOTE**: in the event that you need to stop the simulation, click the "Stop All Workspace Events" button. You should be aware that this action will stop all the running events in the workspace. So, for example, if there are other users login in to the same workspace and are running simulation, their simulations will also be impacted. Or, if the workspace is used in production and there are scheduled events running, those events will also be terminated.
- The simulation results will appear in the message log table.

2.13 Default Menu

The Default Menu is a workspace specific settings, so it needs to be setup on each workspace when needed. Typically, the setup of default menu is performed by the workspace administrator. Other roles will not have access to the **Setup Default Menu** screen, accessible from the Actions – Setup Default Menu.

The purpose of the Default Menu is to catch all un-recognized keywords that are sent to the workspace (or rather the configured short or long code), and to respond with some guidance. The reason it's called "Default Menu" is because traditionally the response guidance is in the form of a menu. For example, the response message for all un-recognized keywords can be as follows:

Please send: bank – for banking, payment – for payment, weather – for weather



The Default Menu takes it a step further by allowing the setup to specify the application(s) and the menu will be generated automatically when requested. Referring to the screen above, the menu has been setup to link to three applications: Banking, Payment and Weather. So, when an un-recognized keyword is sent to the workspace (for example, unknown), the response sent to the mobile customer will be generated automatically using the default menu settings. The menu index is generated automatically, and in addition, the mobile subscriber can response with the menu index of interest minimizing typing, as shown on the following figure (simulated mobile phone).

There are four functionalities presented on the **Setup Default Menu** screen: Default Menu, Add Application to Default Menu, Response Messages, and Activate default Menu. They are described in details on the following.

The "Default Menu" shows the menu setup that will be sent in response to un-recognized keyword. The menu order is the basis for the generated menu indexes. The order can be changed using the or action buttons to move the application order down or up, respectively. The application can be removed from the menu by clicking on the action button. The menu is limited to five applications. The application name is a linked when clicked will navigate to the Application screen showing the corresponding application.

The "Add Application to Default Menu" provides a list of all the applications that can be added to the default menu. The list shows all the interactive applications that are currently in-service. To add an application to the menu, select it from the list and click the "Add to Menu" button. The selected application will appear as the last item on the default menu. When there are five applications on the default menu, the "Add to Menu" button will be disabled.

The "Response Messages" section shows two settings: the text that will be pre-pended to the default menu, and the response message when the default menu is empty. The pre-pended text can be as simple as "Choose:" or a welcome message. When there are no applications assigned to the default menu, an alternative response message, from the "Message when the Default Menu is empty", will be sent to the mobile subscriber. For example, the setup on the above screen advises the mobile subscriber the valid keyword (i.e., info).

Like the application, the default menu also requires activation. When a new workspace is created, the default menu needs to be activated. Any subsequent changes made on the **Setup Default Menu** screen require re-activation before the changes become live. The "Activate Default Menu" section provides the "Activate" button that is enabled when changes have been made and activation is required.

2.14 Reports

The **Reports** main screen is accessible from the navigation bar. The main screen is the dashboard to all available reporting modules. Currently, there is only one "Standard Reports" module. The screen layout is based on the module layout that enables future extension through plug in modules.

The Standard Reports module provides links to Traffic Report and Subscriber Report.

The Traffic Report link navigates to the **Traffic Report** screen.

The Subscriber Report link navigates to the **Subscribers** screen.

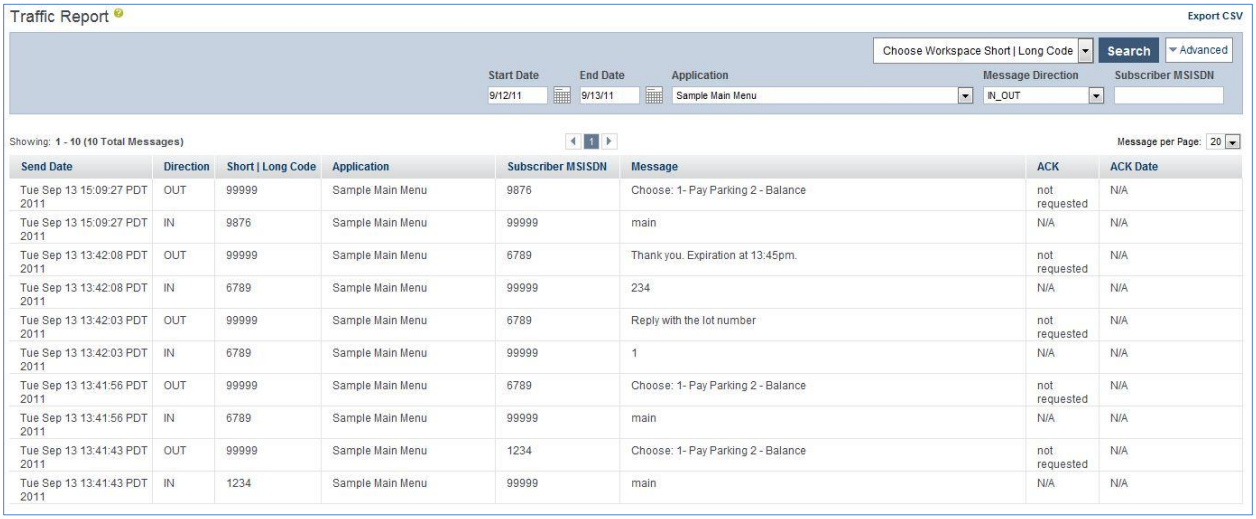
In the current version, there is no graphical report presented in Brand Mobiliser. However, the report screens provide an export capability so that the data can be exported to the CSV file and imported into a more sophisticated reporting tools or applications, or import them into the system of record.

2.14.1 Traffic Report

The **Traffic Report** screen provides a simple search and filter tools to extract traffic report. The traffic report refers to the incoming and outgoing messages, as well as the acknowledgement messages (if returns by the SMS provider). The **Traffic Report** screen is not designed for analyzing the messages. It's merely a tool for filtering the messages and show a preview of the filter results. The preview results are limited to 2000 rows. The traffic report can be exported to a CSV file and can easily be read or imported by other sophisticated analysis and reporting tools.

The filter parameters include: Short or Long code, Start and End Dates, Application Name, Message Direction and Subscriber MSISDN. The short or long code filter narrows the message to a specific short or long code. This filter is mainly useful for interactive applications. The Start and End date narrow the results by the message send date. The Application Name refers to the interactive or event application name. The filter choice for Message Direction is: in, out, or in & out, referring to incoming messages to Brand, outgoing messages sent by Brand, or both, respectively. The Subscriber MSISDN filter can be used to narrow the message to a specific customer based on the MSISDN. This is helpful for those customers that use the interactive applications.

The results presented on the screen is limited to 2000 rows. However, when the results are exported to CSV file the complete results will be available. The Export CSV link is only visible when there are results presented on the screen. When exporting to CSV, the same filter criteria will be used to generate the results. However, the exported results are not limited to 2000 rows.



Send Date	Direction	Short | Long Code	Application	Subscriber MSISDN	Message	ACK	ACK Date
Tue Sep 13 15:09:27 PDT 2011	OUT	99999	Sample Main Menu	9876	Choose: 1- Pay Parking 2 - Balance	not requested	N/A
Tue Sep 13 15:09:27 PDT 2011	IN	9876	Sample Main Menu	99999	main	N/A	N/A
Tue Sep 13 13:42:08 PDT 2011	OUT	99999	Sample Main Menu	6789	Thank you. Expiration at 13:45pm.	not requested	N/A
Tue Sep 13 13:42:08 PDT 2011	IN	6789	Sample Main Menu	99999	234	N/A	N/A
Tue Sep 13 13:42:03 PDT 2011	OUT	99999	Sample Main Menu	6789	Reply with the lot number	not requested	N/A
Tue Sep 13 13:42:03 PDT 2011	IN	6789	Sample Main Menu	99999	1	N/A	N/A
Tue Sep 13 13:41:56 PDT 2011	OUT	99999	Sample Main Menu	6789	Choose: 1- Pay Parking 2 - Balance	not requested	N/A
Tue Sep 13 13:41:56 PDT 2011	IN	6789	Sample Main Menu	99999	main	N/A	N/A
Tue Sep 13 13:41:43 PDT 2011	OUT	99999	Sample Main Menu	1234	Choose: 1- Pay Parking 2 - Balance	not requested	N/A
Tue Sep 13 13:41:43 PDT 2011	IN	1234	Sample Main Menu	99999	main	N/A	N/A

2.15 Workspace Administration

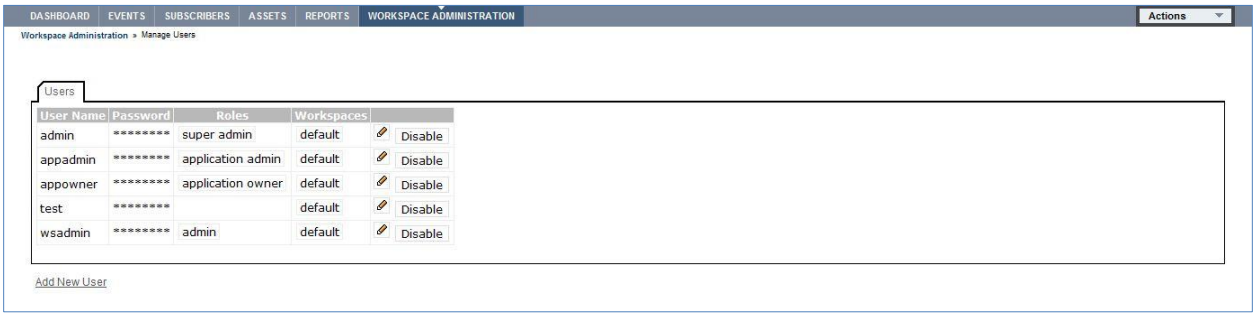
2.15.1 Workspace Administration

The **Workspace Administration** screen is accessible from the navigation bar. The main screen is the dashboard to administration and management modules. Currently, there is only one module, “Workspace Administration”, as shown on the following screen. The screen layout is based on the module layout that enables future extension through plug in modules.

The current workspace administration module provides tools for managing workspaces, users, channels and sessions. Each of these tools will be elaborated later in a separate sections below. Access to the functions is determined by roles. The Manage Workspaces and Manage Channel Configurations are available to the platform administrator only.

2.15.2 Workspace Administration - Manage Users

The **Manage Users** screen presents the users in the current workspace that the login administrator is in, as displayed on the Login Status Bar – Workspace. The login administrator may have access to multiple workspaces, as listed on the Login Status Bar – Workspace dropdown, but only the users in the current workspace will be displayed. Remember, workspace is a logical grouping of applications, users and other artifacts as discussed in “Workspace” section.



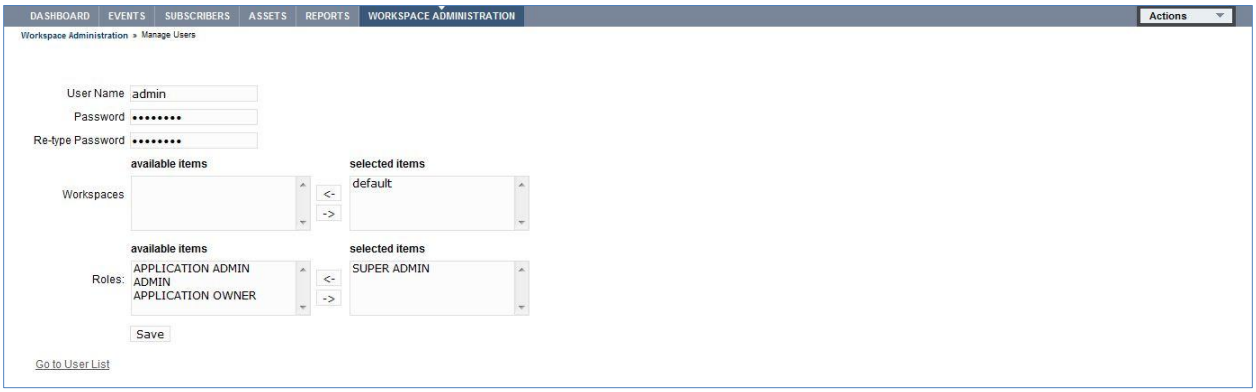
Functionalities include:

	Edit the user on the corresponding row
Disable	Disabled the user on the corresponding row. For audit trail reasons, the delete user function is not provided. However, to revoke the user from accessing the application completely the user can be disabled. For enhanced security, when the disabled user attempts to login the feedback message is the

	generic “Invalid username/password”.
Add New User	hyperlink that navigates to the **New/Modify User** screen discussed in the following section

NOTE: let’s use an example to better illustrate and emphasize the behavior of the filtered user list, since most first time user administrators will be puzzled by it especially after assigning or changing the user’s workspace. Typically, the user administrators will be assigned to multiple workspaces. Adding or managing users can be performed from any workspaces. Let’s say that the administrator has access to the following workspaces: default, ws1, and ws2, currently is in the “default” workspace. The administrator creates a new user using the **New/Modify User** screen, and assigns the user to “ws2”. After successful saving, the screen is navigated back to the **Manage Users** screen showing the users list. However, the newly created user is not shown on the list because the administrator is in the “default” workspace. The administrator needs to switch to the “ws2” workspace using the workspace dropdown list on the Login Status Bar. Switching workspace will navigate the screen to the **Dashboard**. When the administrator navigates back to the **Manage Users** screen, a different users list will be displayed with the newly created user in the list. The same behavior will occur when the administrator re-assign an existing user to a different workspace.

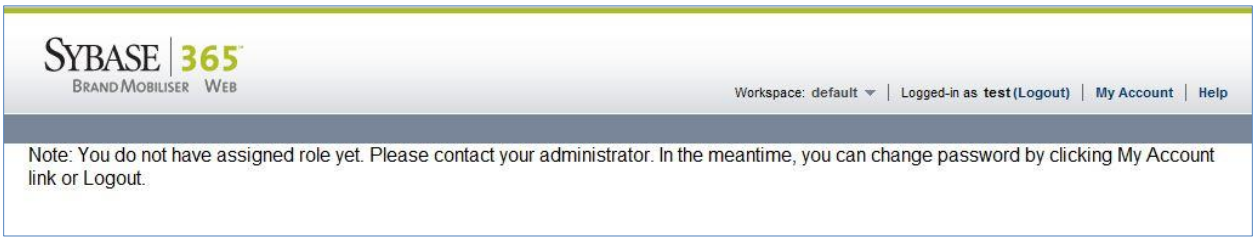
2.15.3 Workspace Administration - Manage Users - New/Modify User



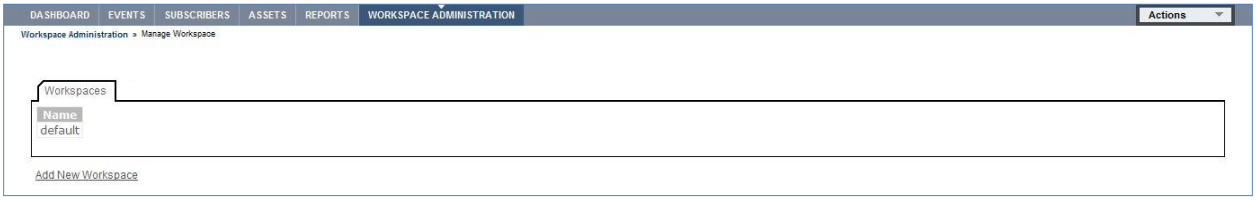
The **Manager User** screen allows creation or modification of a user. Users must have a password associated with their unique Username and requires the assignment of a workspace. If no workspace is assigned to the user, the “default” workspace will automatically be assigned by default during saving. Workspaces are explained in the next section. Roles are explained in “Appendix B”. Saved changes will take effect after the user re-login. For example, after assigning new workspace(s) to an existing user, the user needs to re-login to be able see the newly assigned workspace(s).

When the user with no assigned roles login, the following screen will be presented. There are two enabled functions: My Account, and Logout.

The recommended best practice is to create a new user with no role assignment and default workspace. See recommended best practice for workspace in the “Workspace” section. Then, communicate to the user the newly created user account, in this case the username and generic password. The user needs to login with the generic password and then go to My Account screen to change the password, which essentially means the user takes ownership and responsibility of the account. At this point, the administrator who creates the account will not have access to the account. After getting manager approval (or any corporate approval policy), the administrator can assign appropriate role(s) and workspace(s) to the user. For extra security and enforcing IT policy, the administrator should first ensure that the user has changes the password before assigning a role and workspace.



2.15.4 Workspace Administration - Manage Workspaces

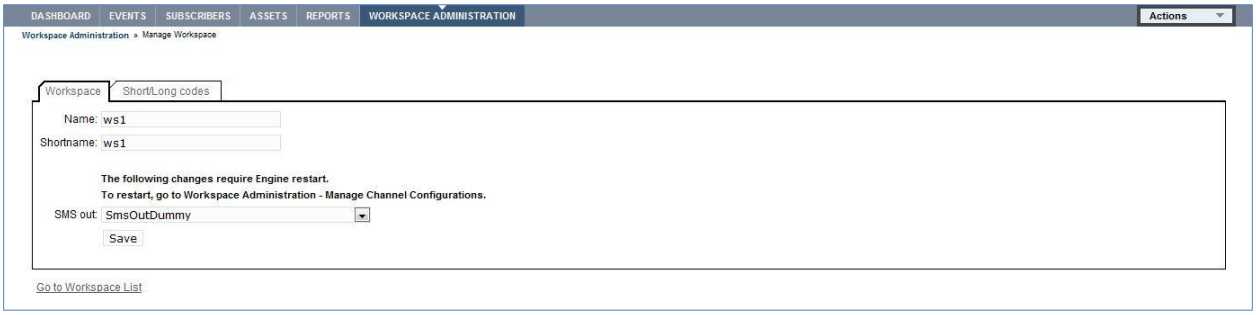


Manage Workspace screen shows a list of the current workspaces in the platform. The workspace name is a hyperlink that navigates to the **Workspace Configurations** screen of the corresponding workspace, discussed in the next section. The "Add New Workspace" link, displayed below the list, also navigates to the **Workspace Configurations** screen but all the fields are empty for configuring new workspace.

Workspace is a logical grouping of applications, users and other artifacts. In addition to the unique name, workspace also has unique short or long code(s), as discussed in the "Workspace" section. The unique short or long code(s) are used by the processing engine to dispatch the incoming messages to an application in the corresponding workspace. Workspace can also be used for separating the development, QA and production environments, in the low traffic volume production environment.

2.15.5 Workspace Administration - Manage Workspace - Workspace Configurations

The **Workspace Configurations** screen allows creation and modification of a workspace. The name and shortname can be the same but they need to be unique in the system.

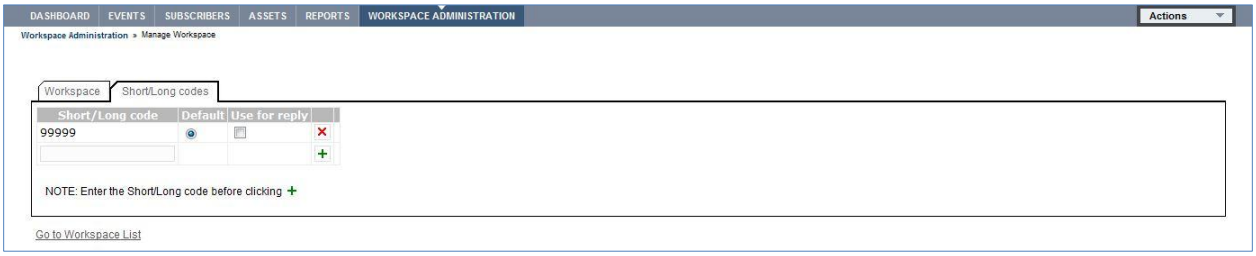


Workspace has a unique outbound channel. For further discussions on inbound and outbound channels, please refer to the "Workspace Administration - Manage Channel Configurations" section. The outbound channel is specified in the SMS out field by selecting it from the available list. The list is gathered from the active channels configured on the platform. By default, a dummy outbound channel (SMSOutDummy) is assigned to a newly created workspace. Any changes to the outbound channel will require a restart in the processing engine.

To be able to receive incoming message, workspace need to have unique Short or Long code(s). There are differences between short code or long code but for the purpose of Brand Mobiliser they are the same, just a unique identifier. Therefore, they will be referred to as just short code in this section. The short code is the unique

workspace identifier for the Brand Mobiliser processing engine, and it's the identifier used to associate an incoming message to a specific workspace.

The Short/Long codes tab allows adding and removal of the codes to uniquely determine how to dispatch an incoming message. Once a short code has been added to a workspace, then all incoming messages with the destination MSISDN equal to that short code will be dispatched to the corresponding workspace. In addition, any outbound messages leaving the workspace generated in response to the inbound message (interactive application) will have the "origination MSISDN" set to the same short code. **However**, if there are more than one short codes assigned to a workspace, then the "Default" and "Use for reply" flags will be used in determining which short code to assign to the "origination MSISDN" of the outgoing message. For example, if the workspace has short codes: A, B and C. Short code "A" has the "Default" flag checked and Short code "B" has the "Use for reply" flag checked; Short code "C" has no flags checked.



Here is what happens to the outbound message sent in response to the inbound message sent to each short codes.

- If the inbound message came in on "B", then the outbound response message will have the origination set to "B" since the "Use for reply" is checked.
- If the inbound message came in on "C", then the outbound response message will have the origination set to "A". As "C" does not have the "Use for reply" flag checked, the system selects the "Default" Short code to use.
- If the inbound message came in on "A", then the outbound response message will have the origination set to "A" since it is the "Default" Short code.

For event application on the other hand, since there is no inbound message, the "Default" short code is ALWAYS used as the origination MSISDN for any outbound messages.

2.15.6 Workspace Administration - Manage Channel Configurations

Manage Channel Configurations screen is available to the platform administrator because the impact of any changes will be platform-wide. Channel is the conduit to deliver inbound or outbound messages to Brand Mobiliser. There are three built-in channels: SmsOutDummy, JMS and SMPP.

The SmsOutDummy is the loopback channel and it is used for simulation test in the development environment. Therefore, there is no configurations needed for SmsOutDummy. When a new workspace is created, the SMS out channel is assigned to SmsOutDummy automatically.

The JMS (Java Messaging Service) channel delivers inbound and outbound via the Message Oriented Middleware. The JMS channel is commonly used to integrate with the external USSD server.

The SMPP (Short Message Peer-to-Peer) channel delivers inbound and outbound using the SMPP protocol to short message service centers (SMSC) and/or External Short Messaging Entities (e.g., Sybase 365 Hub).

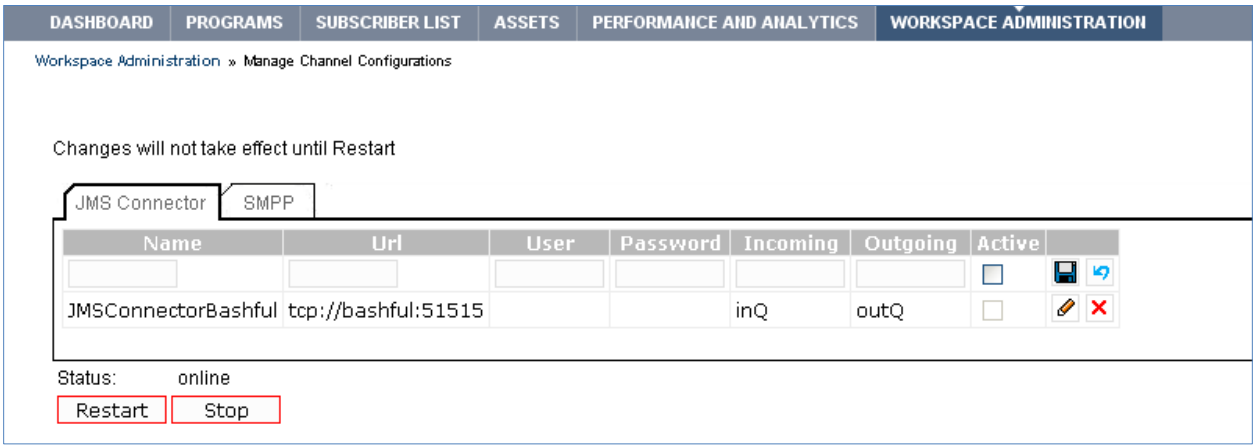
The **Message Channel Configurations** screen are used to configure the JMS and SMPP channels, using the "JMS Connector" tab and the "SMPP" tab, respectively. Detailed discussions on how to configure the channels are provided in a separate sections below.

The followings are common features to the channel configuration screen.

The first empty row is a placeholder for creating new channel. To create a new channel, enter the configurations in the empty row and click the save button when done.

Active	Channels can be added to the channel list but the channels will not be active until the Active column checkbox is checked, saved, and the processing engine is restarted. NOTE: There are no limits to the number of active channels but the active channels consume the system resources.
Status: online Restart Stop	Any changes to the channel configurations including setting the Active checkbox will require restarting of the processing engine. The processing engine control panel is shown below the channel list. The Restart button will stop and restart the engine. The Stop button will stop the engine and requires manual start.
	Edit/modify channel configurations
	Save changes
	Cancel any changes
	Delete the channel

2.15.6.1 JMS Channel



The JMS channel provides a bridge from Brand Mobiliser to send/receive messages using JMS messages. It has been tested with ActiveMQ and a single configuration supports both inbound and outbound traffic. The queue names of the JMS channel dictates whether the message on the queue is an inbound or outbound message.

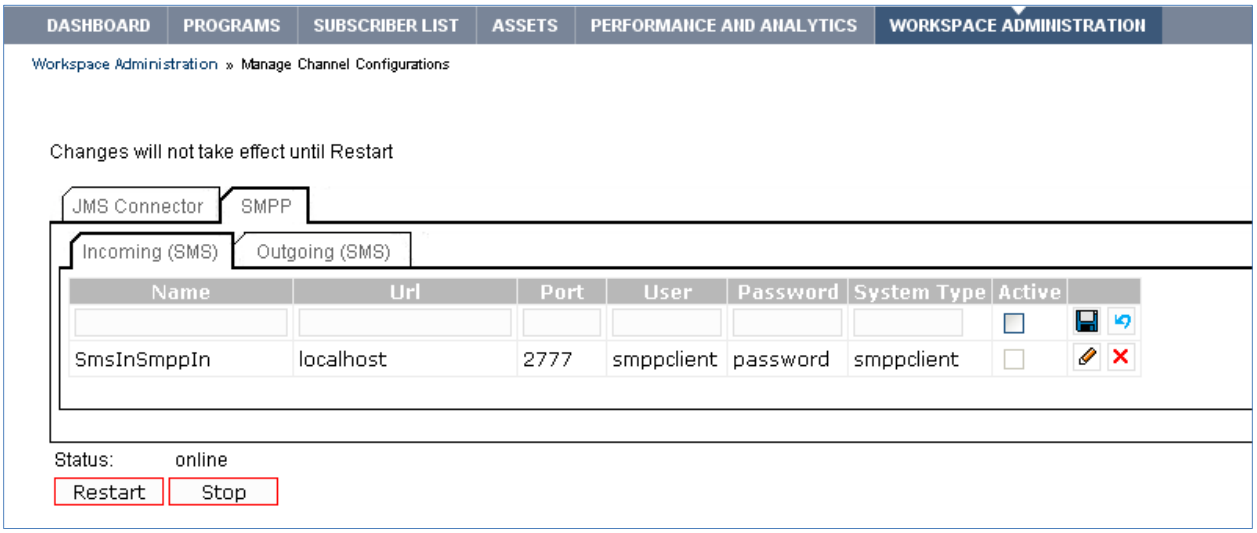
The message format used by Brand Mobiliser is proprietary. Please consult with your Sybase Support contact if you wish to use this particular channel mechanism.

The following are the allowable parameters for each JMS connection.

Parameter	Description
Name	Unique name to allow system to refer to this channel.
Url	JMS broker to connect to.
User	User name to authenticate with.
Password	Password to authenticate with.

Incoming	Queue name to look for inbound messages.
Outgoing	Queue name to send outbound messages.
Active	Whether this channel is active or not.

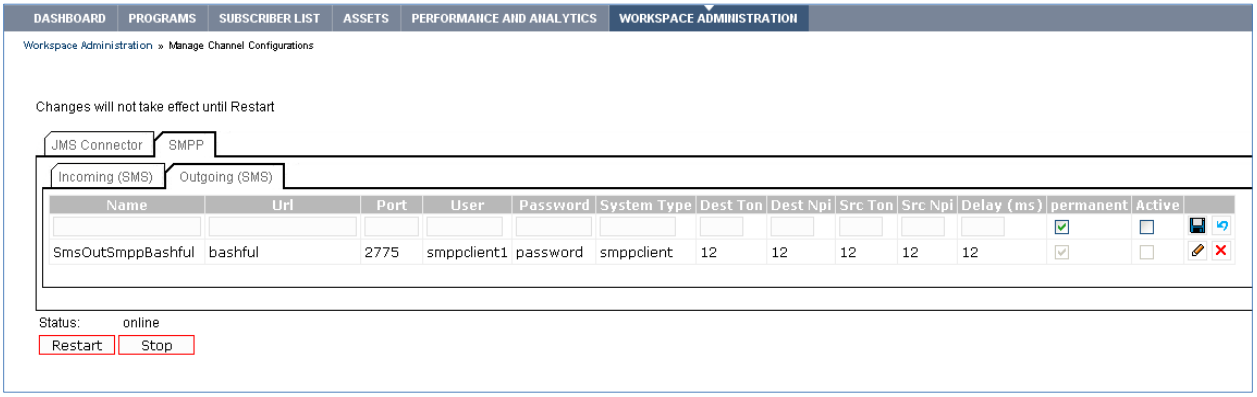
2.15.6.2 SMPP Channel



The SMPP inbound connection is active for all workspace in the system. Use the pencil image to edit the different parameters. After saving the changes, a system restart using the "Restart" button must occur before changes take effect. The following are the allowable parameters for each SMPP inbound connection.

Parameter	Description
Name	Unique name to allow system to refer to this channel.
Url	Host name of the SMSC.
Port	Port number to listen.
User	User name to authenticate with.
Password	Password to authenticate with.
System Type	Identifier for SMSC.
Active	Whether this channel is active or not.

The SMPP outbound connection is active for all clients that have explicitly configured to use the connection. Use the pencil image to edit the different parameters. After saving the changes, a system restart using the "Restart" button must occur before changes take effect.

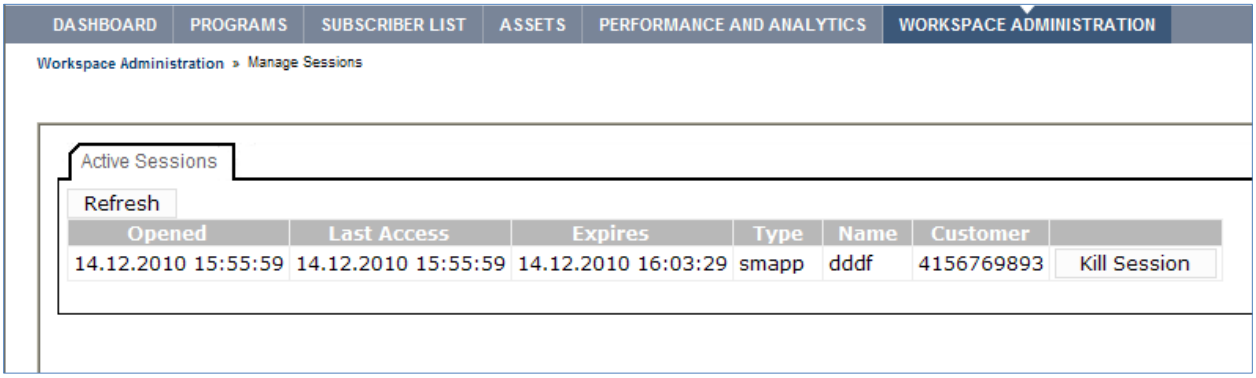


The following are the allowable parameters for each SMPP outbound connection.

Parameter	Description
Name	Unique name to allow system to refer to this channel.
Url	Host name of the SMSC.
Port	Port number to create connection to.
User	User name to authenticate with.
Password	Password to authenticate with.
System Type	Identifier for SMSC.
Dest Ton	Destination Type of Number
Dest Npi	Destination Numbering Plan Identification
Src Ton	Source Type of Number
Src Npi	Source Numbering Plan Identification
Delay(ms)	Message delay before sending next message.
Permanent	Whether connection is persistent or a new connection is made for every message.
Active	Whether this channel is active or not.

Please consult with your SMSC partner to configure the Src/Dest Ton/Npi values.

2.15.7 Workspace Administration - Manage Sessions



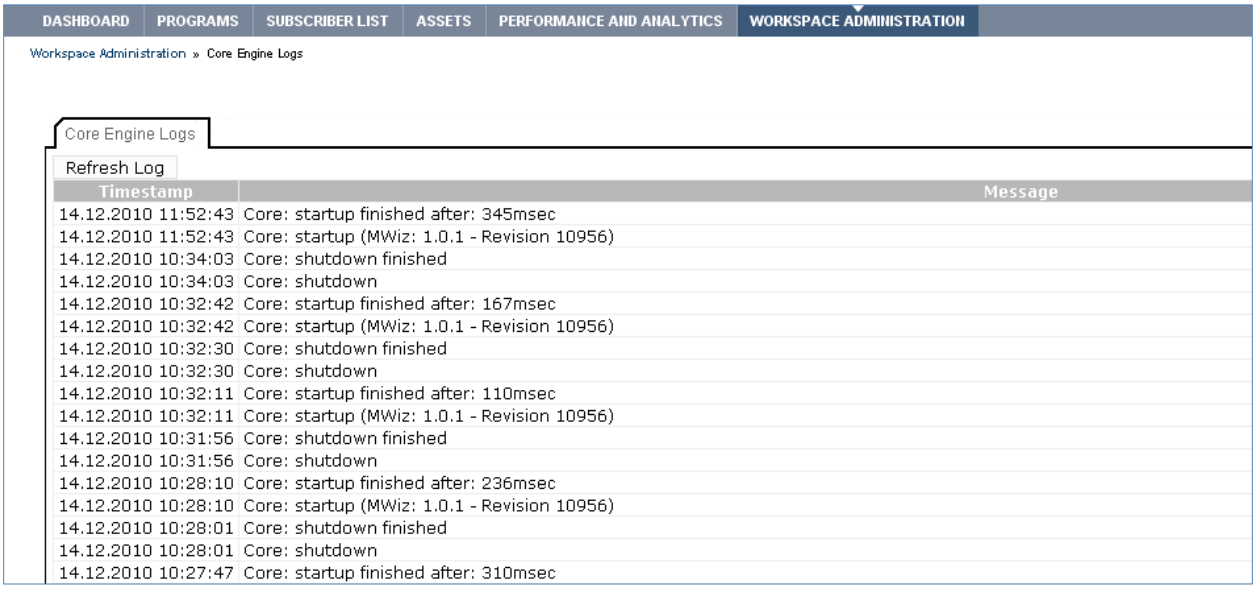
Manage Sessions screen provides the tool to monitor active sessions and terminate them if necessary. As described earlier, the Brand Mobiliser processing engine creates a session for each incoming messages. These sessions are displayed on the **Manage Sessions** screen while they are still active. The refresh button pulls the latest session. Terminating a single active session using the “Kill” button. Active sessions are automatically terminated when the application ends or if the session timeout period has elapsed, and they will not show on the list.

Manage Sessions screen is available to the workspace administrators. In the current implementation, the active session list include all sessions including those from other workspace. Therefore, **caution should be used in stopping or killing a session.**

2.15.8 Workspace Administration - Core Engine Log

Display various informational messages during the startup/shutdown of the Brand Mobiliser engine.

The refresh button pulls back the latest messages logged from the database.



2.15.9 Workspace Administration – Traffic Reports

Traffic Report links to the Traffic Report screen. Please refer to "Traffic Reports" section for details.

3 Installation and Configuration

Brand Mobiliser supports two types of deployment models: Stand Alone and J2EE Web. The Stand Alone deployment (the preferred model) is a self contained Brand Mobiliser running in the Sybase AIMS Server. The J2EE web deployment (or war-file) requires a J2EE servlet container like the Apache Tomcat server, and it is only made available to customer on special request.

This section describes the process of installing Brand Mobiliser in the Stand Alone deployment model. Please review the "Prerequisites" section below prior to installation. Installation and a sample walkthrough with configuration instructions are provided in the following sections.

3.1 Prerequisites

Prerequisites for Brand Mobiliser are as follows. The version number indicates what was used in testing and certification. Other versions may also work but need further testing and certification before deploying it into a production environment.

Software	Vendor	Version
Platform	Redhat Linux	5.5 64-bit
	IBM AIX	6.1.0.6 [JRE 1.6.0 IBM J9 2.4 AIX ppc64-64]
JDK		1.6.0+
Database	Oracle	10g and 11g
	IBM DB2	9.7.2
State Plug-in	Sybase Money Mobiliser	4.5
SMPP Protocol		3.4
J2EE Servlet Container	Apache Tomcat	6.0.32
	IBM Websphere AS	7.0.0.11
Web Browser	Firefox	3.6+
	Chrome	7+
	Internet Explorer	8+

3.2 Software Binary

The Brand Mobiliser distribution binary is a zip file with name "aims-brand-mobiliser-*versionNumber*.zip". The *versionNumber* is the version number of Brand Mobiliser. For example, for Brand Mobiliser Version 1.1.1 the filename is "aims-brand-mobiliser-1.1.1.zip". Copy the file from the distribution medium to the installation location. For the purposes of this document, we assume "/opt" on a Linux server.

3.2.1 UnPackaging

Starting from `/opt` folder

```
unzip aims-brand-mobiliser-versionNumber.zip
```

A directory is created at `/opt/aims-brand-mobiliser-versionNumber` with these contents:

```
bin/
bundle/
conf/
```

```
license/
sql/
README.html
run.bat
run.sh
```

File/Directory	Contents
bin/	Apache Felix kernel and main subsystem bundles.
bundle/	OSGi bundles that make up Brand Mobiliser
conf/	Configuration files for Brand Mobiliser
license/	Open Source License information
README.html	Included documentation on the contents of the installation
run.sh/run.bat	Startup scripts for Brand Mobiliser, linux and Windows scripts, respectively
sql	Database scripts for Brand Mobiliser

3.3 Database Scripts Installation

Current supported databases are: Oracle and IBM DB2. Please note the certified version number listed in the Prerequisites section. Other version numbers requires testing and certification.

Note: please consult your DBA prior to running the database scripts to ensure compliance with the company database policies.

There are two script versions provided: fresh install scripts, and upgrade scripts, and they are saved in the "1.2" and "Upgrade_1.1" folders, respectively.

3.3.1 Oracle Installation – Fresh Install

The SQL scripts are in `sql/oracle` folder for your Oracle installation. There are four script files available for a fresh install. Note: Please consult with your DBA for a custom installation using an existing "tablespace" or user for the Brand Mobiliser schema.

01-BrandMobiliser-Tablespaces.sql

02-BrandMobiliser-Users.sql

03-BrandMobiliser-Objects.sql

04-BrandMobiliser-Base-Data.sql

Note: the first two scripts require DBA privileges in order to create a new tablespace and users for the mwiz2 user. The last two scripts need to be run using the mwiz2 user that was just created using script 02.

Note: The mwiz2 user will be used in the JDBC connection.

A sample installation of the SQL scripts on Oracle 10g follows:

1. Create Tablespace.

```
sh-4.1$  sqlplus sys/ as sysdba  @01-BrandMobiliser-Tablespaces.sql
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 13 15:42:05 2010
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

Tablespace created.
SQL> exit
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
```

2. Create Brand Mobiliser User.

```
sh-4.1$  sqlplus sys/ as sysdba  @02-BrandMobiliser-Users.sql
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 13 15:42:59 2010
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

User created.
Grant succeeded.
User altered.
User altered.
User altered.
SQL> exit
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
```

3. Create schema using mwiz2 user.

```
sh-4.1$ sqlplus mwiz2/   @03-BrandMobiliser-Objects.sql
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 13 15:48:04 2010
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

Table created.
```

```
...
Table created.
View created.

CREATING SEQUENCE SEQ_MWIZ
==========================
Sequence created.

Index created.
...
Index created.
SQL> exit
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
```

4. Insert initial data.

```
sh-4.1$ sqlplus mwiz2/   @04-BrandMobiliser-Base-Data.sql
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Dec 13 15:49:16 2010
Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

1 row created.
...
1 row created.
Commit complete.

SQL> exit
Disconnected from Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
```

The following logs are created by the SQL scripts:

BrandMobiliser_Base_Data.LOG

BrandMobiliser_Objects.LOG

BrandMobiliser_Tablespaces.LOG

BrandMobiliser_Users.LOG

Once the schema is created, the SQL scripts are no longer necessary during application runtime. Please save the scripts in a safe location for future database rebuild if needed.

3.3.2 IBM DB2 Installation – Fresh Install

The SQL scripts are in `sql/db2` folder for your IBM DB2 Version 9.7.2 installation. There are 4 script files available for a fresh install. Note: Please consult with your DBA for a custom installation using an existing tablespace or user for the Brand Mobiliser schema.

01-BrandMobiliser-Tablespaces.sql

02-BrandMobiliser-Users.sql

03-BrandMobiliser-Objects.sql

04-BrandMobiliser-Base-Data.sql

The followings system setup need to be performed prior to installing the scripts. **Note: Steps 1-3 must be run by an user with sudo privileges or as root**.

1. Add the following groups to the system

```
groupadd -g 999 db2iadm1
groupadd -g 998 db2fadm1
groupadd -g 997 dasadm1
```

2. Add the following users to the system

```
useradd -u 1004 -g db2iadm1 -m -d /home/mwiz2 mwiz2
useradd -u 1003 -g db2fadm1 -m -d /home/db2fenc1 db2fenc1
useradd -u 1002 -g dasadm1 -m -d /home/dasusr1 dasusr1
```

3. Change passwords of newly created users

```
passwd mwiz2
passwd db2fenc1
passwd dasusr1
```

Note: The mwiz2 user will be used in the JDBC connection.

4. Create a DB2 instance. Assuming that IBM DB2 is installed on the server.

```
$ pwd
/opt/IBMdb2/V9.7/instance/
$ ./db2icrt –a server –u db2fenc1 mwiz2
```

5. Verifying Installation

- Log in as mwiz2 user on the system
- Start database manager by entering db2start command

  type db2start

- Enter the db2sampl command to create the SAMPLE database

```
            type db2sampl

• Enter DB2 Command Line Processor (CLP)
            type db2

• Connect to the sample database
            type connect to sample
```

6. Create database in the DB2 instance and grant DBA privilege to mwiz2 user. Ensure that the current DB2 instance has write privilege to the data directory. The script assumes "/home/db2".

```
01-BrandMobiliser-DB_DB2.sql
```

7. Connect to newly created "brandmob" database as the mwiz2 user

```
db2 connect to brandmob user mwiz2 using sql
```

8. Run the rest of the scripts using the –vtf option to db2

```
db2 -l db2out.log -vstf "filename"

02-BrandMobiliser-DB-Tablespaces_DB2.sql

03-BrandMobiliser-Objects_DB2.sql

04-BrandMobiliser-Base-Data_DB2.sql
```

3.3.3 Upgrade from Version 1.1.1

When upgrading from Brand Mobiliser version 1.1.* to the current version 1.2.0, you need to run two script files provided in "Upgrade_1.1" folder.

NOTE: Please review the following recommendations prior to performing the upgrade:

- **As always, please consult your DBA prior to running the database upgrade scripts to ensure compliance with the company database policies.**
- **Backup the database before running the upgrade scripts.**
- **Consult with your DBA who maintains the Brand Mobiliser database to find out details on the existing installation: such as customization using an existing tablespace or user for the Brand Mobiliser schema.**

The upgrade script files are as follow:

03a-BrandMobiliser-Objects-NewVer1-2.sql

04a-BrandMobiliser-Base-Data-NewVer1.2.sql

3.4 Stand Alone Deployment

The Stand Alone deployment of Brand Mobiliser uses the Sybase AIMS server. Relative paths in the sections below use the Brand Mobiliser home location created in "/opt/aims-brand-mobiliser-*versionNumber*". Note: the j2ee folder is not used in the Stand Alone deployment.

3.4.1 Binary installation

A directory is created at `/opt/aims-brand-mobiliser-versionNumber` with these contents:

```
bin/
bundle/
conf/
j2ee/
sql/
license/
README.html
run.bat
run.sh
```

The "`run.sh`" starts Brand Mobiliser. The "`run.sh`" should have its run time permission bit set. If not, please run "`chmod 755 run.sh`" to set it.

3.4.2 Platform Configurations

Platform configuration files (`config.properties` and `system.properties`) are required before the startup of Brand Mobiliser. They are saved in the "`conf`" folder.

The "`config.properties`" file contains AIMS server configurations that is based on the Apache Felix implementation. You should not have to change most of these configurations as they pertain to the platform. Those that may need to be changed, for example when applying patches, will be discussed on the following subsections or instructions will be provided in the patch packages. The patches will also come with a step by step instruction on how to apply them to the existing deployment.

The "`system.properties`" file contains subsystem configurations, such as the logging system, the jetty HTTP server, etc.

3.4.2.1 Ports

Brand Mobiliser requires 3 ports. By default, they are configured as follows.

Port Key	Value
RMI_PORT	5366
osgi.shell.telnet.port	5365
org.osgi.service.http.port	8080

Note: RMI port must not conflict with an existing process or the server will not start.

The RMI_PORT is set in the run.sh script.

The telnet port (`osgi.shell.telnet.port`) and the Http port (`org.osgi.service.http.port`) are set in the "`conf/config.properties`" file. The telnet port is for system level administration of the AIMS server and is

locked down to only allow connections from the physical server on which the AIMS server is installed (i.e., localhost). Normal operations will not require any interaction with the telnet system console. The http port is where the Brand Mobiliser UI can be accessed. Please modify these ports if they are in conflict with existing environment.

3.4.2.2 Applying Patches

Occasionally patches are released to fix a specific or emergency problems that cannot wait until the next release. Patches are essentially bundles (or jar files) that needs to be copied to the "bundle" or "bundle/application" folders. A specific step by step instructions is always included in the patch packages. The patched bundles will always have unique file names. In most cases, the AIMS server needs to be restarted after applying the patches.

In addition to copying the bundles, the "`config.properties`" file need to be modified to reflect the new (or patch) bundles. All the bundles that are loaded into the AIMS server at startup is defined in the "`config.properties`" file using the "`felix.auto.start.`*`level`*" property. The level indicates the start level or the sequence that the bundles are loaded into the AIMS server, with the lowest level loaded first. For example, the Brand Mobiliser Web UI bundle is defined with level 20, as follow:

```
…
felix.auto.start.20 = \
 file:bundle/application/mobiliser-brand-webadmin-controls-1.2.0.jar \
 file:bundle/application/web-core-0.1.6.jar \
 file:bundle/application/web-model-api-0.1.6.jar \
 file:bundle/application/mobiliser-brand-webadmin-ui-1.2.0.war \
 file:bundle/application/mobiliser-brand-webadmin-redirector-1.2.0.war
…
```

3.4.2.3 Enabling SSL

The following instructions are for enabling Jetty SSL:

1. Create a keystore if one doesn't exist.

   ```
   keytool -keystore keystore -alias jetty -genkey -keyalg RSA
   ```

   Follow the on screen instructions. The only required information is the First/Last Name Set the Name to match your machine hostname. For example. set to "dolphin" with no DNS domain name. The "keystore" file created should be copied/moved to "conf/keystore" in the Brand Mobiliser directory.

   For more information about importing existing SSL keys, see:

   http://docs.codehaus.org/display/JETTY/How+to+configure+SSL

2. Create the "org.ops4j.pax.web.properties" file in the "conf/cfgbackup" directory (if not already exists) with the following contents:

```
# Enable SSL
org.osgi.service.http.secure.enabled=true

# SSL Port
org.osgi.service.http.port.secure=8443

# Keystore created to hold SSL certificate
org.ops4j.pax.web.ssl.keystore=conf/keystore
# Keys to access SSL certificate
org.ops4j.pax.web.ssl.password=OBF:1o441vnc1u9t1u301vn61pda1fic1fei1pao1vnw1u2a1u9p1vn
```

3. [Optional] - To create the OBF: encrypted passwords. Go to bundle/application and run

```
java -cp pax-web-jetty-bundle-0.7.4.jar org.mortbay.jetty.security.Password
user "PASSWORD"
```

Where user can be anything and PASSWORD is the keystore key to encrypt. For more information:

http://docs.codehaus.org/display/JETTY/How+to+configure+SSL

4. [Optional] - Stop and restart Brand Mobiliser. Verify

   https://hostname:8443/brand

3.4.2.4 System.Properties

Each of the properties are annotated below just above the properties.

```
# Set the verbosity of PAX Logging API by default.
# Valid values are TRACE,DEBUG,INFO,WARN,ERROR,FATAL,or NONE;
# Null is default to DEBUG
org.ops4j.pax.logging.DefaultServiceLog.level=NONE

# OSGi Framework Events are controlled with this value
# Valid values are DEBUG,INFO,WARN,ERROR or if null, default to DEBUG
# By default, set to DEBUG so they don't appear unless we want them to
org.ops4j.pax.logging.service.frameworkEventsLogLevel=DEBUG

# Fix for Jetty problem of checking for large HTML forms, -1 = unlimited form size
org.mortbay.jetty.Request.maxFormContentSize=-1

# Disable URL rewriting from Jetty
#org.mortbay.jetty.servlet.SessionURL=none
```

3.4.3 Configuration Files in the conf/cfgbackup directory

The following are annotated descriptions of the configuration files available with the Brand Mobiliser installation.

3.4.3.1 Node Discovery

Filename: com.sybase365.arf.management.system.responder. properties

The configurations for node discovery mechanism. Apply to multi-node environment and need special setup to use this feature.

3.4.3.2 Felix Webconsole

Filename: org.apache.felix.webconsole.internal.servlet.OsgiManager.properties

Felix Webconsole is useful for development environment only. NOTE: please do not add other IPs to the "allowed.ip.list".

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Felix Webconsole
#
################################################################################
# The allowable IP addresses to access the console from
allowed.ip.list=127.0.0.1

# The name of the default configuration page when invoking the console
default.render=bundles
```

```
# The root path to the console
manager.root=/system/console

# The plugins to display on the console
plugins=org.apache.felix.webconsole.internal.compendium.LogServlet,org.apache.felix.we
bconsole.internal.core.BundlesServlet,org.apache.felix.webconsole.internal.core.Servic
esServlet,org.apache.felix.webconsole.internal.misc.LicenseServlet,org.apache.felix.we
bconsole.internal.system.VMStatPlugin,com.sybase365.mobiliser.framework.event.manageme
nt.webconsole.EventManagementPlugin

# The name of the HTTP Authentication Realm
realm=AIMS Management Console

# Currently hardcoded to username=sybase365 password=fr4nt1c
# Leave as empty
username=
password=
```

3.4.3.3 Logging with log4j

Filename: org.ops4j.pax.logging.properties

Logging is controlled by the org.ops4j.pax.logging.properties, the properties are dynamic and uses the log4j configuration format. See http://logging.apache.org/log4j/1.2/manual.html for information about the configuration format.

3.4.3.4 Jetty Servlet Container

Filename: org.ops4j.pax.web.properties

The Jetty servlet container is used to host web application.

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Jetty Servlet Container
#
################################################################################
# Settings for the embedded jetty servlet container
# default session timeout is 30 minutes
org.ops4j.pax.web.session.timeout=30

# AJP13 connector
#org.ops4j.pax.web.config.file=conf/jetty.xml
#org.ops4j.pax.web.worker.name=ajp13_worker
```

3.4.3.5 Brand Mobiliser Web UI

Filename: service.brand_webapp.properties

Configuration file for Brand Mobiliser UI.

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Brand Mobiliser UI
#
################################################################################
```

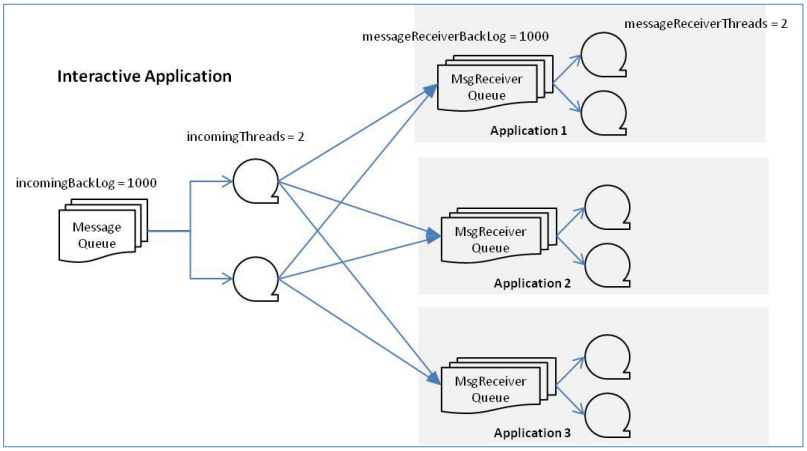
```
# Production mode(true) or development mode(false)
# In development mode, the web browser will show a link to debug popup window
productionMode=true

# Application theme
# Please do not change
theme=cupertino

# Folder to store the uploaded subscriber file
# Path relative to the Brand Mobiliser home directory
# The upload folder exists is the default standalone deployment
uploadFolder=upload
```

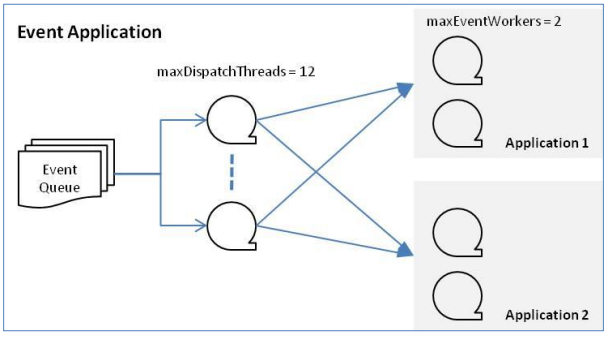
3.4.3.6 *Brand Mobiliser Processing Engine*

Filename: service.coreprocessing.properties

The configuration file for Brand Mobiliser Processing Engine enables tuning of the internal processing engine and queues for increase throughput based on a specific hardware configurations and channel throughput. Understanding of the internal mechanism and the configuration parameters will help in tuning the processing engine for optimal performance.

The processing engine system for interactive application is slightly different than the one for the event application as shown below, respectively.





Note: Changes to the file will not take effect until a restart of the server.

```
##############################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Brand Mobiliser Processing Engine
```

```
#
################################################################################
# Configuration for Incoming Queue - Interactive Application
# Number of threads to handle the dispatch of the incoming queue.
# Note that this should not exceed the messageReceiverThreads
incomingThreads=2

# Configuration for Incoming Queue - Interactive Application
# Maximum number of messages allowed in the incoming queue.   This is a global queue
# that handles all incoming messages before dispatching to a specific message
# receiver.
incomingBackLog=1000

# Configuration for MessageReceiver - Interactive Application
# A message receiver is created for each Live brand mobiliser application
# Configure the number of threads to handle the message queue for each message
# receiver.
messageReceiverThreads=2

# Configuration for MessageReceiver - Interactive Application
# Maximum number of messages allowed in each message receiver queue
# There is one queue for each message receiver.
messageReceiverBackLog=500

# Configuration for Event Application
# Maximum number of dispatching thread to process the event queue
maxDispatchThreads=12

# Configuration for Event Application
# Number of Processor per event application
maxEventWorkers=1

# Batch size in processing the subscriber file
batchLoadSize=100
```

3.4.3.7 DataSource

Filename: service.dsprovider.properties

Configuration for database JDBC connections and connection pool.

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       DataSource and Connection Pool
#
################################################################################

# Oracle JDBC Driver
driverClassName=oracle.jdbc.driver.OracleDriver

# Change the url to point to your Brand Mobiliser database installation
# url=jdbc:oracle:thin:@(host):(port):(sid)
url=jdbc:oracle:thin:@localhost:1521:xe

# Change the username and password to match that of the Brand Mobiliser database
username=MWIZ2
password=sql

# Database connection pool.
# Number of connections to keep open when idle.
maxIdle=2
```

```
# Database connection pool.
# Maximum number of connections that can be opened.
maxActive=50

# Validate connection settings for pool - useful when db
# connections are being dropped by an intervening network firewall
# - in which case set all testXYZ to true and adjust eviction
# time settings as appropriate
validationQuery=select 1 from DUAL
testOnBorrow=false
testOnReturn=false
testWhileIdle=false
timeBetweenEvictionRunsMillis=300000
numTestsPerEvictionRun=5
minEvictableIdleTimeMillis=600000

removeAbandoned=false
removeAbandonedTimeout=300
logAbandoned=false

# Specific optional setting for ExtendedDataSource
# Set if necessary to timeout on reads - in millis
# Default of 0 means no timeout.
readTimeout=0
```

3.4.3.8 Event Processor

Filename: service.event.core.properties

Event Processing engine configurations.

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Event Processing
#
################################################################################
# number of events to be retrieved during each regeneration
# select during handler catchup processing
regeneration.batch.size=10

# virtual capacity of the delay (events that will be processed after a short delay)
# if capacity reached a warning message is output in the logs
delayedq.capacity=1000

# virtual capacity of the processq (events that will be processed asap)
# if capacity reached a warning message is output in the logs
processq.capacity=1000

# virtual capacity of the catchupq (events that will be processed by handlers in
# catchup mode) - if capacity reached a warning message is output in the logs
catchupq.capacity=1000
```

3.4.3.9 Event Scheduler

Filename: service.event.quartz.properties

Event Scheduler configurations.

```
#############################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Event Scheduler (Quartz)
#
#############################################################################
# Main settings
scheduler.instanceName=MobiliserEventScheduler
scheduler.instanceId=AUTO

# Threadpool settings
threadPool.class=org.quartz.simpl.SimpleThreadPool
threadPool.threadCount=5
threadPool.threadPriority=5

# JobStore settings
jobStore.tablePrefix = QTZ_
jobStore.misfireThreshold=60000
jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreTX
jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
jobStore.useProperties=false
jobStore.selectWithLockSQL=SELECT * FROM {0}LOCKS UPDLOCK WHERE LOCK_NAME = ?

# Logging settings
triggHistory.class=org.quartz.plugins.history.LoggingTriggerHistoryPlugin
triggHistory.triggerFiredMessage=Trigger {1}.{0} fired job {6}.{5} at: {4, date,
HH:mm:ss MM/dd/yyyy}
triggHistory.triggerCompleteMessage=Trigger {1}.{0} completed firing job {6}.{5} at
{4, date, HH:mm:ss MM/dd/yyyy} with resulting trigger instruction code: {9}
```

3.4.3.10 Money Mobiliser

Filename: service.mobiliser.plugin.properties

Configure the Mobiliser Money end points. If you do not use the Mobiliser Web Services backend, please disregard this configuration file.

```
#############################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Mobiliser Endpoints
#
#############################################################################

# Please change url and authentication information to active Mobiliser installation.
wsBaseUrl=http://orinoco.sybase.com:7070/ws-core/
wsUsername=selfcare
wsPassword=6NWFRQnUPFI=
countrycode=49
language=en

# Please change url and authentication information to active Mobiliser installation.
jaxbBaseUrl=http://orinoco.sybase.com:7070/mobiliser/services
jaxbServiceConnectTimeout=2500
jaxbServiceReadTimeout=10000
jaxbServiceUser=confmgr
jaxbServicePassword=6NWFRQnUPFI=
jaxbServiceKeystore=
jaxbServiceKeystorePassword=
jaxbServiceKeyAlias=
```

```
# The following should not be modified.
jaxbBalanceServiceUrn=/money/balance
jaxbAuthoriseServiceUrn=/money/authorise
jaxbPreAuthoriseServiceUrn=/money/preauthorise
jaxbPayInvoiceServiceUrn=/money/payinvoice
jaxbTxnUpdateServiceUrn=/transaction/update
```

3.4.3.11 Web UI Security

Filename: service.webui.security.properties

Configure the security for the web UI. Currently, the configuration for integrating into the LDAP system.

```
################################################################################
#       AIMS - Brand Mobiliser Configuration
#
#       Web UI Security
#
################################################################################
# Choose one of the following beans to use as the system authentication manager
# AuthenticationManager = DB Authentication/DB Roles Sourced Authorisation
# LdapAuthenticationManager = LDAP Authentication/DB Roles Sourced Authorisation
authentication.bean=AuthenticationManager

# Only applicable when using LdapAuthenticationManager bean
# username=admin pwd=brandldap
#ldap.host=localhost
#ldap.port=389
#ldap.userpath=uid={uid},ou=people,o=sybase365
#ldap.security.authentication=SIMPLE
ldap.host=msw-rh464vm.sybase.com
ldap.port=10389
ldap.userpath=uid={uid},ou=people,o=sevenSeas
ldap.security.authentication=SIMPLE
```

3.4.4 Startup and Runtime

After the initial configuration is complete, Brand Mobiliser can be started using the “run.sh” script.

The following options are available:

Action	Description
start	Starts the server if not started.
stop	Stops the server if running.
restart/reload	Perform a stop then start of the server.
status	Checks to see if the server is running and displays the running process information.

Note that a “brand.pid” file is created on start to store the current instance’s process identifier.

3.4.4.1 Sample run session

```
sh-4.1$ ./run.sh start
Container is not running.
```

Starting container …

sh-4.1$ ./run.sh status

Running container instance:

UID PID PPID C STIME TTY TIME CMD

all 27520 1 40 10:32 pts/0 00:00:21 /usr/lib/jvm/java-6-sun/bin/java -Xms256M -Dnet.paybox.util.prefs.secret=paybox -Dcom.sun.management.jmxremote.port=5366 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.local.only=false -Djava.rmi.server.hostname=bashful -jar bin/felix.jar

sh-4.1$ ./run.sh stop

Running container instance:

UID PID PPID C STIME TTY TIME CMD

all 27520 1 24 10:32 pts/0 00:00:21 /usr/lib/jvm/java-6-sun/bin/java -Xms256M -Dnet.paybox.util.prefs.secret=paybox -Dcom.sun.management.jmxremote.port=5366 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.local.only=false -Djava.rmi.server.hostname=bashful -jar bin/felix.jar

Stopping container with pid: 27520

Checking for running container ...

**INFO * Logger terminated.*

**INFO * Logger terminated.*

Shutting down cascading work queue

Container is not running.

Container stopped.

After the server has started, you can then easily access the Brand Mobiliser WebAdmin UI by pointing your web browser to http://localhost:8080/

3.4.4.2 Diagnostics

If you encounter any issues with the running the container, check the logs under the “logs” directory.

By default, log groups are created for the different subsystems of Brand Mobiliser.

Log	Description
felix	Server kernel and misc log for all non application related messages.
brand	Main Brand Mobiliser application log.
spring	DI Application framework used by Brand Mobiliser.
ip2-iwf	Framework libraries used by Brand Mobiliser.
persist	Persistence log

Note that these are just the default grouping. If the logging properties configurations are modified, then the mapping described may no longer apply.

Some of the information may only be useful to Support Personnel. The "brand.log" provides the most useful application information and is the first place to check when issues arise.

3.5 J2EE Web Deployment

The J2EE web deployment model allows Brand Mobiliser to run on a J2EE compliant servlet container, like Apache Tomcat server, as a web application. **NOTE: The "j2ee" folder containing the binary (war file) is not included in the standard version.** The j2ee war file is provided to customer on a special request basis.

The relative paths in the sections below use the Brand Mobiliser home location created in "/opt/aims-brand-mobiliser-*versionNumber*".

3.5.1 Binary installation

A directory is created at `/opt/aims-brand-mobiliser-versionNumber` with these contents:

```
bin/
bundle/
conf/
j2ee/
sql/
license/
README.html
run.bat
run.sh
```

The war file is located in the "j2ee" folder with name "aims-brand-mobiliser-webapp-*versionNumber*.war". The *versionNumber* is the release version number of Brand Mobiliser.

3.5.2 Initial Setup and Configuration

The following system setup and configuration are necessary prior to deploying the war file. In the current version, the system setup is slightly different depending on the runtime J2EE servlet container.

3.5.2.1 Apache Tomcat

You need to know the root context where the Tomcat server is started. For the purpose of the following discussion, let's assume that Tomcat was started from the "tomcat_home/bin" folder.

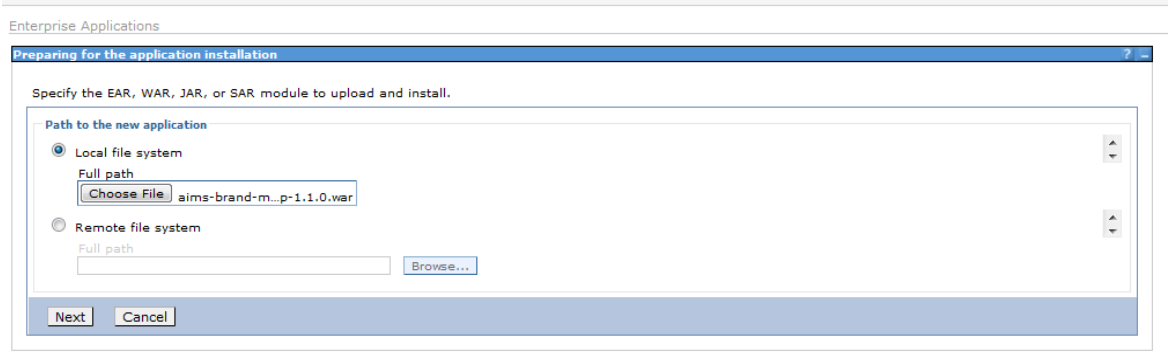
- Create the following folder structures in the "bin" directory:
  - aims
  - aims/_brand
  - aims/_brand/conf
  - aims/_brand/conf/cfgload
  - aims/_brand/conf/cfgbackup
- Note that "_brand" is constructed from the web application context path. If the context "brand" is already taken or the war is deployed to a different context, please modify the above directory structure accordingly by appending a "_" to the front of the web application context path.

- Copy the configuration properties files from the “conf/cfgbackup” of the binary distribution to the “aims/_brand/conf/cfgload” folder.
- Modify the properties files to meet the environment, as described in **“**Files in the conf/cfgbackup directory” section of the Stand Alone Deployment.
- Copy the war file (“aims-brand-mobiliser-webapp-*versionNumber*.war”) from the “j2ee” folder of the binary distribution to the “tomcat_home/webapps” folder. Note: the war file name will become the web application context name. **Note: it is recommended to change the war filename to “brand.war”, if not already used, prior to deploying it to tomcat. The web application is access as http://uri/brand**
- Start tomcat server.

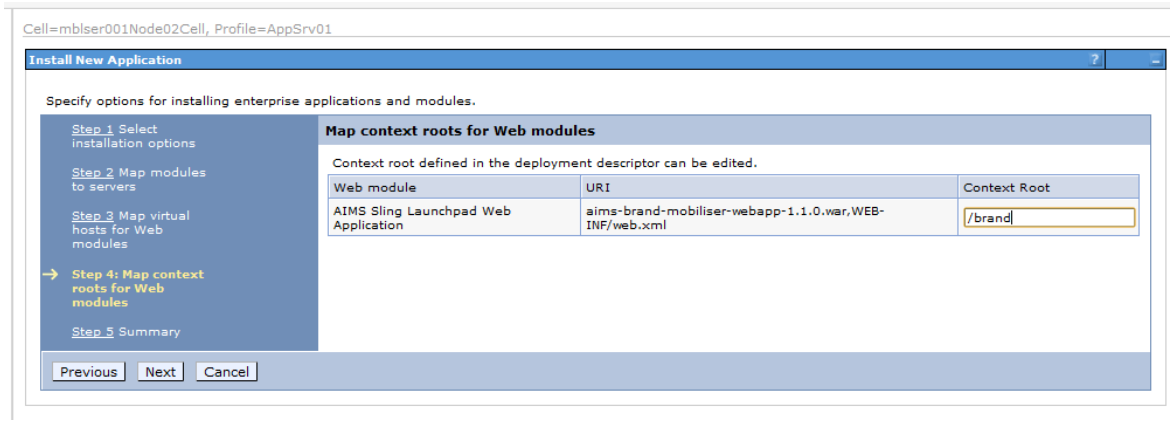
3.5.2.2 IBM Websphere Application Server

You need to know which server “profile” to deploy the web application to. Typically, the default profile is at “profiles/AppSrv01” under the WebSphere Application installed directory. For the purpose of these instructions, we refer to default profile directory as “was_home”

- Create the following folder structures in the “was_home” directory:
  - aims
  - aims/_brand
  - aims/_brand/conf
  - aims/_brand/conf/cfgload
  - aims/_brand/conf/cfgbackup
- Copy the configuration properties files from the “conf/cfgbackup” of the binary distribution to the “aims/_brand/conf/cfgload” folder.
- Modify the properties files to meet the environment, as described in **“**Files in the conf/cfgbackup directory” section of the Stand Alone Deployment.
- You should have the war (“aims-brand-mobiliser-webapp-*versionNumber*.war”) from the “j2ee” folder of the binary distribution available for upload from a web browser. Use the Websphere Integrated Solution console to deploy the war file.



- During the deployment using the Websphere Integrated Solution console, make sure to set the context root to “brand” (if not already used), so that the application is accessed as http://uri/brand.


Cell=mblser001Node02Cell, Profile=AppSrv01
Install New Application
Specify options for installing enterprise applications and modules.
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Map virtual hosts for Web modules
Step 4: Map context roots for Web modules
Step 5 Summary
Map context roots for Web modules
Context root defined in the deployment descriptor can be edited.
Web module
URI
Context Root
AIMS Sling Launchpad Web Application
aims-brand-mobiliser-webapp-1.1.0.war,WEB-INF/web.xml
/brand
Previous
Next
Cancel

4 Platform Maintenance & Tuning

This section provides some guidance on how to perform regular maintenances for the application. The maintenance frequency varies based on the amount of traffics (or in this cases messages) served by the application.

4.1 Database Tables Maintenance

Database is used to persist configuration, transaction and session data, as well as for storing logging. The information provided in this section is to help the DBA to incorporate the application tables into the overall database maintenance plan. As always, regular backups are essential for recovery in the event of unexpected failure.

The following tables requires regular archiving & truncation.

Logging Tables [Need regular Archiving in high traffic volume]

Table Name	Descriptions	Volume	Archive & Purge
M_CORE_STATUS_LOG	Engine Logs	High	See Core Rule
M_LOGIN_HISTORY	Grows with number of user login	High	See Login History Rule
M_MESSAGE_LOGS	Grows with number of subscribers and traffics	High	See Message Logs Rule
M_MESSAGE_RECEIVERS	Grows with number of subscribers and traffics	High	See Message Receivers Rule
M_SESSION_ATTRIBUTES	Grow with traffics; Session data	High	See Session Attributes Rule
M_SESSIONS	Grow with traffics; Session data	High	See Session Rules
M_SMAPP_TRANSITION_LOG	Grow with traffics; Session data	High	See Transitions Rules

4.1.1 Archive – Core Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_CORE_STATUS_LOG table can be purged based on the TIMESTAMP2 column

4.1.2 Archive – Login History Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_LOGIN_HISTORY table can be purged based on the TIMESTAMP2 column

4.1.3 Archive – Message Logs Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_MESSAGE_LOGS table can be purged based on the TIMESTAMP2 column

4.1.4 Archive – Message Receivers Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_MESSAGE_RECEIVERS table can be purged based on the MARKED_DELETED = 1 AND if there is no references in the M_MESSAGE_LOG table

4.1.5 Archive – Session Attributes Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule:

- M_SESSION_ATTRIBUTES table needs to be purged prior to purging the M_SESSIONS table
- M_SESSION_ATTRIBUTES table can be purged based on SESSIONS_ID column. The session_id no longer exists in the M_SESSIONS table.

4.1.6 Archive – Session Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_SESSIONS table can be purged based on CLOSED_DATE column.

4.1.7 Archive – Transitions Rule

Archiving and purging should be defined by the DBA based on the company policy. The following recommendation should be treated as guidance and be reviewed by the DBA.

Rule: M_SMAPP_TRANSITION_LOG table can be purged based on the TIMESTAMP2 column

4.2 Special Maintenances

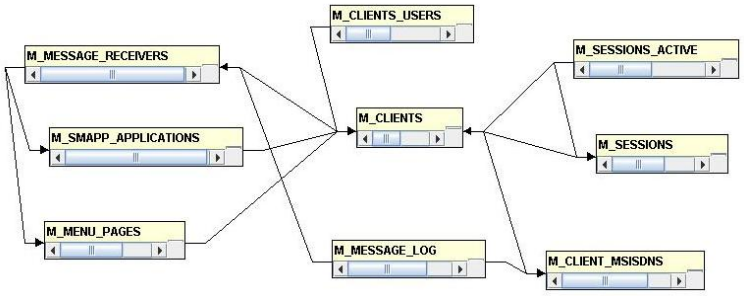
Currently, there is no mechanism provided on the UI to delete Workspace and User. This can be accomplished using the following steps.

4.2.1 Delete Workspace

The ability to delete workspace will be provided through the UI in the future release. In the interim, please follow the following steps to delete a workspace. **NOTE: database backup is recommended prior to performing the following operations.**

- Login using the username with ADMIN role and has been assigned to the workspace
    - Go to Setup Default Menu page. Remove all application assigned to the Default Menu. Then Approve the Default Menu.
    - Remove all the applications in the workspace from the Asset Page
    - Go to the Manage User page, unassigned all users from this workspace including the login user. If the login user is assigned to this workspace only, first assign the login user to a different workspace. Then unassigned the login user from this workspace.

- Using the database management tool
    - Go to M_CLIENTS table, and find the ID (let's say it's 9999) of the workspace to be deleted. The workspace name is in the NAME column.
    - Go to M_CLIENT_MSISDNS table, and find the ID (let's say it's 2222) where the CLIENTS_ID = 9999
    - Delete all entries in M_MESSAGE_LOG where CLIENTS_MSISDNS_ID = 2222
    - Delete an entry in M_CLIENT_MSISDNS table where ID = 2222
    - Delete all entries in M_SESSIONS_ACTIVE table where CLIENTS_ID = 9999
    - Delete all entries in M_SESSIONS table where CLIENTS_ID = 9999
    - Delete all entries in M_MESSAGE_RECEIVERS where CLIENTS_ID = 9999
    - Go to M_MENU_PAGES table, and find the ID (let's say 4444) where the CLIENTS_ID = 9999
    - Delete all entries in the M_MENU_PAGES_LANGS where ID = 4444
    - Delete all entries in the M_MENU_PAGES where CLIENTS_ID = 9999
    - Delete an entry in the M_CLIENTS table where ID = 9999



4.2.2 Delete User

The ability to delete user will be provided through the UI in the future release. In the interim, please follow the following steps to delete a user. **NOTE: database backup is recommended prior to performing the following operations.**

- Login using the username with ADMIN role and has been assigned to the same workspace as the user to be deleted.
    - Go to the Manage User page, unassigned the users from any roles and any workspaces but the default workspace.
- Using the database management tool
    - Find out the ID of the user from the M_USERS table (let's say 8888)
    - Delete all entries in M_CLIENTS_USERS table where the USERS_ID = 8888. (There should only be one entry)
    - Delete the entry in M_USERS table where ID = 8888

4.3 Log Files

Brand Mobiliser creates log files that are stored in the "log" folder. Please refer to "Installation and Configuration" section for the location of the log files. Please make sure that there are enough disk space for the log file.

4.4 Performance Tuning

Depending on the hardware configurations and the host server utilizations, the Brand Mobiliser processing engine can be tuned when necessary to increase throughput. The default setup (see **service.coreprocessing.properties**) should work in typical scenario. Depending on the configurations and utilization of the host server, the processing engine can be tuned to increase throughput.

Please refer to "Brand Mobiliser Processing Engine" section for detailed diagrams of the processing engine architecture and how to tune it.

Notes:

- **Please run performance benchmark on the new configurations before deploying to the production system.**
- **Changes to the "service.coreprocessing.properties" configuration file will not take effect until a restart of the Brand Mobiliser platform.**

Summary of the performance conducted at Sybase data center is presented below.

Hardware: HP DL 360 G6 – 1U standalone; 2x Intel X5450; 8x 2GB (16GB)

Server Configurations: One dedicated server for DB, and another for Brand Mobiliser

Performance Test of Interactive Application:

Software Configurations:

Processing Engine: messageReceiverThreads=10 messageReceiverBackLog=50000 incomingThreads=10 incomingBackLog=50000	Datasource: maxIdle=50 (50 maximum performance) maxActive=25 (50 maximum performance)

Load: 10,000 messages
Result: 110 messages/sec

Performance Test of Event Application:

Software Configurations:

Processing Engine: maxDispatchThreads=12 maxEventWorkers=4	

Load: 5 million messages

Results:

Up to 88	Rate = 88 msgs/sec
Up to 3,520	Rate = 93 msgs/sec
Up to 52,695	Rate = 92 msgs/sec
Up to 174,668	Rate = 94 msgs /sec
Up to 3,576,594	Rate = 82 msgs/sec
Up to 4,999,899	Rate = 97 msgs/sec

A. Appendix - Glossary

A.1 Acronyms

- SMS: The **S**hort **M**essage **S**ervice is a service available on most digital mobile phones, other mobile devices (e.g. a PocketPC) and some fixed phones that permits the sending of short messages.
- MMS: The **M**ultimedia **M**essaging **S**ervice is a standard for a telephony messaging systems that allow sending messages that includes multimedia objects (images, audio, video, rich text) and not just text messages as in SMS
- WAP: The **W**ireless **A**pplication **P**rotocol is an open international standard for applications that use wireless communication. Its principal application is to enable access to the Internet from a Mobile Phone or PDA.
- User: In this manual a User is the person that is working with the Mobile Wizard e.g. a Product Manager or Marketing Manager
- MSISDN: The **M**obile **S**tation **I**ntegrated **S**ervices **D**igital **N**etwork (MSISDN) refers to a fixed number of digits (may be 10 or 15-digit, depends of service provider) that is used to refer to a particular mobile device.
- MO: **M**obile **O**riginated. All incoming messages from the Customer to the Mobile Wizard
- MT: **M**obile **T**erminated. All outgoing messages from the Mobile Wizard to the Customer
- SOAP: SOAP (originally **S**imple **O**bject **A**ccess **P**rotocol) is a protocol for exchanging XML-based messages over computer networks, normally using HTTP.
- SMPP: The **S**hort **M**essage **P**eer-to-**P**eer protocol is a telecommunications industry protocol for exchanging SMS messages between SMS peer entities such as short message service centers

A.2 Definitions

- Shortcode: (also known as short numbers) are special telephone numbers, significantly shorter than full telephone numbers that can be used to address SMS and MMS messages from mobile phones or fixed phones. There are two types of short codes: dialing and messaging. Short codes are widely used for value-added services such as television program voting, ordering ringtones, charity donations and mobile services. Messages sent to short code can be billed at a higher rate than a standard SMS and may even subscribe a customer to a recurring monthly service that will be added to their mobile phone bill until they text, for example, the word "STOP" to terminate the service. [Source: Wikipedia]

B. Appendix - User Roles

B.1 Roles, Purposes & Accesses

Role and permission are used interchangeably in this release but role will be used in this document as well as in the future release of the software. The role is used to control access to screen or controls (such as, button, input field, etc.) Currently, there are three pre-defined roles:

- SUPER ADMIN
- ADMIN
- APPLICATION ADMIN

The purpose of these roles and who should be assigned to these roles are described below.

Roles	Purposes and Access
SUPER ADMIN	▪ The default user (i.e., username=ADMIN) has a SUPER ADMIN role. It is recommended to leave it unmodified except to change the password. The default password is “brand”, with no quotes. Additional user(s) can be created and assigned a SUPER ADMIN role ▪ The SUPER ADMIN user can also be referred to as the platform administrator. Assign this role to the individual that will administer the system and Brand Mobiliser application ▪ Unlimited access (Client comprehensive) ▪ Exclusive access to *Workspace Administration* → **Manage Workspace** screen for creating or managing workspaces ▪ Exclusive access to *Workspace Administration* → **Manage Channel Configurations** screen for configuring channels. Channels are used to deliver inbound and outbound messages. Also, the processing engine can be ”Stop”, “Start” or ”Restart” from this screen ▪ The workspace module on the Dashboard (top right hand column) displays the short or long code(s) that have been provisioned for the current workspace. The platform administrator can click on the module title (i.e., Workspace) to go to the *Workspace Administration* → **Manage Workspace** with the current workspace pre-selected. ▪ All other functionalities available to the roles below
ADMIN	▪ The ADMIN user can also be referred to as the workspace administrator. Each workspaces should have an assigned workspace administrators who can create users for the workspace, setup the Default Menu, and Manage Sessions ▪ In the development environment with no active channels, the application developer can be assigned to the workspace administration role. ▪ Access to all screens except: *Workspace Administration* → **Manage Workspace** and *Workspace Administration* → **Manage Channel Configurations**

	▪ Create and Manage workspace user from the *Workspace Administration* → **Manage User** screen ▪ Manage active sessions that are created for the interactive application(s) using the *Workspace Administration* → **Manage Sessions** screen ▪ Monitor the Brand Mobiliser engine log using the *Workspace Administration* → **Core Engine Log** screen ▪ Create report for the traffic generated by the interactive application(s) in the workspace. ▪ All other functionalities available to the roles below
APPLICATION ADMIN	▪ Non administration role; Assign to QA team in the QA environment ▪ No access to the Workspace Administration navigation menu ▪ Users who will work with application, event and subscriber. Have full access to the development tools including: create new, modify, approve, simulate and delete. ▪ Also have access to Manage Categories
APPLICATION OWNER	▪ Currently, has the most restrictive access. Typically assigned QA team and application developers in the production environment when necessary ▪ Access to mostly view functionalities; Definitely do not have access to activation functionalities.

Note: More roles may be added as needed in a future release.

C. Appendix - Regular Expressions

C.1 Regular Expression Introduction

This introduction will quickly get you up to speed with regular expressions. Obviously, this brief introduction cannot explain everything there is to know about regular expressions. For detailed information, consult a regular expression tutorial that you can find e.g. here: http://www.regular-expressions.info/tutorial.html. Each topic in this quick start corresponds with a topic in the tutorial, so you can easily go back and forth between the two.

Introduction

A regular expression, or regex for short, is a pattern describing a certain amount of text.

Literal Characters

The most basic regular expression consists of a single literal character, e.g.: a. It will match the first occurrence of that character in the string. If the string is Jack is a boy, it will match the a after the J.

This regex can match the second a too. It will only do so when you tell the regex engine to start searching through the string after the first match. In a text editor, you can do so by using its "Find Next" or "Search Forward" function. In a programming language, there is usually a separate function that you can call to continue searching through the string after the previous match.

Eleven characters with special meanings: the opening square bracket [, the backslash \, the caret ^, the dollar sign $, the period or dot ., the vertical bar or pipe symbol |, the question mark ?, the asterisk or star *, the plus sign +, the opening round bracket (and the closing round bracket). These special characters are often called "metacharacters".

If you want to use any of these characters as a literal in a regex, you need to escape them with a backslash. If you want to match 1+1=2, the correct regex is 1\+1=2. Otherwise, the plus sign will have a special meaning.

Character Classes or Character Sets

A "character class" matches only one out of several characters. To match an a or an e, use [ae]. You could use this in gr[ae]y to match either gray or grey. A character class matches only a single character. gr[ae]y will not match graay, graey or any such thing. The order of the characters inside a character class does not matter.

You can use a hyphen inside a character class to specify a range of characters. [0-9] matches a single digit between 0 and 9. You can use more than one range. [0-9a-fA-F] matches a single hexadecimal digit, case insensitively. You can combine ranges and single characters. [0-9a-fxA-FX] matches a hexadecimal digit or the letter X.

Typing a caret after the opening square bracket will negate the character class. The result is that the character class will match any character that is not in the character class. q[^x] matches qu in question. It does not match Iraq since there is no character after the q for the negated character class to match.

Shorthand Character Classes

\d matches a single character that is a digit, \w matches a "word character" (alphanumeric characters plus underscore), and \s matches a whitespace character (includes tabs and line breaks). The actual characters matched by the shorthand's depend on the software you're using. Usually, non-English letters and numbers are included.

Non-Printable Characters

You can use special character sequences to put non-printable characters in your regular expression. Use \t to match a tab character (ASCII 0x09), \r for carriage return (0x0D) and \n for line feed (0x0A). More exotic non-printables are \a (bell, 0x07), \e (escape, 0x1B), \f (form feed, 0x0C) and \v (vertical tab, 0x0B). Remember that Windows text files use \r\n to terminate lines, while UNIX text files use \n.

Use \xFF to match a specify character by its hexadecimal index in the character set. E.g. \xA9 matches the copyright symbol in the Latin-1 character set.

If your regular expression engine supports Unicode, use \uFFFF to insert a Unicode character. E.g. \u20A0 matches the euro currency sign.

All non-printable characters can be used directly in the regular expression, or as part of a character class.

The Dot Matches (Almost) Any Character

The dot matches a single character, except line break characters. It is short for [^\n] (UNIX regex flavours) or [^\r\n] (Windows regex flavours). Most regex engines have a "dot matches all" or "single line" mode that makes the dot match any single character, including line breaks.

gr.y matches gray, grey, gr%y, etc. Use the dot sparingly. Often, a character class or negated character class is faster and more precise.

Anchors

Anchors do not match any characters. They match a position. ^ matches at the start of the string, and $ matches at the end of the string. Most regex engines have a "multi-line" mode that makes ^ match after any line break, and $ before any line break. E.g. ^b matches only the first b in bob.

\b matches at a word boundary. A word boundary is a position between a character that can be matched by \w and a character that cannot be matched by \w. \b also matches at the start and/or end of the string if the first and/or last characters in the string are word characters. \B matches at every position where \b cannot match.

Alternation

Alternation is the regular expression equivalent of "or". cat|dog will match cat in About cats and dogs. If the regex is applied again, it will match dog. You can add as many alternatives as you want, e.g.: cat|dog|mouse|fish.

Repetition

The question mark makes the preceding token in the regular expression optional. E.g.: colou?r matches colour or color.

The asterisk or star tells the engine to attempt to match the preceding token zero or more times. The plus tells the engine to attempt to match the preceding token once or more. <[A-Za-z][A-Za-z0-9]*> matches an HTML tag without any attributes. <[A-Za-z0-9]+> is easier to write but matches invalid tags such as <1>.

Use curly braces to specify a specific amount of repetition. Use \b[1-9][0-9]{3}\b to match a number between 1000 and 9999. \b[1-9][0-9]{2,4}\b matches a number between 100 and 99999.

C.2 Regular Expression Examples

Examples:

[egh]	one of the indications „e“, „g“ or „h“
[0-6]	a number of „0 “to „6 “(hyphens are indicator for a range)
[A-Za-z0-9]	any Latin letter or any number
[^a]	any indication except „a “(„^ “at the beginning of an indication class negates it)
[:alpha:]	Letter: [:lower:] and [:upper:].
[:blank:]	Blank and tabulator.
[:cntrl:]	Control character. In the ASCII that is the characters 00 to 1F, and 7F (DEL).
[:digit:]	Numbers: 0, 1, 2,... to 9.
[:graph:]	Graphic indications: [:alnum:] and [:punct:].
[:lower:]	Small letter: not necessarily only from a to z.
[:print:]	Printer graphics: [:alnum:], [:punct:] and blanks.
[:punct:]	Indication how: ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ .
[:space:]	Whitespace: Horizontal and vertical tabulator, line and page feed, carriage return and blank.
[:upper:]	Capital letter : not necessarily only from A to Z.
[:xdigit:]	Hexadecimal numbers: 0 to 9, A to F, a to f.

- **\d** : a number [0-9]
- **\D** : an indication, which is not a number, thus [^\d]
- **\w** : a letter, a number or underlined, thus [a-zA-Z_0-9] (and possibly further letters)
- **\W** : an indication, which neither letter nor number still underlined are, thus [^\w]
- **\s** : Whitespace; mostly the class of the control characters \f, \n, \r, \t and \v
- **\S** : an indication, which is not Whitespace [^\s]

Example Phone Number

- (0049(\d{12,13})) matches all phone Numbers that begin with 0049 and got 12 or 13 digits after that.
- (0049(\d+)) matches all Numbers that begin with 0049 and min 1 digits after that.

Example Amount

- (\d+[\.\,]?\d*) matches 50000,00 or 8976.12 but not 50.988,12

Example Letter and Number

- ([:upper:] [0-6]) Finds everything that looks like: A1 … G4 … but not G8

SYBASE, INC.
WORLDWIDE HEADQUARTERS
ONE SYBASE DRIVE
DUBLIN, CA 94568-7902 USA
Tel: 1 800 8 SYBASE

www.sybase.com

DOCUMENT ID: DC01691-01-0120-01
LAST REVISED: September 2011

SYBASE®
An SAP® Company