

Technical Brief: BlackBerry Dynamics App Policies

Table of Contents

- 1 APPLICATION POLICIES.....3
 - 1.1 FEATURE OVERVIEW3
 - 1.2 FEATURE DESCRIPTION3
 - 1.2.1 *Publication and Loading Options*.....4
 - 1.2.2 *Application Version Compatibility*4
 - 1.3 TECHNICAL DOCUMENTATION5
 - 1.3.1 *Mobile Applications*5
 - 1.3.2 *File Format*5
 - 1.4 FEATURE UTILISATION DISCUSSION.....6
- 2 APPENDIX: EXAMPLE OF APPLICATION-SPECIFIC POLICIES7
 - 2.1 APPLICATION EXAMPLE.....7
 - 2.2 EXAMPLE POLICY DEFINITION.....7
 - 2.2.1 *Example Policy Definition Code*8
 - 2.3 EXAMPLE POLICY PAGE.....10
 - 2.4 EXAMPLE MOBILE IMPLEMENTATION.....11
 - 2.4.1 *Example Settings Object*.....12
 - 2.4.2 *Example Policy Change*12

1 Application Policies

Feature Overview

The application policies feature enables BlackBerry Dynamics developers to add policies that are specific to their application to a Good Control server or BlackBerry UEM, depending on the environment that they are using.

As installed, the Good Control (GC) or BlackBerry UEM console policy screens show a fixed list of policies. These policies relate to password complexity, device compliance, and other generic aspects of usage that apply to all BlackBerry Dynamics apps.

This feature allows the developer of a BlackBerry Dynamics app to create additional policies that are specific to their app. These specific policies appear in the same user interface as the generic policies.

This feature can be used to enable configuration of a BlackBerry Dynamics app to take place in the GC or BlackBerry UEM console, instead of in a separate user interface.

Feature Description

The BlackBerry Dynamics app developer can create a number of policies for administration of their app. These will be defined in terms of the following:

ID and Display Name of the policy

Data Type that the policy value can take. For example, the policy could have a true or false value, meaning that the administrator can configure the policy setting as on or off. This also affects how the policy is presented in the GC console. The following table lists the value types that are offered and how policies of that type appear in the GC or BlackBerry UEM console.

Value Type	Presentation	Notes
Boolean	Checkbox	
List, for single selection or multiple selection	Drop-down, or other list control	The definition includes the list of values
Free text, allowing either a single line or multiple lines	Text box	
Hidden text	Not shown	This could be used to support versioning, see under Application Version Compatibility, below.

Default value that is set for the policy if no selection has been made in the GC or BlackBerry UEM console. For a hidden policy, the value would always be the default.

A number of options are provided for the publication and loading of application policies, see below. In all cases, the outcome is that the policy definitions are loaded on a GC or BlackBerry UEM server. The GC or BlackBerry UEM console will then show controls for the app's policies in the policy set editing screens.

An administrator can then use the console to add settings for the application's policies to the policy sets in the GC or BlackBerry UEM.

When an end user installs and runs the app, the policy set that applies to the end user will be retrieved to the device. Any application policies will be included in that retrieval. Generic policies will be enforced by the BlackBerry Dynamics runtime. Application policies will be enforced by the application code.

See also the Example of Application-Specific Policies in the appendix.

Publication and Loading Options

A number of options are provided for the publication and loading of policy definitions.

A policy definition can be uploaded directly to the GC or BlackBerry UEM, using the console. The option to do this is on the app management screen. This can be used for applications that are in-house and not published through the BlackBerry Dynamics Network (BDN). This is also useful for developing and debugging BlackBerry Dynamics apps with specific policies.

A policy definition can also be uploaded to the BDN, for Partner applications. When a Partner application is published to an enterprise, the policy definition is pushed to GC or BlackBerry UEM.

The policy definition can be uploaded to the BDN at the same time as an app is registered, before the app is published to any customers. A policy definition can also be uploaded for a Partner app that has already been published. In that case, the definition is immediately pushed to GC or BlackBerry UEM servers of the enterprises to which the app has been published.

Application Version Compatibility

The BlackBerry Dynamics platform does not as such support versioning of application policy definitions. However, the platform does provide administrators and developers with the tools to manage upgrading of apps that have associated specific policies.

A policy definition can include policies that are not shown in GC or BlackBerry UEM: hidden policies. A version number for the policy definition could be set as the default value of a hidden policy. The app would then retrieve the version number with the rest of the policy values and could take appropriate action.

For example, the app on the device could check whether the policy definition is a later version than it supports, and then display an error message to the end user. The message could advise the user of the need to upgrade.

The administrator can already block specific versions of BlackBerry Dynamics apps from running. This means that the administrator could, in theory, block an old version of an app that was known to be incompatible with new app policies. However, this type of blocking would lead to the container being wiped and the loss of any user data on the device.

Technical Documentation

Technical documentation for developers is available on the BlackBerry Dynamics Network (GDN) website, as follows.

Mobile Applications

There is a dedicated API for reading the application-specific policy settings that apply to the current end user. This can be accessed through either of the following functions in the GD runtime object interface class:

```
getApplicationPolicy  
getApplicationPolicyString
```

These functions are documented in the [relevant API References](#).

An event will be dispatched to the application code when application-specific policies are changed on the GC console. The event will be a `GDApEvent` with `GDApEventPolicyUpdate` type.

This is documented in the [relevant API references](#).

File Format

The policy definition file format is [documented in the appendix of either API Reference](#).

The same documentation is included in both locations.

Feature Utilisation Discussion

To use this feature, a BlackBerry Dynamics app developer must implement all of the following:

- Policy definition, to be uploaded to BDN or GC or BlackBerry UEM.
- Application code to utilise the BlackBerry Dynamics API to access the policy settings.
- Application code to enforce the policy settings in the mobile client's user interface and processing.

Alternatively, to deliver the kind of configuration that this feature supports without using this feature, the developer would have to implement all of the following instead:

- Policy store and user interface in own app server, with own admin console.
- Application code to retrieve policy settings from own app server.
- Application code to enforce the policy settings, as above.

BlackBerry Dynamics developers can choose whether to use this feature. The choice may be made based on the relative difficulty of implementing the above items, for the developer. An established application might already include all the items listed in the alternative implementation.

The following factors might also be considered.

The GC or BlackBerry UEM console app policy user interface will not include any app branding other than the name of the app and policies. The developer's logo, colour scheme and other branding elements will not be displayed, for example. There would be no scope for including a URL, for example to the developer's upselling site.

Without this feature, this kind of configuration would have to be done in a separate administrative tool that was supplied with the BlackBerry Dynamics app. Administrators would then have two consoles to learn and use, which is a disincentive to purchase.

Appendix: Example of App-Specific Policies

This section gives an example of usage of the BlackBerry Dynamics app policies feature. This is for readers wanting something more concrete than the above descriptions. The example is for illustration purposes only and is not based on any known requirements, customers or partners.

Application Example

In this example, the application is as follows.

The app shows a dashboard presenting up-to-the-minute internal enterprise news. There are a number of possible tabs on the dashboard:

- *Personnel* shows information about recent hires, departures, offers and interviews.
- *Sales* shows deals that are approaching closure, or have just been closed or lost.
- *Press* shows press releases and current media coverage.
- *Projects* shows the status of roll-outs, office moves, and other internal programmes.

The application can also be used to post news updates, which can include photographs.

The application is called Dash-dash.

Dash-dash includes a mobile BlackBerry Dynamics app, and an app server that is installed behind the enterprise firewall. Data from the server is retrieved by the app and stored securely on the device.

Dash-dash is developed by the software house Fillwater, who are a registered BlackBerry Dynamics Partner.

Example policy definition

Fillwater defines the following policies for their application Dash-dash.

Policy	Display name	Data Type	Notes	Default
Pers	Show Personnel	On or Off	Sets whether the named tab is shown to the user	Off
Press	Show Press	On or Off		On
Proj	Show Projects	On or Off		On
Sales	Show Sales	On or Off		Off
Retain	Retention Period	One of: 1 day 1 week 1 month 1 year	Updates older than the selected age are deleted from the user's device	1 year
canUpdate	Can Update	On or Off	Sets whether the user can post updates	Off

Policy	Display name	Data Type	Notes	Default
photoUpdate	Can Post Photos	On or Off	If the user can update, sets whether the user can post photos	Off

Fillwater creates a policy definition for the above. The policy definition is uploaded to BDN and associated with the Dash-dash application, which is published through BDN.

Example Policy Definition Code

The following shows how the code for the above policy definition might look.

```
<?xml version="1.0" encoding="utf-8"?>
<apd:AppPolicyDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:apd="urn:AppPolicySchema1.good.com"
  xsi:schemaLocation="urn:AppPolicySchema1.good.com AppPolicySchema.xsd" >

  <pview>
    <pview type="tabbed" key="display">
      <title>Display settings</title>
      <pe ref="tabs" />
      <pe ref="Retain" />
    </pview>

    <pview type="tabbed" key="update">
      <title>Update settings</title>
      <pe ref="canUpdate" />
      <pe ref="photoUpdate" depends="canUpdate" />
    </pview>

    <pview type="tabbed" key="about">
      <title>About</title>
      <heading>Dash-dash</heading>
      <desc>Up-to-the minute news about your enterprise,
        powered by Fallwater.</desc>
    </pview>

    <pe ref="policyVersion" />
  </pview>

  <setting name="policyVersion" >
    <hidden>
      <key>version</key>
      <value>2.0</value>
    </hidden>
  </setting>

  <setting name="tabs">
    <multiselect>
      <key>tabs</key>
      <label>Dashboard Tabs to Display</label>
      <value ref="defaultTabs" />
      <options ref="allTabs"/>
    </multiselect>
  </setting>

  <dl name="allTabs">
    <dv><desc>Show Personnel</desc><value>Pers</value></dv>
    <dv><desc>Show Press</desc><value>Press</value></dv>
    <dv><desc>Show Projects</desc><value>Proj</value></dv>
    <dv><desc>Show Sales</desc><value>Sales</value></dv>
  </dl>

  <do name="defaultTabs">
    <value>Press</value>
    <value>Proj</value>
  </do>
</apd:AppPolicyDefinition>
```

```

</do>
<setting name="Retain">
  <select>
    <key>Retain</key>
    <label>Retention Period</label>
    <value>1 year</value>
    <options ref="RetainOptions" />
  </select>
</setting>
<dl name="RetainOptions" dtype="string">
  <dv><desc>1 Day</desc><value>1 day</value></dv>
  <dv><desc>1 Week</desc><value>1 week</value></dv>
  <dv><desc>1 Month</desc><value>1 month</value></dv>
  <dv><desc>1 Year</desc><value>1 year</value></dv>
</dl>

<setting name="canUpdate" >
  <checkbox>
    <key>canUpdate</key>
    <label>Can Update</label>
    <value>>false</value>
  </checkbox>
</setting>
<setting name="photoUpdate" >
  <checkbox>
    <key>photoUpdate</key>
    <label>Can Post Photos</label>
    <value>>false</value>
  </checkbox>
</setting>
</apd:AppPolicyDefinition>

```

The code contains a definition of the data structure for the policies, and instructions for how the policies are presented in the GC or BlackBerry UEM console user interface.

Note that a hidden setting is included, for the policy version.

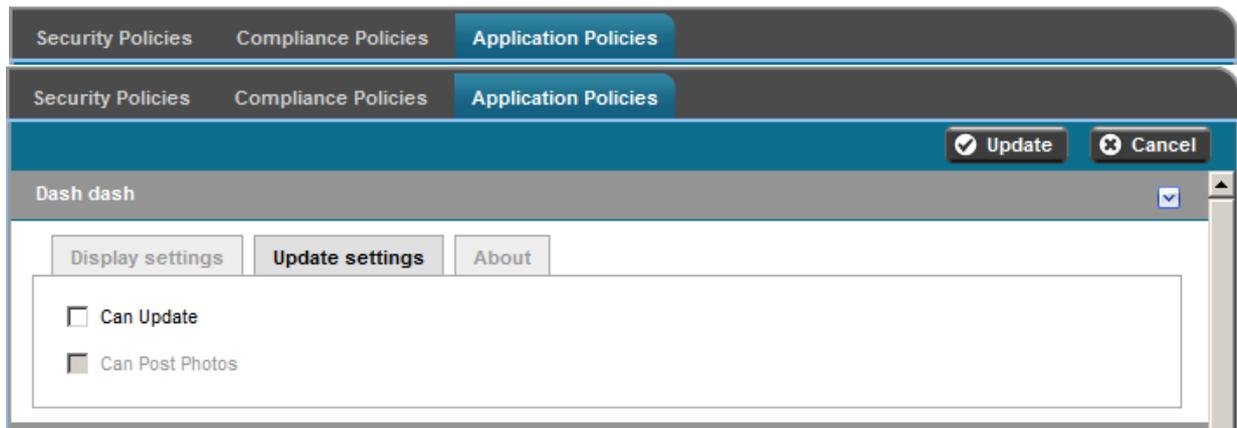
Example policy page

A large enterprise, MacFinn Corporation, purchases Dash-dash from Fillwater.

In the BDN, MacFinn Corporation makes a vendor connection to Fillwater. Fillwater then publishes Dash-dash to MacFinn Corporation. The application details for Dash-dash are sent to the MacFinn GC server, including the policy definition.

An admin at MacFinn logs in to the GC or BlackBerry UEM console. The admin opens the Policy Set editing user interface. The policies for Dash-dash are displayed there and can be edited, like other policies.

The following screen captures show how the policies appear in the GC or BlackBerry UEM console user interface. There are two tabs that contain actual settings: the Display settings tab, and the Update settings tab.



The on-off policies appear as checkboxes. The one-of policy, Retention Period, appears as a drop-down. Settings that are dependent on other settings are dimmed, when the other setting is not ticked.

The MacFinn admin can now make different Dash-dash policy selections for different GD policy sets. For example:

Policy Set	Notes	Show Personnel	Show Press	Show Projects	Show Sales	Retention Period	Can Update
Default	Default for all users	Off	On	On	Off	1 year	Off
HR	Human Resources staff	On	On	On	Off	1 year	Off
Sales	Sales staff	Off	On	On	On	1 year	Off
Executive	Executives only	On	On	On	On	1 month	Off

Note that these are existing GD policy sets, to which application-specific policies have been added.

The admin can select which dashboard information is displayed to which users. In this case, the admin has selected to show all information to executives, but only show Press and Projects information to users in general. (Formally, executives means those users to whom the Executives policy set applies.) The admin has selected that no users can post updates.

If Dash-dash did not utilise application-specific policies, then the admin would have to use another console to configure the product. This might be a console hosted on the Dash-dash application server. There are advantages and disadvantages to this approach, see under Feature Utilisation Discussion, above.

Example Mobile Implementation

The mobile client piece of the Dash-dash solution is a BlackBerry Dynamics app. The Dash-dash mobile application would therefore include an instance of the BlackBerry Dynamics runtime. The Dash-dash application code would make calls into the BlackBerry Dynamics runtime library API.

For application policies, there would be a call to the `getApplicationPolicy` function. See under Reference Documentation, above. Following that, the application would read the returned policies and adjust behaviour accordingly.

In the case of Dash-dash, this would mean:

- Deleting old data from the secure store, in line with the Retain setting.
- Only retrieving data for a particular tab if the corresponding policy is set to on. For example, only retrieving Personnel data if the Pers policy is on.
- Deleting data that was previously retrieved but which is no longer allowed for the current user following a policy change.

Note that the BlackBerry Dynamics API makes the application policy settings available but does not enforce them. The BlackBerry Dynamics API implicitly only returns the policy settings that apply to the current end user.

Example Settings Object

The following shows a possible settings object that would be returned by the application policy API. In this example, the object has been retrieved as a collection and printed in a JSON-like notation.

```
{
  about = {
  };
  display = {
    Retain = "1 year";
    tabs = (
      Press,
      Proj
    );
  };
  update = {
    canUpdate = 0;
    photoUpdate = 0;
  };
  version = "1.0";
}
```

Note that the tab structure is reflected in the data structure.

The policy settings show a retention period of one year, that the dashboard tabs for Press and Projects are to be displayed, and that updates are not allowed. The version of the policy settings object is 1.0.

Example Policy Change

The BlackBerry Dynamics runtime retrieves the Dash-dash specific policies at the same time as it retrieves generic policies. This means at application start-up, but also when triggered by a change to the applicable policy set in the GC or BlackBerry UEM console.

Changes to Dash-dash policies trigger the dispatch of an event to the app, which must handle the event appropriately. For example:

1. A decision is made to remove visibility of the Projects tab from general users.
2. The MacFinn administrator edits the default BlackBerry Dnamicss policy set, selecting Proj Off by unchecking the Show Projects checkbox.
3. The policy change is pushed out to all Dash-dash clients being run by MacFinn end users to whom the Default policy set applies.
4. The GD runtime dispatches a policy-change event to the handler in the app.
5. The app immediately removes the Projects tab from its user interface, and deletes any Projects updates from its secure store.

6,454 along with numerous other U.S. and foreign patents and applications pending.